

IBM Republishing Permission #30456

IBM gave permission to republish:

- P1071 - Installing and Supporting OS2 2.1 - 1st Edition
- P1071 - Installing and Supporting OS2 2.1 - April 1994
- P1072 - OS2 2.1 Advanced Supported Workshop Course
- S7049 - Extended Services for OS2 Communication Manager - Student - Lab Notebook

IBM hereby grants to You a revocable, non exclusive, nontransferable, worldwide permission to reproduce, distribute to the public or display to the public, in electronic form only, the Documents, for educational and non-commercial purposes, under all copyright interests IBM owns in the Documents. For the avoidance of doubt, You may not prepare or reproduce or distribute any derivative works of the documents.

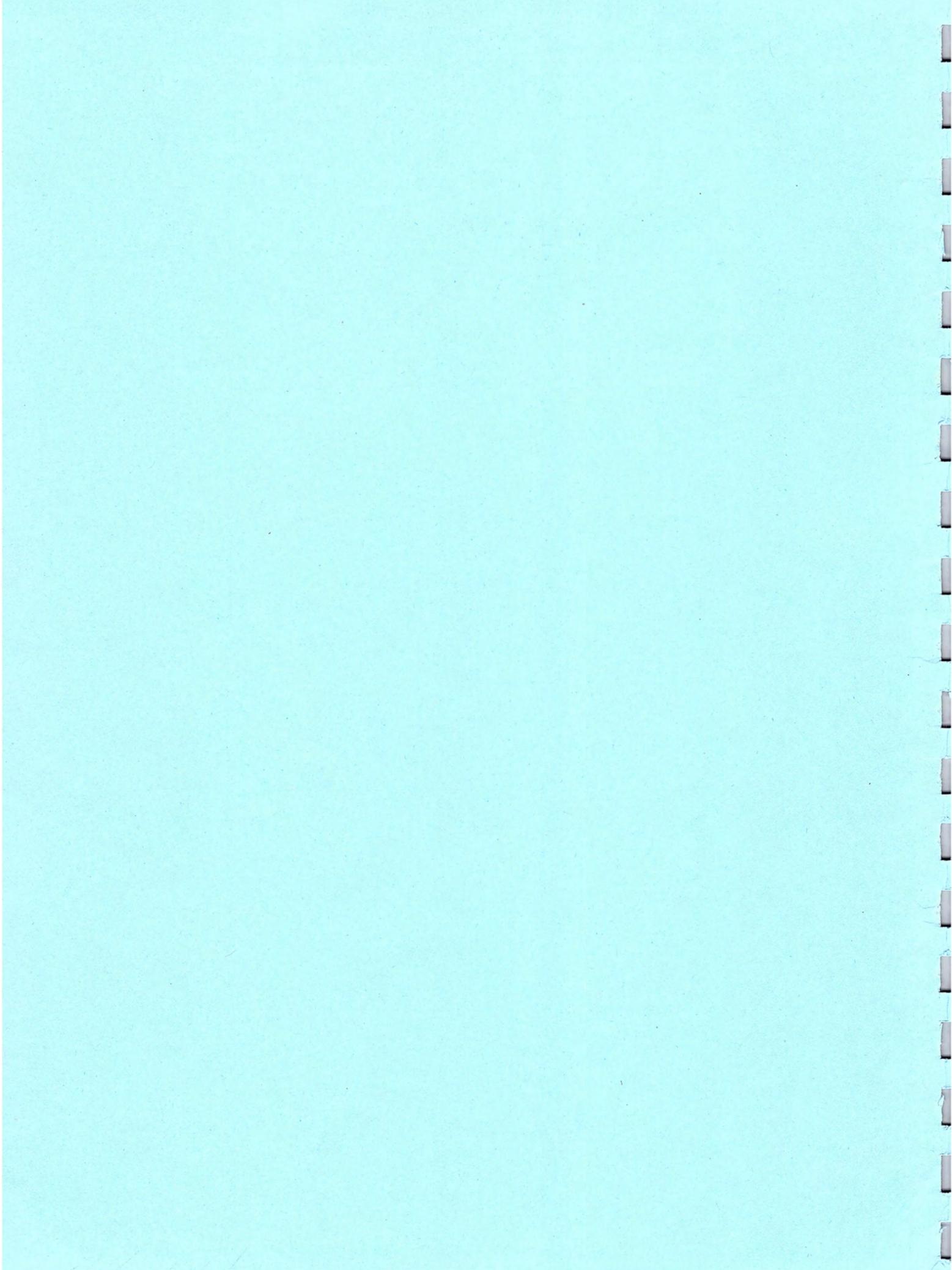
IBM MAKES NO WARRANTY, REPRESENTATION OR INDEMNIFICATION OF ANY KIND, EITHER EXPRESSED OR IMPLIED, CONCERNING ANY DOCUMENT, DESCRIPTION, OR THIS POLICY, WHICH ARE ALL PROVIDED "AS IS". THE WARRANTIES THAT IBM EXPLICITLY DISCLAIMS INCLUDE THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTY OR REPRESENTATION OF NON-INFRINGEMENT OF ANY THIRD PARTY'S INTELLECTUAL PROPERTY RIGHTS.

Dated: 2021-08-23

**Installing and Supporting
OS/2 2.1
P1071**

April 20, 1994

Student Notes



**Installing and Supporting
OS/2 2.1
P1071**

April 20, 1994

Student Notes

This publication has been produced using the IBM Document Composition Facility
(program number 5748-XX9) and CorelDRAW!™ by Corel Systems
Corporation ©. It was printed on the IBM 3820 Page Printer.

P1071, Third Edition (April, 1994)

This edition applies to product OS/2 Version 2.1 and to all subsequent releases until otherwise indicated in new editions. Changes are continually made to the information herein. Before using these student notes in connection with the operation of IBM systems, consult the latest IBM publications that are available and current.

© Copyright International Business Machines Corporation 1994

reface

The *Restricted Materials of IBM*, if any, included in this document are made available subject to the terms and conditions of either:

- The Agreement for IBM Licensed Programs, or
- The IBM Agreement for Education Courses with Restricted Materials as stated in your country's IBM Education Catalog.

The terms and conditions of these contracts covering education materials are summarized as follows:

The customer shall use these Restricted Materials furnished by IBM solely for the purposes of the IBM education class in which they are provided. The Customer agrees not to copy any Restricted Materials. In the event the Customer has a license, under the terms and conditions of the Agreement for IBM Licensed Programs, for an IBM licensed program to which the provisions concerning Restricted Materials apply, the Customer may additionally utilize such restricted Materials solely for the following purposes:

1. making modification to the Customer's products and/or programs so that they will function with the licensed programs to which the Restricted Materials apply;
2. making modifications, subject to the provisions of the section entitled "Permission to Modify" of the Agreement for IBM Licensed Programs, to the licensed programs to which the Restricted Materials apply; and/or;
3. assisting the Customer in problem determination, problem source identification and/or problem resolution activities associated with the use of the licensed programs to which the Restricted Materials apply.

The Customer will not provide or otherwise make available any Restricted Materials, furnished by IBM under the Agreement for IBM Licensed Programs or the IBM Education Agreement for Restricted Materials as provided in your country's IBM Education Catalog, in any form without IBM's prior written consent except to the Customer's employees or IBM employees, or to other persons during the period such other persons are on the Customer's premises, for purposes specifically related to the Customer's authorized use of the IBM Licensed Program to which such Restricted Materials apply.

Trademarks

The following are trademarks of the International Business Machines Corporation:

- IBM
- Micro Channel
- Multimedia Presentation Manager/2 (MMPM/2)
- Operating System/2 (OS/2)
- Personal System/2 (PS/2)
- Presentation Manager (PM)
- Systems Application Architecture (SAA)

Bibliography

- *Books and Manuals:*

ISBN 1-55755-184-7	OS/2 2.1 Complete
ISBN 1-56205-206-3	Inside OS/2 2.1
ISBN 0-672-30240-3	OS/2 2.1 Unleashed
ISBN 0-672-30317-5	OS/2 2.1 Consultant
ISBN 0-442-01833-9	Client/Server programming with OS/2 2.1
ISBN 0-442-01522-4	OS/2 2.X Notebook
S10G-6268-00	Procedures Language 2/REXX Reference
S10G-6269-00	Procedures Language 2/REXX User's Guide
96F8488	IBM Network Transport Services/2 Redirected Installation and Configuration Guide
96F8489	LAN Adapter and Protocol Support Configuration Guide
G362-0002-02	OS/2 Application Solutions

- *Technical Bulletins:*

GG24-3948-00	OS/2 2.1 Technical Update
GG24-3730	OS/2 Version 2 Volume 1: Control Program
GG24-3731	OS/2 Version 2 Volume 2: DOS and Windows Environment
GG24-3732	OS/2 Version 2 Volume 3: Presentation Manager
GG24-3774	OS/2 Version 2 Volume 4: Writing Applications
GG24-3775	OS/2 Version 2 Volume 5: Print Subsystem
GG24-3948	OS/2 Version 2.1 Technical Update
GG24-3780	OS/2 Version 2.0 Remote Installation and Maintenance
GG24-3783	Automated Installation for CID Enabled OS/2 V2.0

Related education

OS/2 User

- P1069 OS/2 2.1 User workshop (1.0 days)
- P1069F Atelier OS/2 version 2.1 (1.0 days)
- P1070 Using and Customizing OS/2 2.1 (2.0 days)
- P1070F Utilisation et personnalisation d'OS/2 version 2.1 (2.0 days)

OS/2 Support

- P1071 Installing and Supporting OS/2 2.1 (3.0 days)
- P1071F Installation et soutien d'OS/2 Version 2.1 (3.0 days)
- P1072 OS/2 2.1 Advanced Support Workshop (2.0 days)
- P1081 OS/2 2.1 Performance and Tuning Workshop (2.0 days)
- P1082 OS/2 Problem determination (4.5 days)
- P1066 REXX programming for OS/2 (3.0 days)
- P1066F Programmation REXX pour OS/2 (3.0 days)

OS/2 Programming

- S7027 Introduction to OS/2 Presentation Manager programming (5.0 days)
- S7027F Introduction la programmation pour le gestionnaire de Presentation d'OS/2 (5.0 days)
- S7031 Advanced topics in OS/2 Presentation Manager programming (5.0 days)
- S7031F Programmation pour le gestionnaire de Presentation d'OS/2 (5.0 days)
- S7055 DB2/2 Application programming in C (3.0 days)
- S7056 C Programming for OS/2 (5.0 days)
- S7056F Programmation en C pour OS/2 (5.0 days)
- S7059 Introduction to OS/2 2.0 System Object Module and Workplace Shell programming (3.0 days)
- P1066 REXX programming for OS/2 (3.0 days)
- P1066F Programmation REXX pour OS/2 (3.0 days)

DB2/2 for OS/2

- S7052 DB2/2 Query Manager workshop (2.0 days)
- S7052F Atelier sur le gestionnaire de requites DB2/2 (2.0 days)
- S7053 DB2/2 Administration Workshop (4.0 days)
- S7054 DB2/2 Performance Workshop (4.0 days)
- S7055 DB2/2 Application programming in C (3.0 days)
- S6311 SQL Workshop (2.0 days)
- S6966 Distributed relational database planning (2.0 days)
- S9139 Distributed database workshop, OS/2 to DB2 (4.0 days)

Communications Manager/2

- S7049 OS/2 Communications Manager/2 (4.0 days)

- S7049F Gestionnaire de communications/2 (4.0 days)

OS/2 LAN Server

- G3795 Introduction to integrated networking (1.0 days)
- G4360 OS/2 LAN Server administration workshop (3.0 days)
- G4370 OS/2 LAN Server workshop planning and installation (2.0 days)
- G4560 OS/2 LAN Server administration workshop II (4.0 days)
- G4570 OS/2 LAN Server and Netware Client coexistence (2.0 days)
- G4760 OS/2 LAN Server performance workshop (4.0 days)

TOPIC 1: Introducing OS/2

Topic objective:

Terminal objective:

After attending this topic the student should have a general understanding of the main features of OS/2.

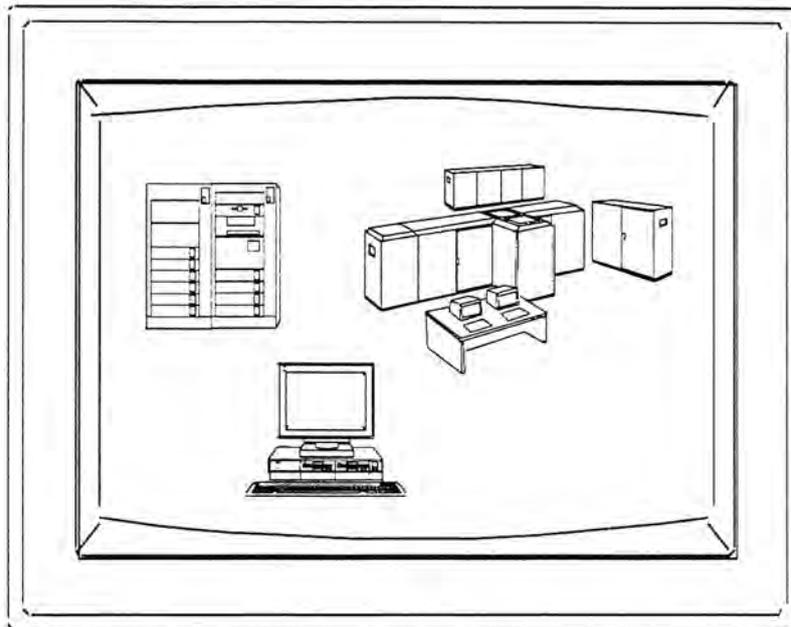
Enabling objectives:

Upon completion of this topic the student should be able to recognize the following as features of OS/2:

- Multitasking
- Memory Capabilities
- Protection Mechanism
- Graphical Interface
- Application Support
- Database Manager
- Communication Capabilities



Evolution of the PC



PS511000

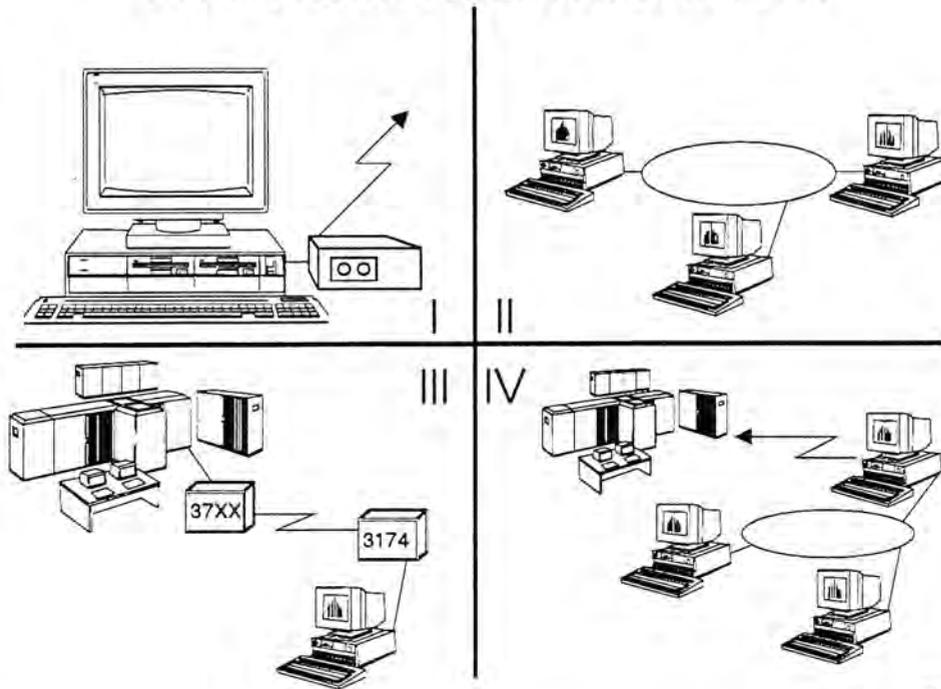
Figure 1-1. PC Evolution

The employment of Personal Computers is at a turning point. PCs are no longer confined to support and enhance personal productivity only, but are seen as windows to company-wide resource access and mission critical applications. In order to deal with all these aspects, PCs have to fulfill several requirements:

- Industrial strength reliability
- Comprehensive network support
- Ease of use
- Investment protection and exploitation
- Performance in terms of overall throughput and concurrency
- Costs should not exceed the benefits

It is important to state that all these requirements are to be satisfied by the operating environment.

Workstation Environment



PS511002

Figure 1-2. Workstation Environments

- **Standalone Workstation**
Personal productivity and standalone versions of small business, department or special purpose applications, no LAN attachment, eventually asynchronous communication to external host database services.
- **Enterprise Workstation**
No LAN attachment, Controller based connection to a midrange or mainframe host, local personal productivity
- **Workgroup LAN**
LAN based resource sharing, local and distributed personal productivity, mail, small business, department and special purpose applications, Server based systems management.
- **Enterprise LAN**
Large LAN networks, Host communications, Workgroup applications and distributed host applications, Host controlled system and network management.

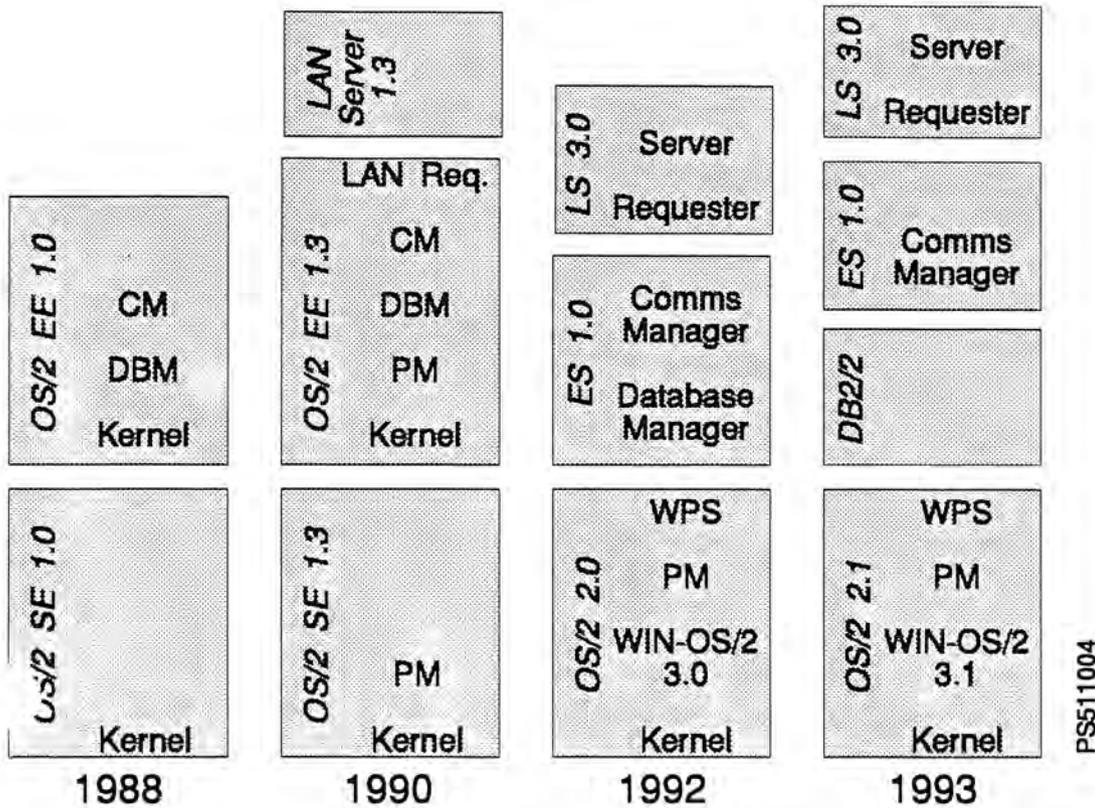


Figure 1-3. OS/2's Evolution

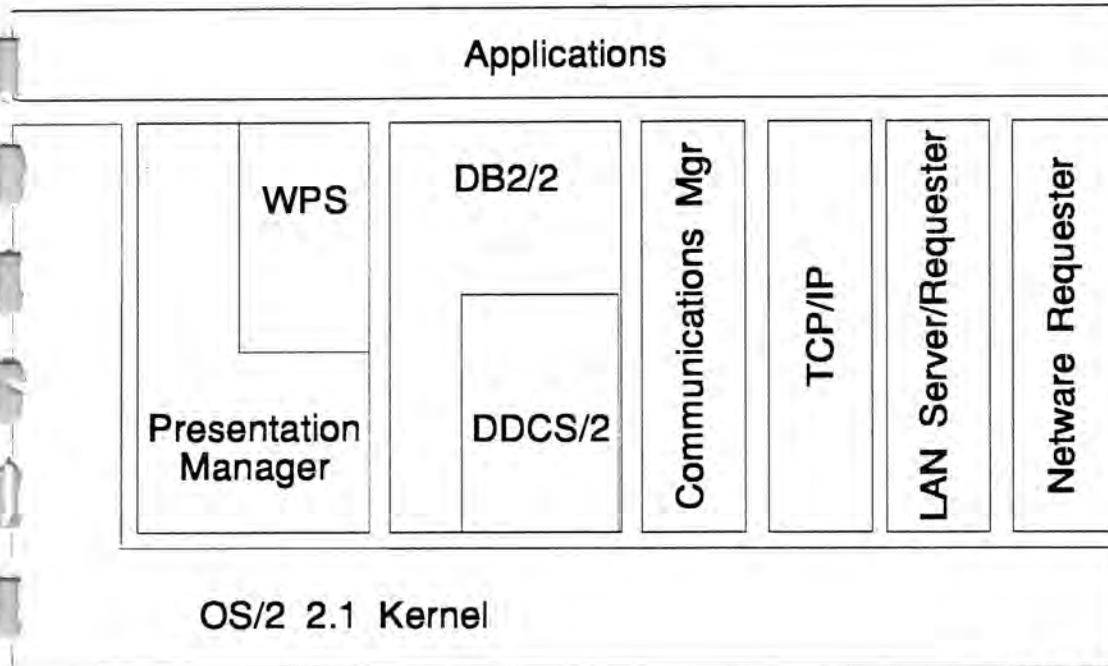
In the mid-1980's the need for an advanced operating environment that was capable of exploiting the functions of the present processors was obvious. IBM met these demands by a continuous development from a first character-based OS/2 version in 1987/88 up to the integrating platform of OS/2 2.1 in 1993.

The major features of OS/2 2.x consist in its being based on the 80386 micro-processor which provides function in the areas of:

- Utilization of the flat memory model
- Exploitation of the micro channel architecture
- Multiple virtual DOS machines
- The Workplace Shell
- Compatibility

NTS - network transport services.

OS/2 Components

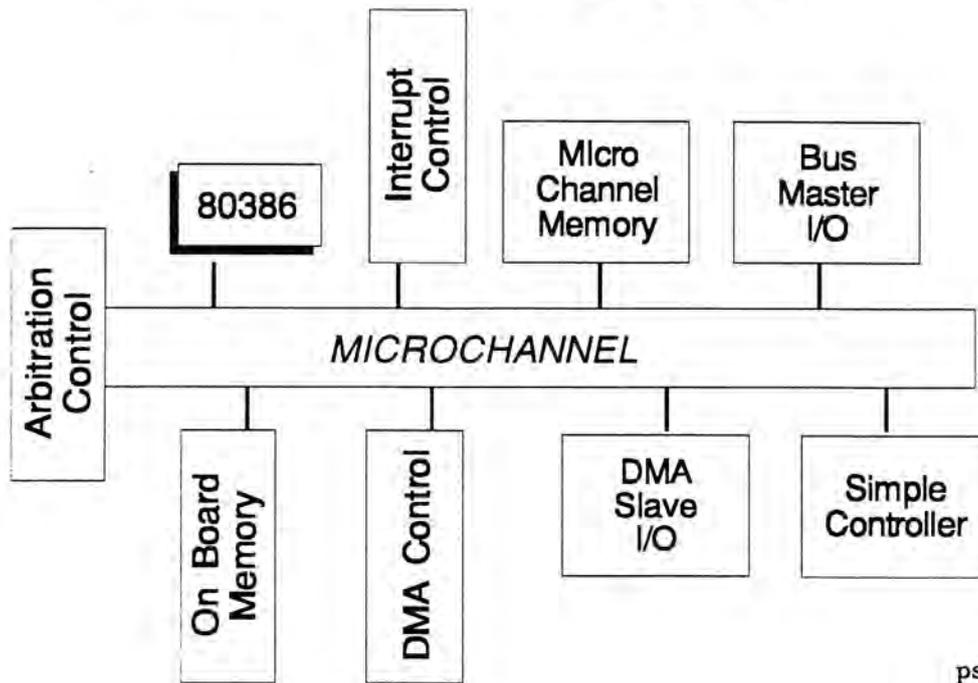


PS511006

Figure 1-4. OS/2 Components

OS/2 2.1 enhances the performance of the Workplace Shell. In order to improve productivity without purchasing additional software, a variety of mini-applications and tools, applets, are provided. Included in the base package are also some games for new users to become familiar with the new environment.

Micro Channel Support



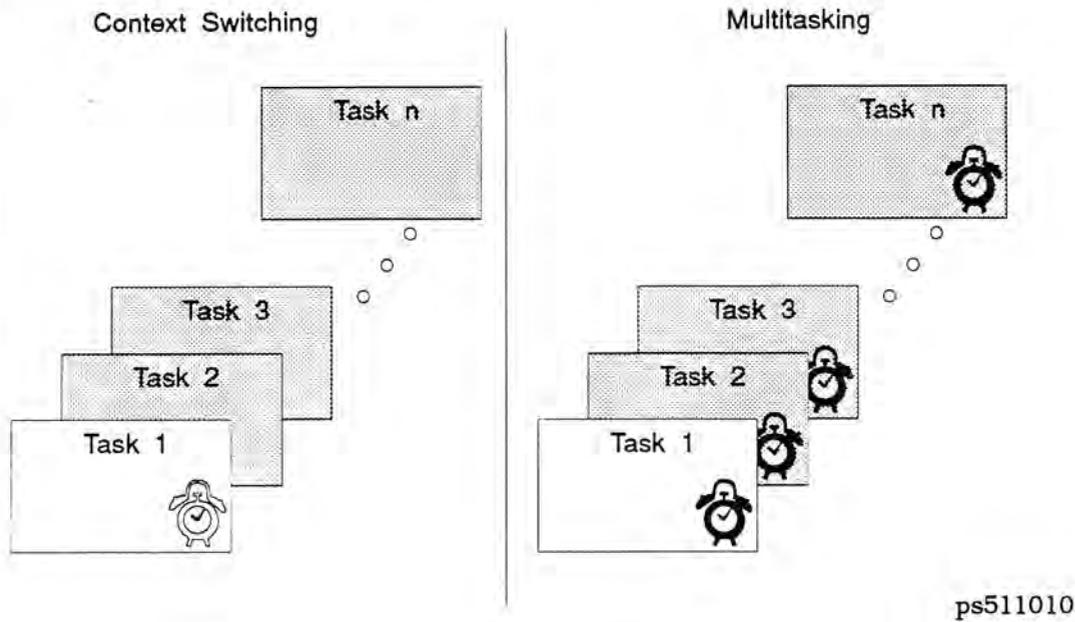
ps511008

Figure 1-5. Micro Channel Support

The objective of PS/2 Micro Channel architecture is I/O and processor simultaneity. Unless the operating system is designed to make use of it, however, the hardware capabilities are useless.

OS/2 2.1 has the ability to take advantage of IBM's Micro Channel architecture and advanced, intelligent adapters while supporting a variety OEM hardware. The effect of this use of the Micro Channel is that it relieves the 80386 Microprocessor of most disk related I/O activities.

OS/2's Multitasking

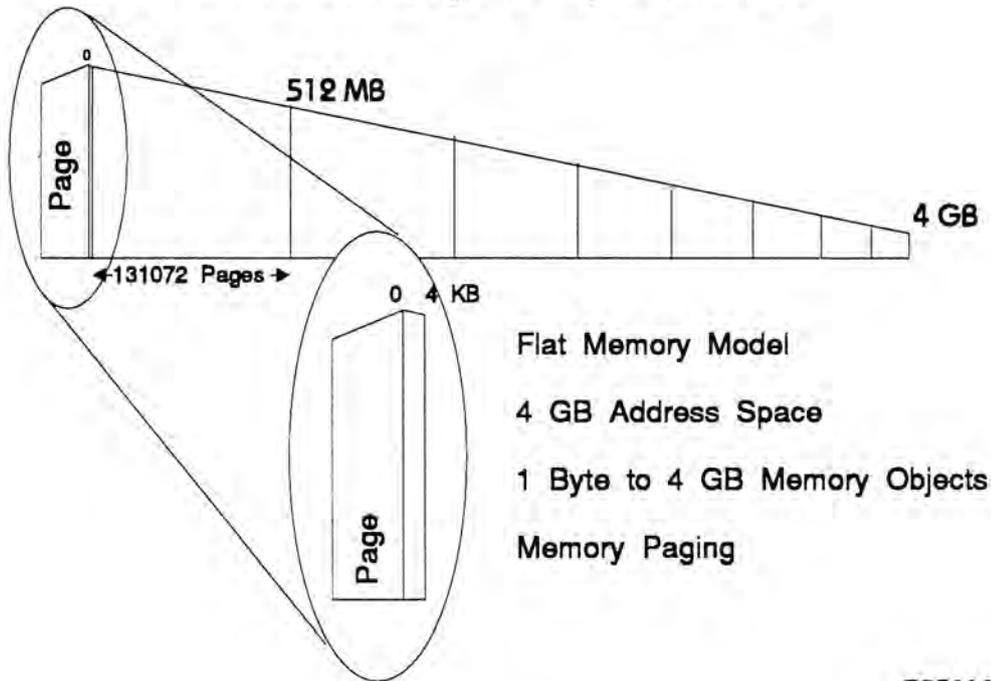


ps511010

Figure 1-6. OS/2's Multitasking

OS/2 allows the user to run multiple concurrent applications, but there is only one CPU and that CPU can run only one task at a time. The CPU handles this by dividing the time between multiple processes to make it appear to the user that those processes are running simultaneously. Note the difference between "context switching", where applications lie dormant until given the user's focus and "multitasking", where applications continue to execute when in the background.

OS/2's Memory Capabilities



PS511012

Figure 1-7. OS/2's Memory Capabilities

OS/2 2.1 supports a 32-bit addressing model. Although based on the Intel 386 processor family, it is more a 32-bit system than Intel-specific. A part of the design of OS/2 2.1 was to gain the best platform for future portability.

The 386 processor supports up to 4 GB of linear address space, but for compatibility reasons with 16-bit OS/2 applications, the address space per process is restricted to 512 MB. OS/2 2.1 is therefore able to use more memory than is physically installed in most systems. That's why it allows the use of the disk for additional memory, usually referred to as virtual memory.

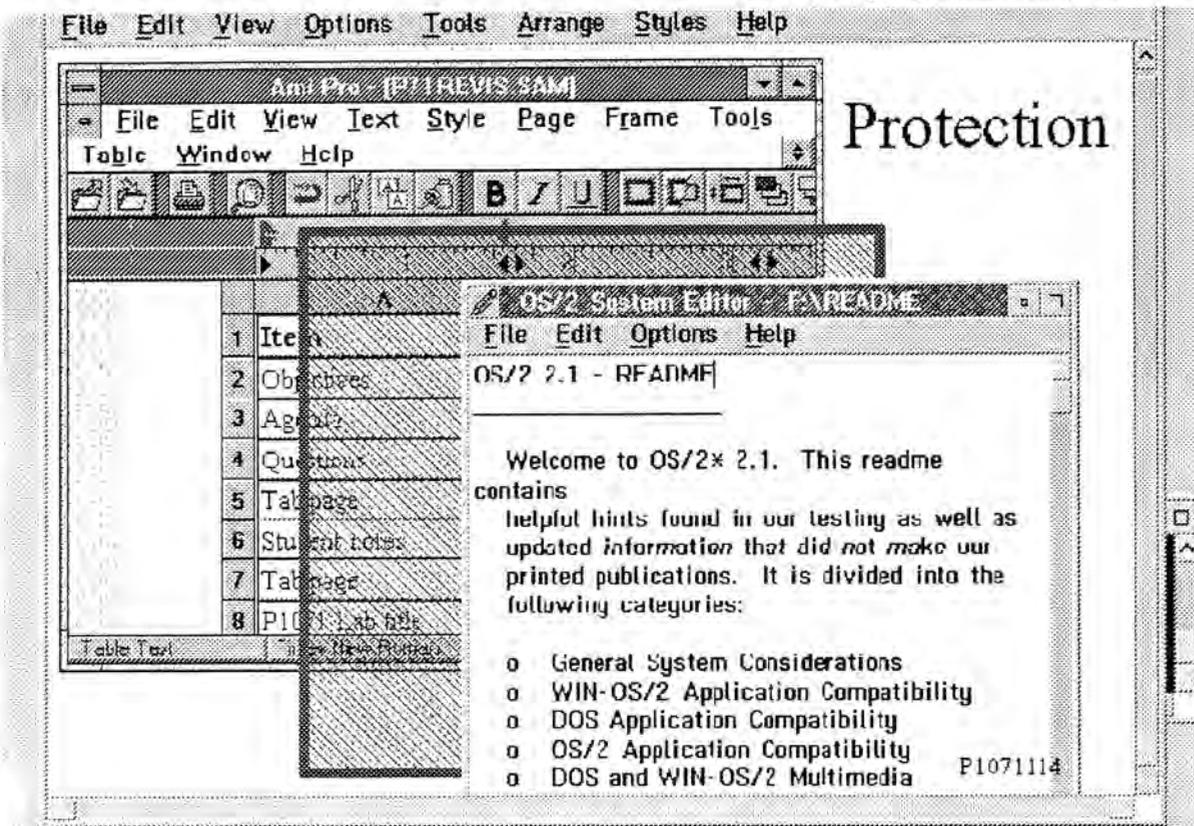


Figure 1-8. OS/2's Protection Mechanism

In a multitasking system it is very important that tasks be protected from each other and that sharing of information be carefully controlled by the operating environment. All process address spaces are therefore kept separate from each other, whether they be DOS, Windows, OS/2 applications or even parts of the OS itself. If a process attempts to use memory outside its own allocated memory space, it will be caught by the operating system.

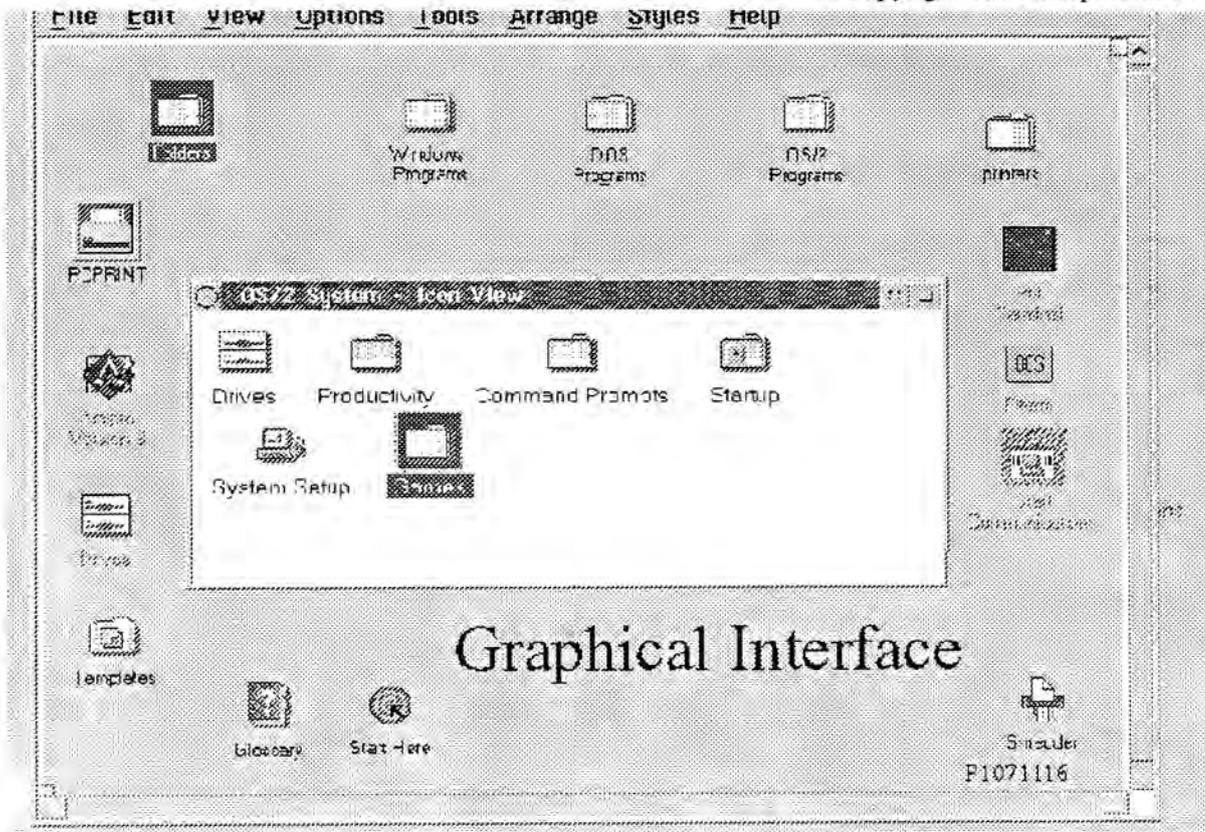
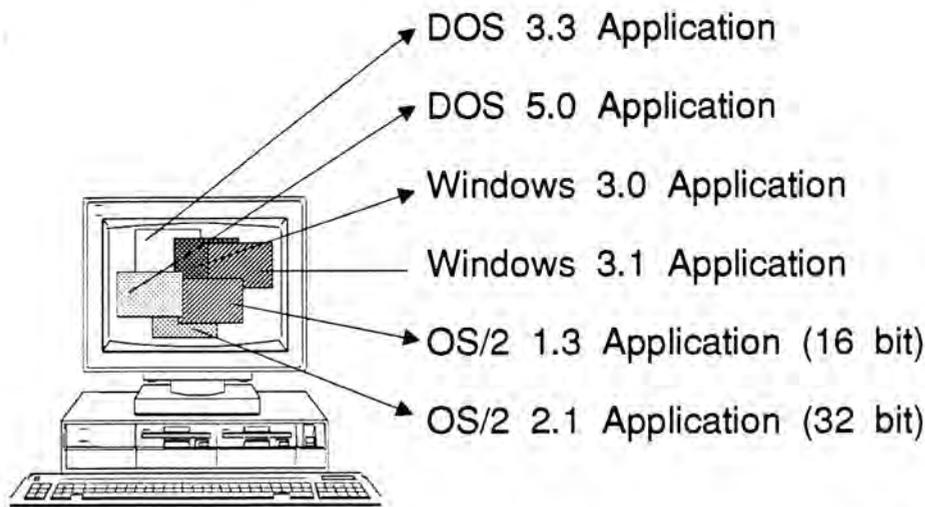


Figure 1-9. OS/2's Graphical Interface

The Workplace Shell is on the one hand the user interface for the operating environment itself and on the other hand it is an environment in which applications can be integrated in order to optimize the cooperation between themselves and OS/2.

The OS/2 Workplace Shell is an object-oriented user interface. A user interacts with icons representing objects such as printers, folders, shredders etc.. The main idea behind this is to allow users to work in a more task-oriented way and therefore to focus on what they want to do and not on how to do it.

OS/2's Application Support



ps511018

Figure 1-10. OS/2's Application Support

OS/2 2.1 provides the widest range of application choice by running OS/2, DOS and Windows 3.X applications. It adds to the existing DOS and Windows applications the benefits of preemptive multitasking, comprehensive memory support and, reliability. OS/2 acts as a base for powerful 32-bit OS/2 applications.

In addition - the whole is greater than the sum of its parts - OS/2, DOS and Windows applications are integrated on the same Desktop. The user has the advantage of making working combinations and is still communicating with only one interface.

OS/2's DB2/2

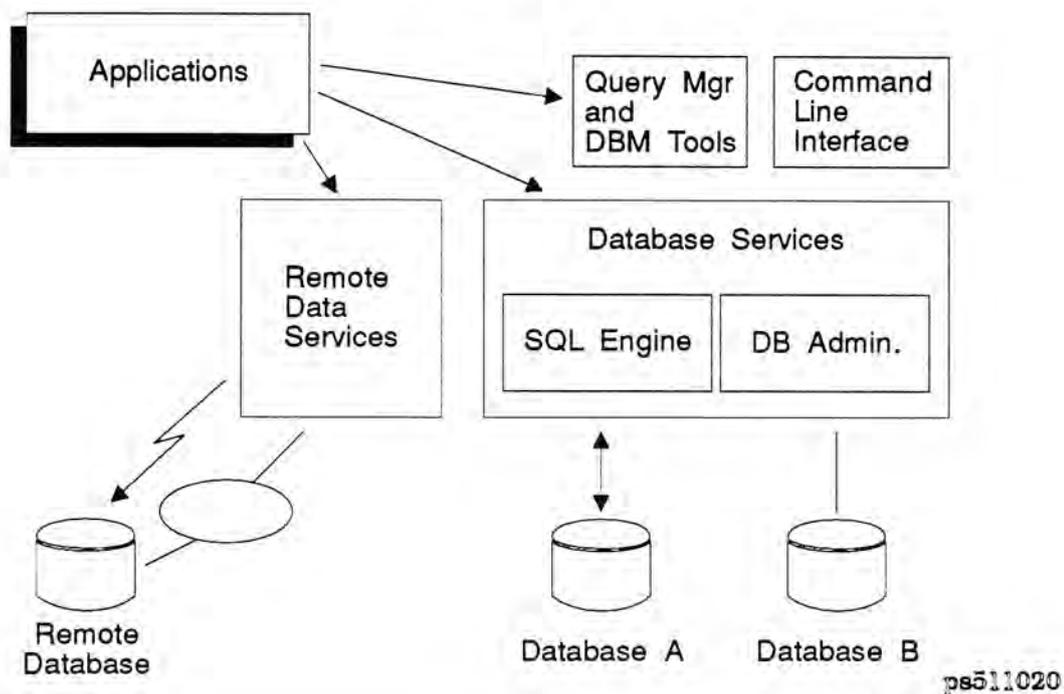


Figure 1-11. OS/2 DB2/2

OS/2's DB2/2 is based on the relational database model with SQL (Structured Query Language) as the common interface. Query Manager is a front-end to DB2/2. It assists users in creating complex queries without the user requiring SQL syntax knowledge. With DDCS/2, access to host databases that conform to the Distributed Relational Database Architecture (DRDA), which includes DB2, SQL/DS and OS/400, is possible.

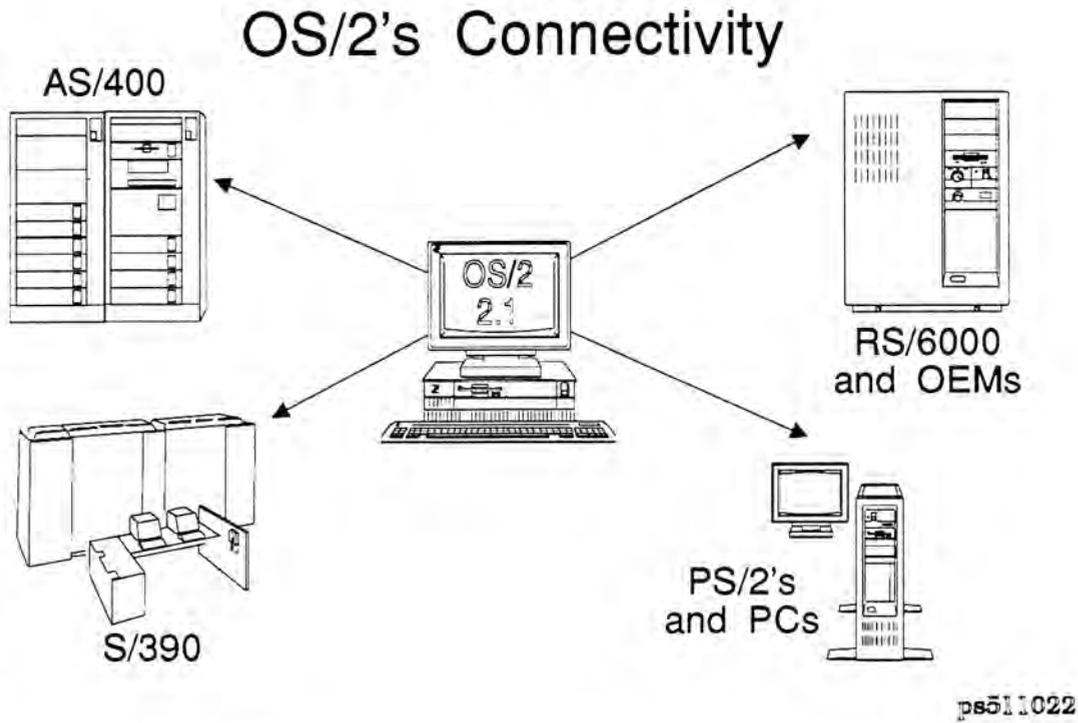


Figure 1-12. OS/2's Connectivity

OS/2 provides a comprehensive communication support to other operating environments via various protocols. As a consequence of its multitasking capabilities, it is excellently equipped for full client/server support to multiple other systems simultaneously.

Topic Summary

In this topic the student was introduced to highlights of OS/2 and some of its' other components:

- Multitasking
- Memory Capabilities
- Protection Mechanism
- Graphical Interface
- Application Support
- Database Manager
- Communication Capabilities

This concludes the first topic
"Introduction."

TOPIC 2: Installing OS/2 Version 2.1

Topic objective:

Terminal objective:

After attending this topic the student should be able to install the OS/2 Version 2.1 operating system such that OS/2 provides the most efficient support for the user's work on that particular system.

Enabling objectives:

Upon completion of this topic the student should be able to plan an OS/2 installation using one of the installation types listed below.

- Basic Installation
- Dual Boot Installation
- Boot Manager Installation
- Selective Installation of Options
- Response File Installation

pl022 - Advanced OS2.

Primary Planning

Subtopic objectives

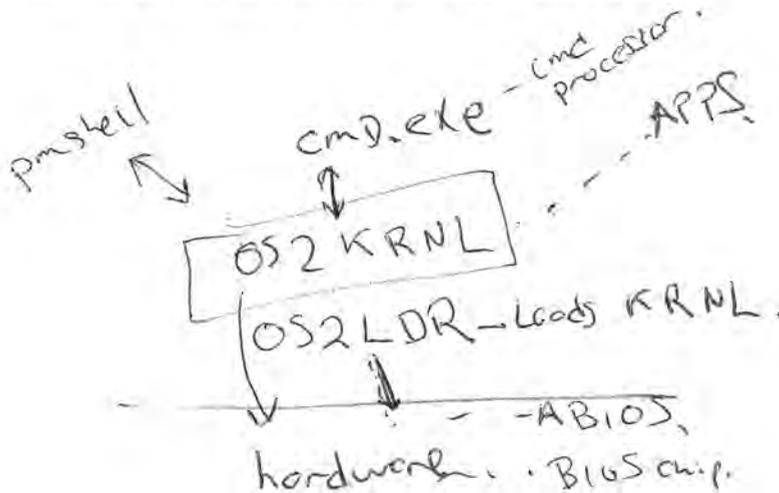
Terminal objective:

After attending this subtopic the student should be able to plan an OS/2 2.1 installation.

Enabling objectives:

After attending this subtopic the student should be able to plan an OS/2 2.1 installation by considering the following:

- Hardware requirements
- Backing-up of existing files
- The different OS/2 2.1 configurations that are possible
- The various media available for the installation of OS/2 2.1



BOOT2K
→ single bootable disk for debugging.

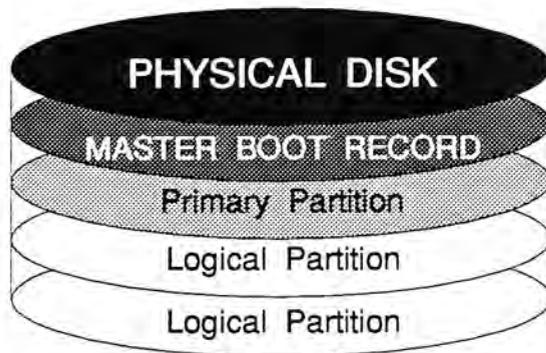
Hardware

- Intel 80386 or higher processor
- 6MB RAM
- Hard disk with 20 to 50 MB free space
- Floppy disk (CD ROM drive would be nice)
- VGA monitor
- Mouse

ps512106

Figure 2-1. Hardware Considerations

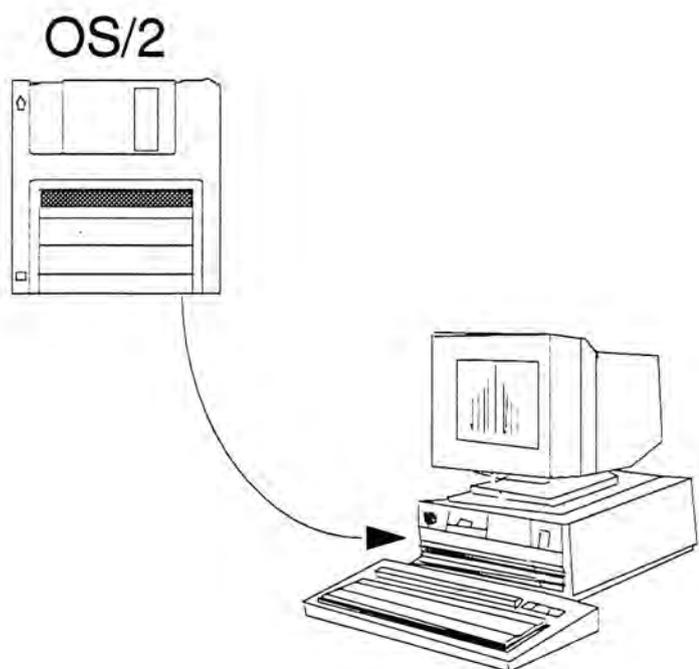
Before you install OS/2, it would be beneficial to plan the installation process. Consider what hardware OS/2 will be running. Installation provides a good opportunity to tune the system.



ps512102

Figure 2-2. Back-up

Depending on your installation options, it may be a good idea to back up your hard disk.



ps512104

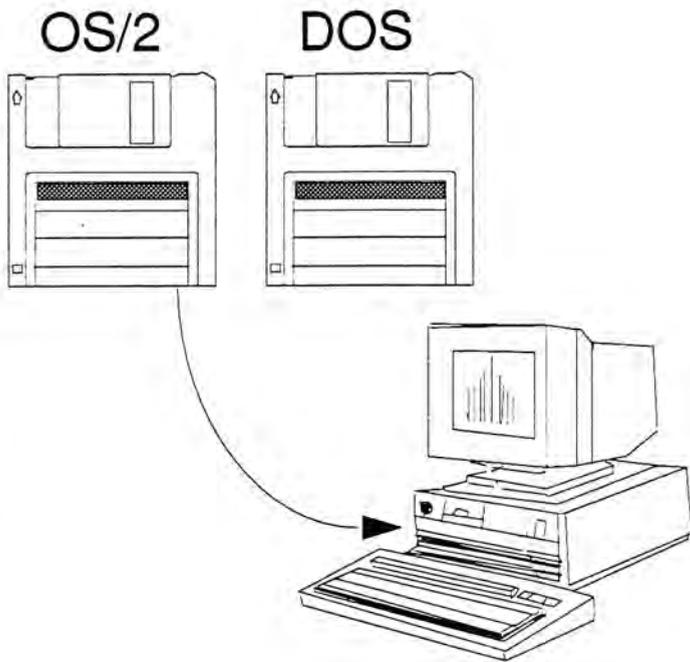
Figure 2-3. Basic System

A basic installation sets up the system so OS/2 is the only operating system on the computer.

Keep in mind that OS/2 Version 2.1 alone can run:

- Existing OS/2 programs,
- DOS programs, and
- Windows programs.

In most cases, the basic installation will be sufficient.

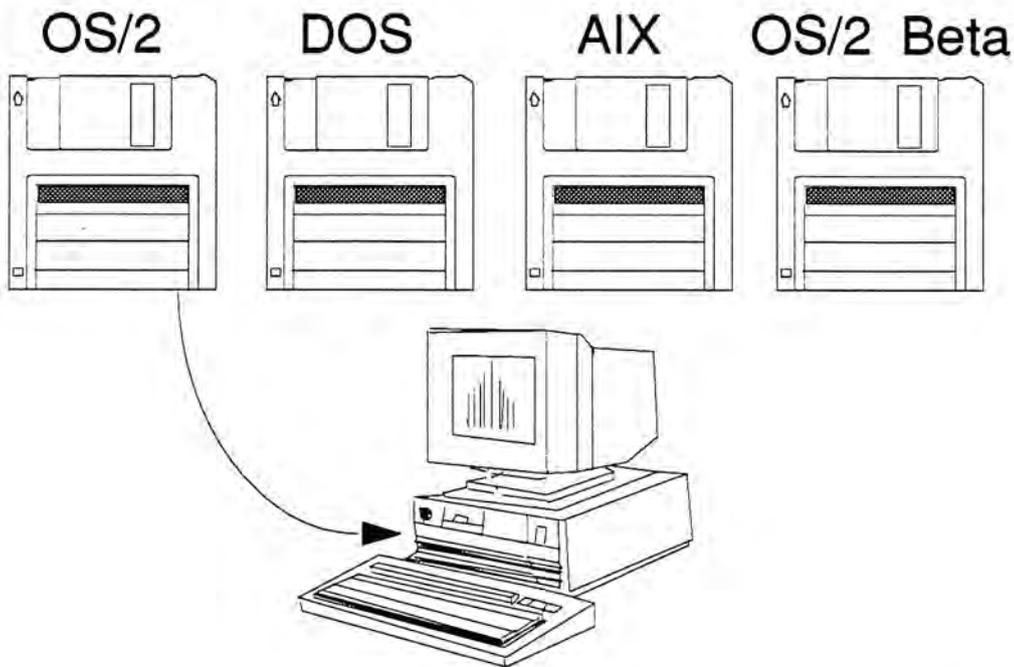


ps512106

Figure 2-4. Dual Boot System

If there is a version of DOS already installed on the hard disk of the computer, OS/2 can be installed on the hard disk along with it so that the dual boot feature can be used. Dual boot permits the user to switch back and forth between OS/2 and DOS.

If the user has DOS programs that will not run under the OS/2 operating system, then he/she can switch to Native DOS and run those programs.

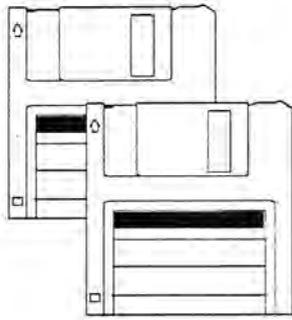


ps512108

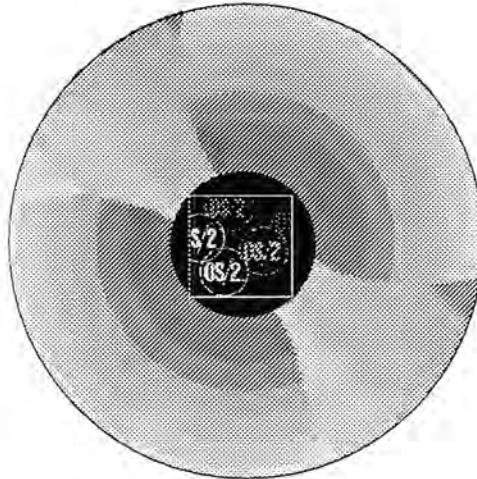
Figure 2-5. Boot Manager Installation

The advanced installation procedure sets up the computer's hard disk for the installation and operation of several operating systems. Boot Manager coordinates the booting of the operating systems. With Boot Manager each operating system is installed in its own disk partition. Each operating system is displayed on a menu and selected by the user at system start-up.

Installation



Diskette 1



ps512110

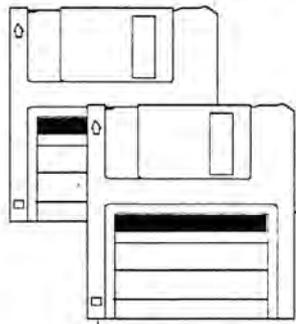
Figure 2-6. CD-ROM Install

As operating systems and applications get bigger, users require ways to shorten the installation time and manufacturers look for more efficient ways to package the files.

During OS/2 2.1 beta testing, IBM introduced the capability to install OS/2 from CD-ROM. At the printing this course, the following were supported CD ROM Devices:

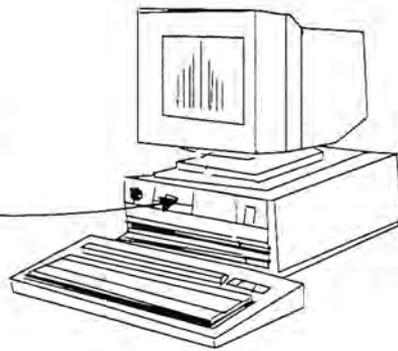
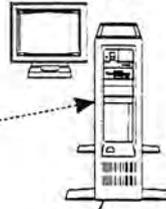
- IBM
- Hitachi
- NEC
- Panasonic
- Sony
- Texel
- Toshiba

Installation Diskette



LT Diskette

Server



ps512112

Figure 2-7. Remote LAN Install

If you are a system administrator or coordinator who must install and configure operating systems on a number of computers, installing OS/2 across a network may be of interest to you.

Topic Summary

In this subtopic the student learned to plan an OS/2 2.1 installation.

The student was taught to consider the following:

- Hardware requirements
- Backing-up of existing files
- The different OS/2 2.1 configurations that are possible
- The various media available for the installation of OS/2 2.1

This concludes the subtopic of the topic
"Installing OS/2 Version 2.1".

Installation

Subtopic objectives

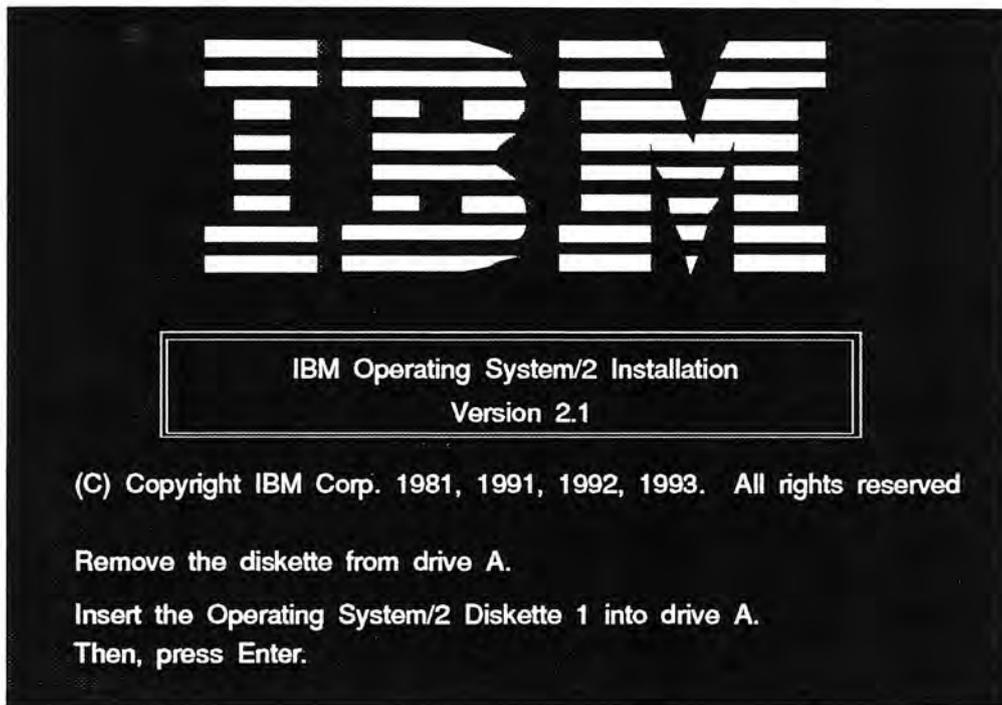
Terminal objective:

After attending this subtopic the student should be able to install OS/2 Version 2.1 as the ONLY operating system on the computer.

Enabling objectives:

After attending this subtopic the student should be able to:

- Begin the installation of OS/2 Version 2.1.
- Select the installation options desired for your system.
- Follow the instructions on the screen to complete the installation procedure.



ps512200

Figure 2-8. The First Screen

This subtopic describes how to install OS/2 as the ONLY operating system on your computer. The basic installation described in this topic can be used to install OS/2 on a new system or on a system that already contains data.

If you have OS/2 Version 2.0 on your system, the basic installation will replace it with OS/2 Version 2.1. Similarly, if you have DOS on your system, the basic installation can replace it with OS/2 Version 2.1.

To begin the installation, insert the installation diskette into drive A and turn on the computer. If it is on, press and hold CTRL + ALT and press DEL to restart the system.

The screen shown above appears.

Throughout the installation, you will be asked to remove the diskette in drive A and insert another diskette. Leave the diskette in drive A until the instructions direct you to replace it.

You will replace diskettes several times before completing the installation. You should have at least 13 operating system diskettes not including the

installation diskette. During installation, the installation diskette will be requested more than once. During installation, diskette number 1 may be requested more than once.

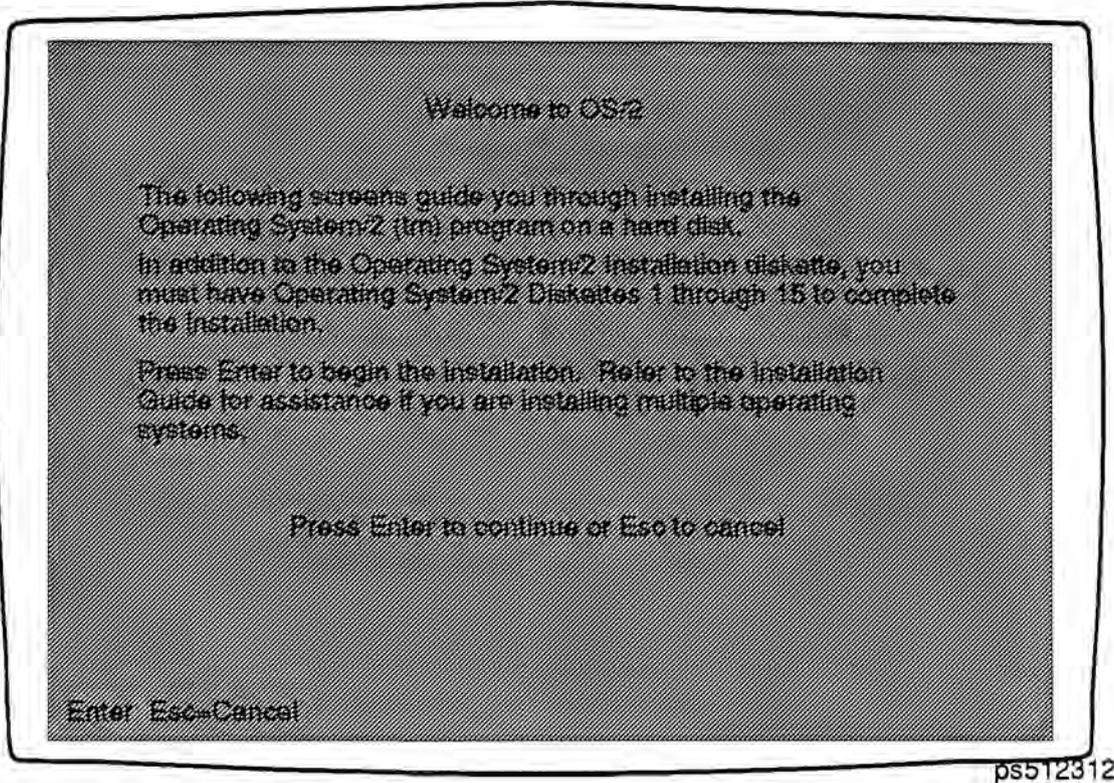


Figure 2-9. Welcome to OS/2

To be able to start OS/2 from diskette, you have to insert the installation diskette first and then diskette 1.

On this panel, where you are welcomed to OS/2, notice the ESC = Cancel option. It will provide you with a command prompt.

From that command prompt you can start OS/2 full screen applications or OS/2 commands, and have full access to files on HPFS drives.

Installation Drive Selection

If you want multiple versions of DOS, OS/2 or other operating systems on the same hard disk, refer to the Installation Guide for information on OS/2 Hard Disk Management before continuing.

If you have multiple partitions set up on your hard disk, select option 2 to verify that the correct partition is active.

OS/2 will be installed on drive C:

Select an option:

1. Accept the drive

2. Specify a different drive or partition

If you select option 2, the FDISK screen is displayed.

Enter Esc=Cancel F3=Exit F1=Help

ps512202

Figure 2-10. Drive Selection

When the above screen appears, you are being asked whether you want to accept the default installation partition as indicated on the panel. If you are installing on a system that already contains partitions, the installation program indicates which partition it will use to install OS/2 2.1:

OS/2 will be installed on drive C:

A partition is an area on the hard disk of a fixed size. The size and number of partitions is dependent on the size of your hard disk. You can choose not to break your hard disk into separate logical units (partitions). In that event, the installation program installs the operating system in one partition that takes up the entire hard disk.

To accept the default, press ENTER. Creating new or changing existing partitions will be covered in the subtopic **Boot Manager**.

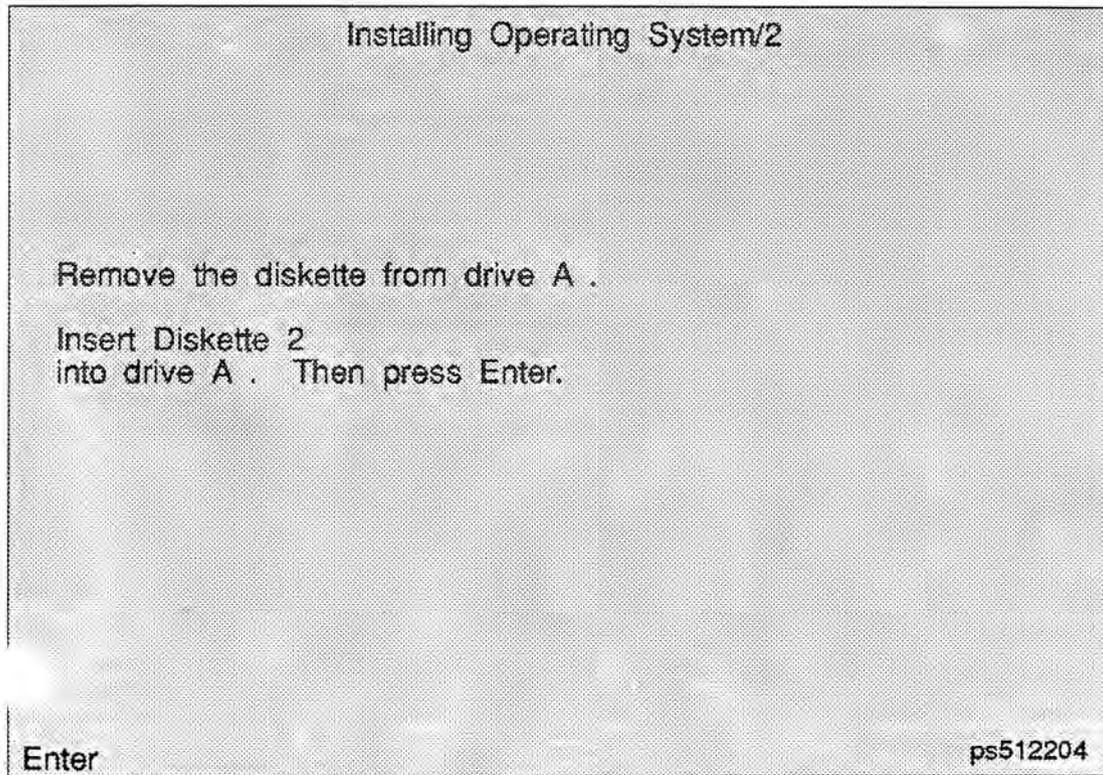


Figure 2-11. Change Diskettes (Full Screen)

The panel instructing you to change the diskette will be a full screen text panel like the one above. The same kind of panel will instruct you when it is time to insert diskettes.

Formatting the Installation Partition

The partitions on a hard disk must be formatted before information can be placed on them. If the partition in which you are going to install the OS/2 operating system has been formatted by DOS or the OS/2 operating system, it is not necessary to format it again.

The installation partition can be formatted to use either the High Performance File System (HPFS) or the FAT file system.

Formatting erases all files. If you need these files, use the **BACKUP** command from your existing operating system to back them up.

Select an option.

1. Do not format the partition

2. Format the partition

Enter F1=Help

ps512206

Figure 2-12. Format the Hard Disk

At this point during the installation, you must decide whether or not to format your disk. When a disk is formatted, its track and sector control information is rewritten.

Formatting is required on a **new** partition because it has no track and sector control information.

If you did not change the partitions and the drive on which you are installing has been previously formatted, it is not necessary to format the disk.

Select the File System

A file manages the information on a partition. The operating system provides two file systems. You must select one for the partition in which you install the OS/2 operating system.

If you have other partitions on your hard disk, you can format them to use either file system after you complete the installation process. Files can be copied between partitions that use different file systems.

Select an option.

1. High Performance File System

2. Fat file system

Enter ESC=Cancel F1=Help

ps512208

Figure 2-13. File System

If you should decide to format your hard disk partition, then you have another choice to make. OS/2 provides a choice of two file systems that may be installed on a partition:

- File Allocation Table File System (FAT)
- High Performance File System (HPFS)

There can only be one file system per partition. You must choose one for the partition on which OS/2 will be installed. A good rule of thumb is to use the HPFS on data disk partitions greater than 100MB. The added performance on a hard disk of lesser size is negligible compared to the FAT system.

The OS/2 operating system can read or write HPFS or FAT files. DOS, on the other hand, can read and write FAT files. This is important to keep in mind if you should decide to configure a dual boot system. This issue will be discussed in greater detail in the subtopic **Dual Boot Installation**.

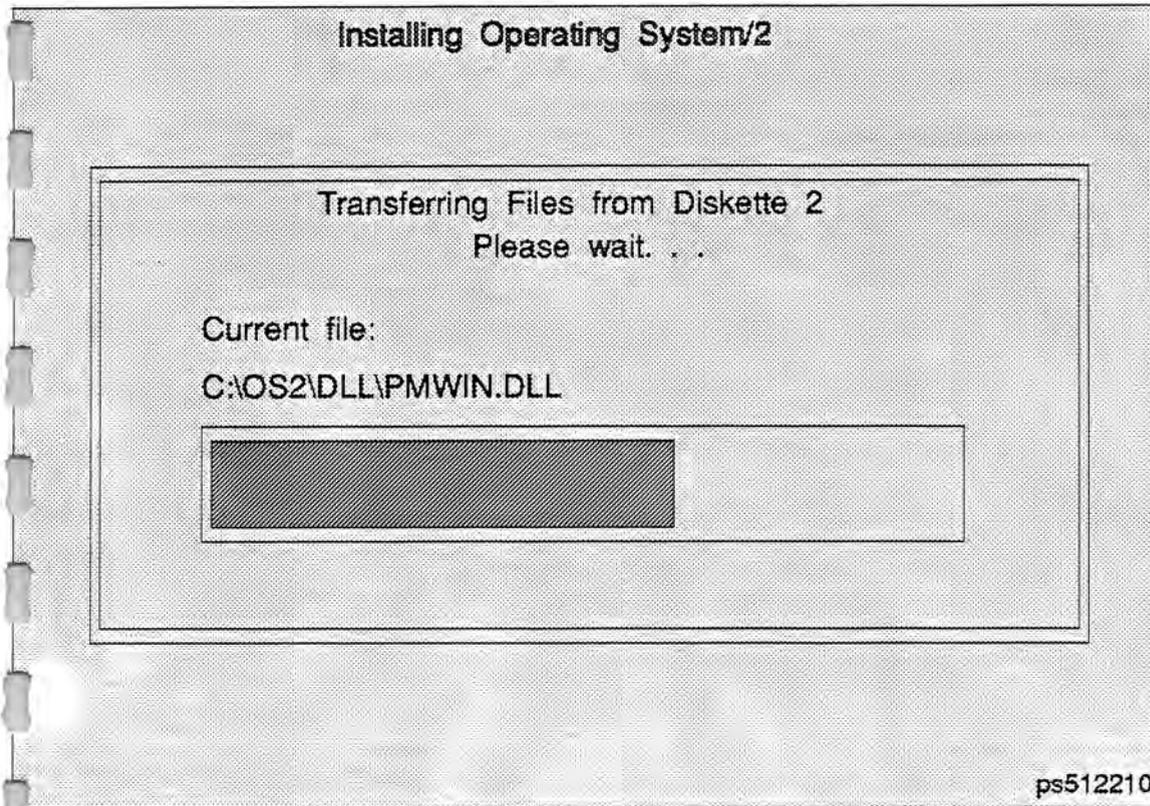
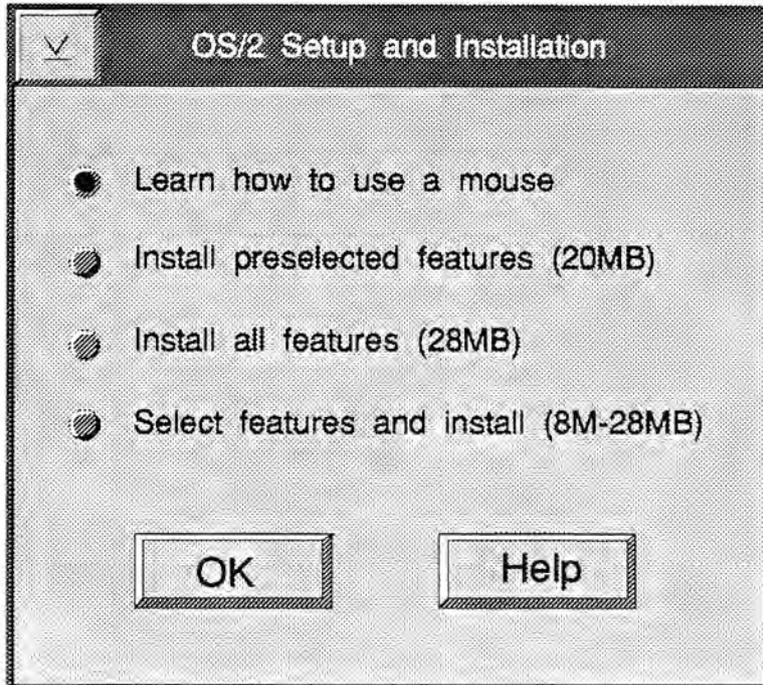


Figure 2-14. Transferring Files

When the installation program begins transferring files to the hard disk, a panel that shows the target file and a progress indicator will be displayed.

If you chose to format the partition, a panel similar to the one above will be displayed which indicates that the partition is being formatted. Once formatting is complete, the above panel will be shown.

Once all the necessary files are transferred to the target location from the diskette, a similar panel will prompt you to replace it with the next diskette. Follow the instructions.



ps512212

Figure 2-15. Setup and Installation Dialog

After the system has been restarted, the OS/2 Setup and Installation screen appears. This panel permits you to make choices regarding which features to install and learn about using the mouse.

1. Learn how to use a mouse
2. Install preselected features
3. Install all features
4. Select features and Install

To select a choice:

- Use the Down Arrow key to highlight the choice, and then press ENTER.
- Move the mouse pointer until it is on the choice, and double click mouse button one twice.

By default, mouse button one will be the first button on the left side of the mouse or directly beneath your right forefinger. Mouse buttons one and two can be swapped later if this is desired.

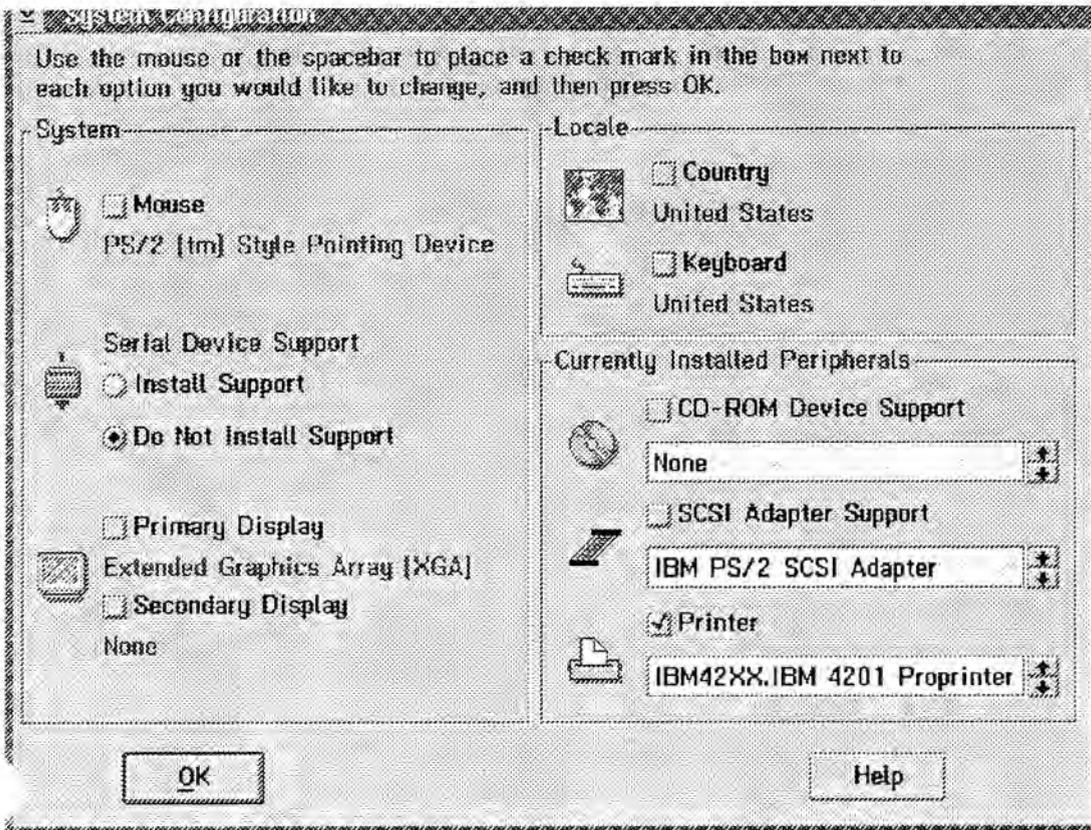


Figure 2-16. System Configuration

If Selective Install is started, the System Configuration screen is displayed.

By selecting check - boxes on this screen and then clicking on the *OK*, push button, further dialogs are displayed

- These dialogs can be used to specify installation details for:
 - SCSI adapter support
 - SCSI based CD-ROM devices
 - Printer support
 - Mouse
 - The installation procedure automatically detects the type of mouse currently attached to the system (if any) and sets the default option accordingly.
 - Keyboard
 - The installation procedure automatically detects the type of keyboard currently attached to the system and sets the default option accordingly. When you select a keyboard, you indicate to the operating system which character to expect when you press a key. The keytop character arrangement varies from country to country.

- Country
 - The default settings are United States.
 - Primary code page of 437 is used. Code page of 850 is used as the secondary code page. These are used for United States and Canada.
- Display

If you have two displays attached to your system, choose the display with the highest resolution as your primary display; this one will be used for displaying the Workplace Shell. Select the display with the lowest resolution as your secondary display; this one, will be used for displaying the OS/2 full screen sessions.
- When all the options are specified, click on *OK* to display the **OS/2 Setup and Installation** screen.

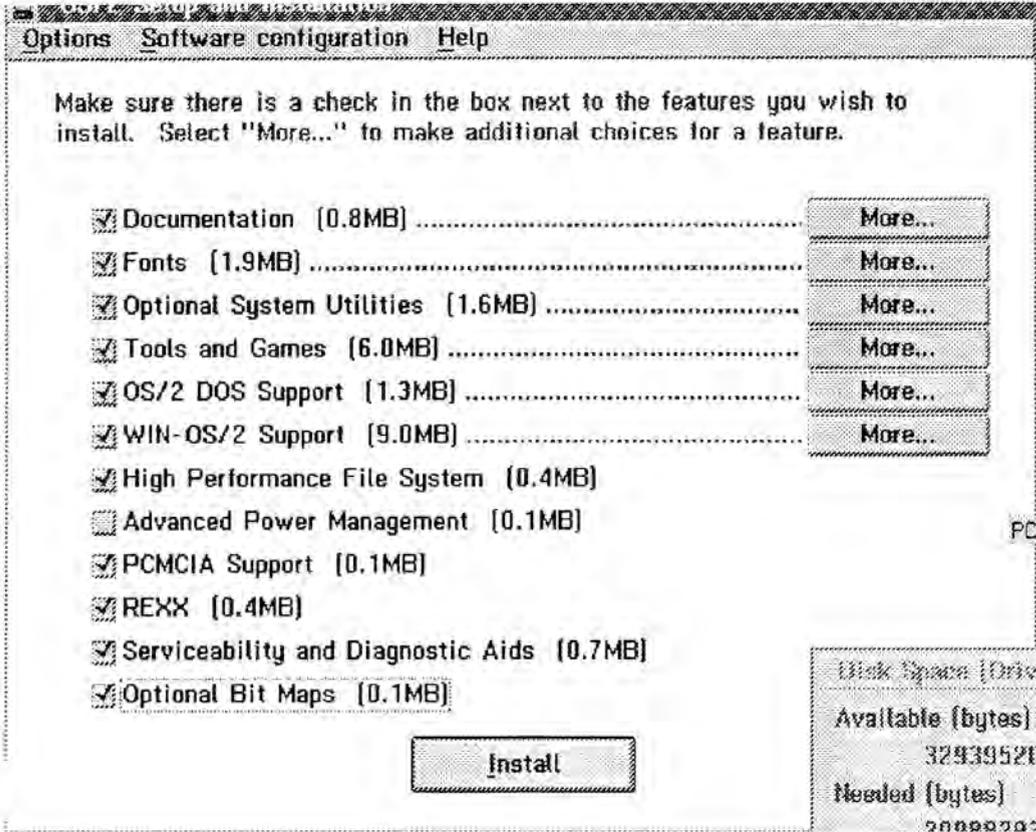


Figure 2-17. OS/2 Setup and Installation Panel

If you chose **Select features and install**, a screen is presented on which you indicate which features you do **not** want to install. When the screen first appears, all of the items have been selected with a in the box next to the feature. To remove the from the feature (keep the feature from being installed), do ONE of the following:

- Move the mouse pointer to the feature and click once with the mouse.
- Move the cursor to the feature and press the Spacebar.

Some features have a **More** push button displayed next to them.

Select the **More** push button to fine tune the selections for that feature.

For example, if you decide you want to install only some of the *documentation*:

1. Select the **More** push button to the right of **Documentation** to display a list of the documentation available.
2. Remove the that is next to any documentation you do not want to install.

Here is a short explanation of the available options to select;

- *Documentation*

Use this to indicate which of the following informational units you want installed on your system.

MORE offers to select either OS/2 Tutorial and/or OS/2 Command Reference and/or REXX Information.

- **Fonts**

Select the fonts you want installed on the hard disk. If no fonts are selected then, only the System proportional, Helvetica fonts and, Courier fonts will be available on your system.

- **Optional System Utilities**

This provides a full set of system utility programs. Some of the utilities are; Installation Aid, Backup, Restore, Label, etc..

- **Tools and Games**

These are tools and games that are provided as productivity aids as well as games.

- **OS/2 DOS Support**

Enables DOS programs to run on the OS/2 operating system. If you select this option, you will be prompted, at a later time, to indicate which of your existing DOS applications you want to migrate to OS/2.

- **WIN-OS2 Support**

Enables MicroSoft windows programs to run on the OS/2 operating system. If you select this option, you will be prompted, at a later time, to indicate which of your existing Windows applications you want to migrate to OS/2.

- **High Performance File System**

Provides fast access to large disk volumes.

- **Advanced Power Management**

Select this option if your computer has Advanced Power Management capabilities. (During initial installation of the operating system, this choice will be checked if the installation program detects this capability.

- **PCMCIA Support**

Select this option if you have a computer that has a PC Memory Card International Association Adapter.

- **REXX**

Use this option to install the REXX OS/2 procedures language on your system. This batch language can be used to develop Systems Application Architecture programs.

- **Serviceability and Diagnostic Aids**

Select this to enable your system to gather information that can be used to isolate and correct system problems.

- **Optional Bit Maps**

Provides a set of bitmaps that you can use to change the background of your system.

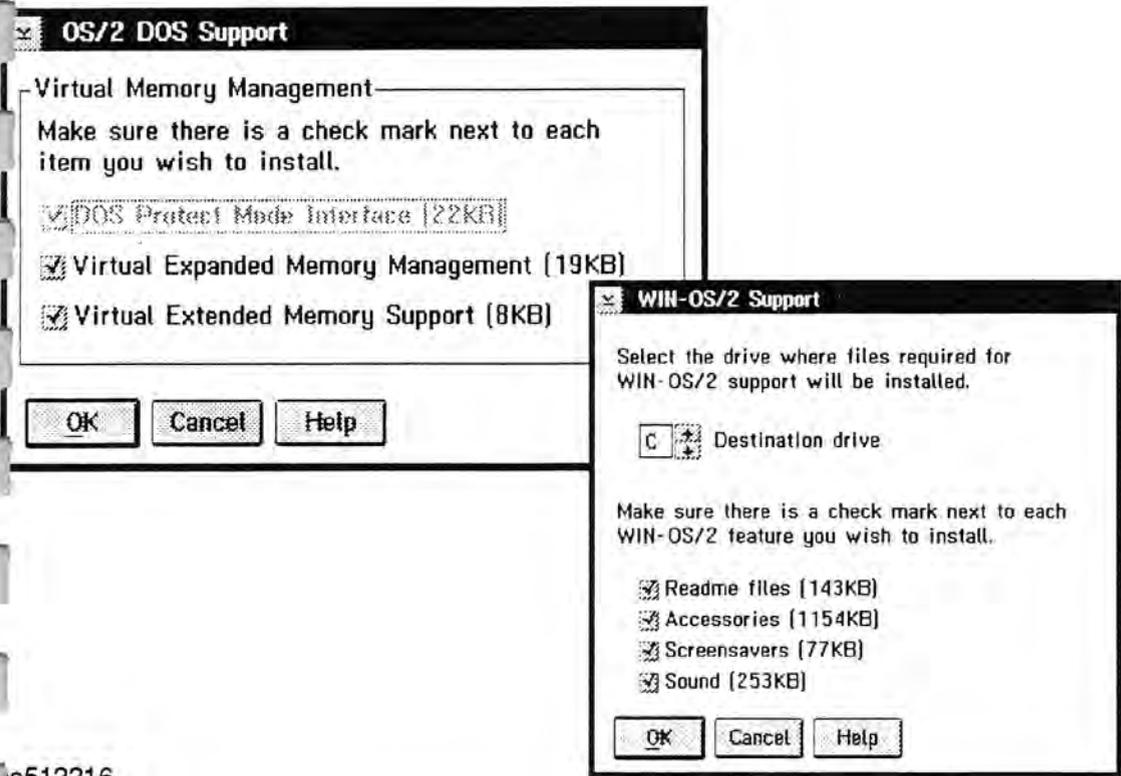
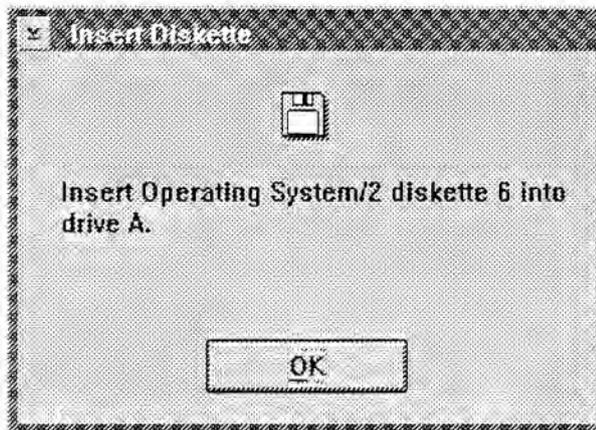


Figure 2-18. More...(Fine Tuning)

If you decide you want only *some* of the available DOS environment. Select the **More** push button to the right of *OS/2 DOS Support* and/or *WIN-OS/2 Support*. When the panels shown above appear, select the options you want to include with your DOS or WIN-OS/2 features.

The number that is shown to the right of each feature indicates the amount of hard disk space required to install that **entire** feature.



ps51231H

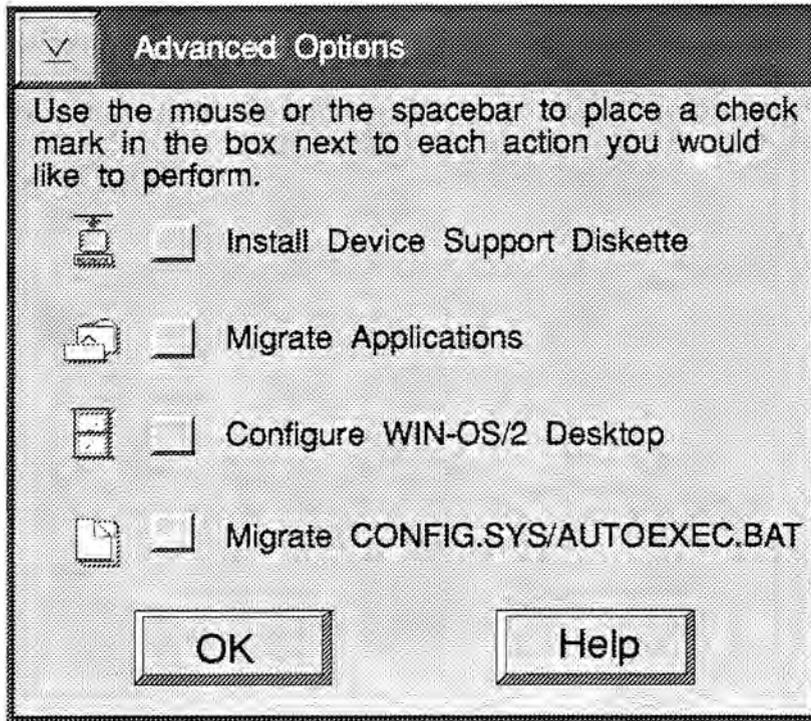
Figure 2-19. Insert Diskette

After you proceed past OS/2 Setup and Installation, you will be prompted to to insert diskettes.

Insert diskettes and select the **OK** push button. When all of the files required from this diskette have been transferred to the hard disk, you will be prompted to insert the next diskette in the sequence.

While files are being transferred, the progress indicator will be shown on the screen.

The panel indicates which diskette is being processed and the progress indicator will show the percentage of required files that have been transferred. Above the progress indicator, the text will show the name of the file which is currently being processed.



ps512220

Figure 2-20. Advanced Options

The installation process then shows you the **Advanced Options** dialog. Make sure there is a check mark next to any action you want to perform and then select *OK*. *Install Device Support Diskette* Check this button if you want to install additional device drivers that are not supplied by the operating system and the device you are installing comes with a device support diskette. *Migrate Applications* Check this option if you want to migrate DOS and Windows programs on your hard disk to the OS/2 WPS. The programs will be placed in a folder on the Desktop. *Configure WIN-OS/2 Desktop* Check this box if you want to install a standard WIN-OS/2 Desktop or preserve a currently installed WIN-OS/2 Desktop. *Migrate CONFIG.SYS/AUTOEXEC.BAT* Check this box if you want to copy statements from an existing CONFIG.SYS or AUTOEXEC.BAT file to your new CONFIG.SYS and AUTOEXEC.BAT.

Depending on how many of the above boxes you check, you will get dialogs with configuration details pertaining to those options

Subtopic Summary

In this subtopic the student learned how to install OS/2 Version 2.1 as the ONLY operating system on the computer.

This subtopic illustrated how to:

- Begin the installation of OS/2 Version 2.1.
- Select the installation options desired for your system.
- Follow the instructions on the screen to complete the installation procedure.

This concludes the subtopic of the topic
"Installing OS/2 Version 2.1".

Boot Installation

Subtopic objectives

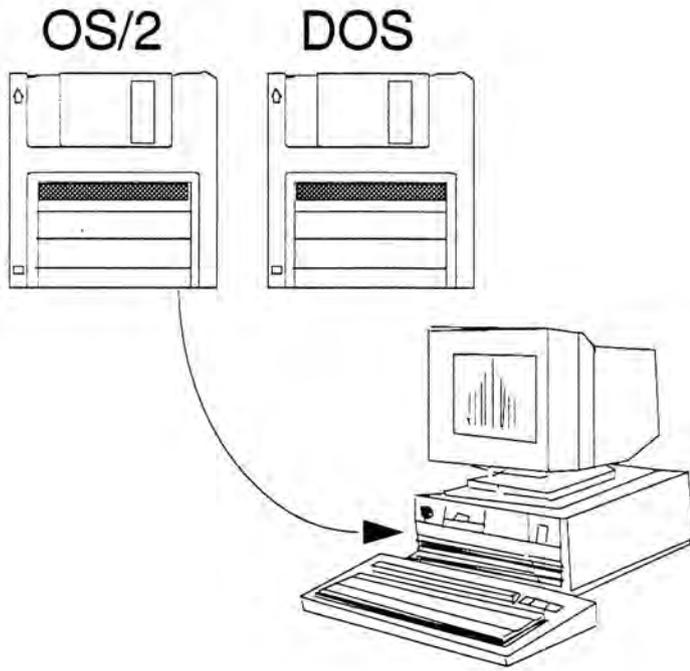
Terminal objective:

After attending this subtopic the student should be able to install the OS/2 version 2.1 operating system on a hard disk that already has DOS installed so as to be able to use both operating systems.

Enabling objectives:

After attending this subtopic the student should be able to:

- Set up the DOS directory structure for dual boot.
- Modify the DOS CONFIG.SYS and AUTOEXEC.BAT files for dual boot.
- Install the OS/2 Version 2.1 operating system for dual boot.



ps512300

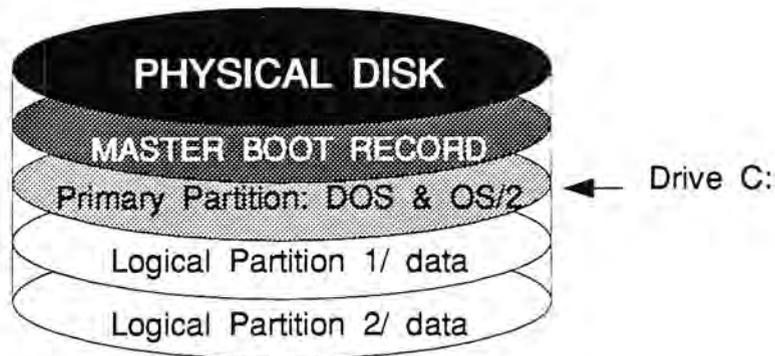
Figure 2-21. Dual Operating Systems

With dual boot, you set up your **primary** partition to allow you to switch back and forth from OS/2 to DOS. The OS/2 operating system is installed with DOS in the same primary partition of your hard disk.

Note: It is assumed that DOS is already installed on your system and that you are familiar with using DOS.

The DOS version must be 3.2 or higher. The DUAL BOOT feature is operable only after OS/2 is installed.

Keep in mind that although both operating systems will reside on the same hard disk partition, they cannot both be running at the same time. You switch between the operating systems.

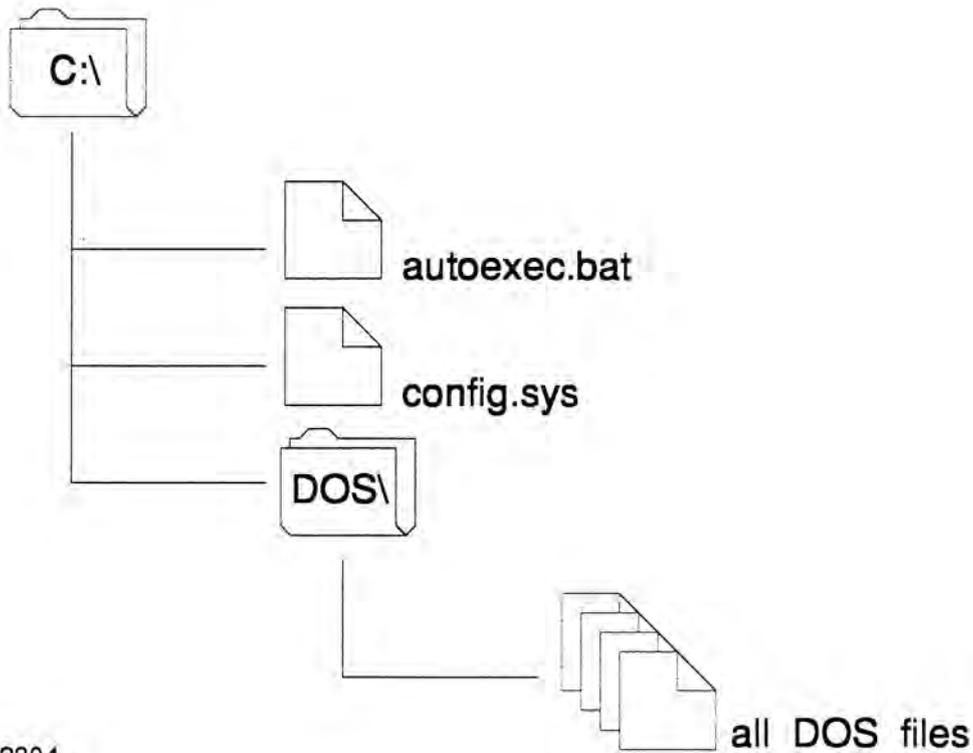


ps512302

Figure 2-22. The Primary Partition

The DOS operating system must exist on drive C: before the OS/2 operating system is installed. Dual boot will operate only after OS/2 2.1 is installed. Notice that both operating systems will exist on the C: drive. Either can be started from this partition on your hard disk.

OS/2 versions 1.3 and 2.0 also have dual boot capability. If DOS currently exists on your C:, either alone or as a dual boot system with a previous version of OS/2, you must be able to start the computer with it in order for dual boot to work correctly.



ps512304

Figure 2-23. Directory Structure

Before installing OS/2 Version 2.1, check to see that your **DOS** system is setup correctly:

- The **DOS** **AUTOEXEC.BAT** and **CONFIG.SYS** files must be in the root directory of your hard disk before OS/2 is installed.
- All DOS commands and utilities must be located in a subdirectory (such as C:\DOS) and **not** in the root directory.
- **COMMAND.COM** used by DOS must be in the DOS subdirectory.

AUTOEXEC.BAT

```
set comspec = c:\dos\command.com  
path = c:\dos  
append = c:\dos  
copy c:\dos\command.com c:\ > nul
```

ps512306

Figure 2-24. Modifications to AUTOEXEC.BAT

Before installing the OS/2 operating system, you must modify the DOS AUTOEXEC.BAT files on your hard disk.

Using any editor, add the following statements to the AUTOEXEC.BAT file:

```
SET COMSPEC=C:\DOS\COMMAND.COM  
PATH=C:\DOS  
APPEND=C:\DOS  
COPY C:\DOS\COMMAND.COM C:\ >NUL
```

Note: These statements include the minimum required parameters for dual boot. You can add additional parameters if you wish. Ensure that the DOS directory is specified as indicated above.

CONFIG.SYS

```
shell = c:\dos\command.com /p
```

ps512308

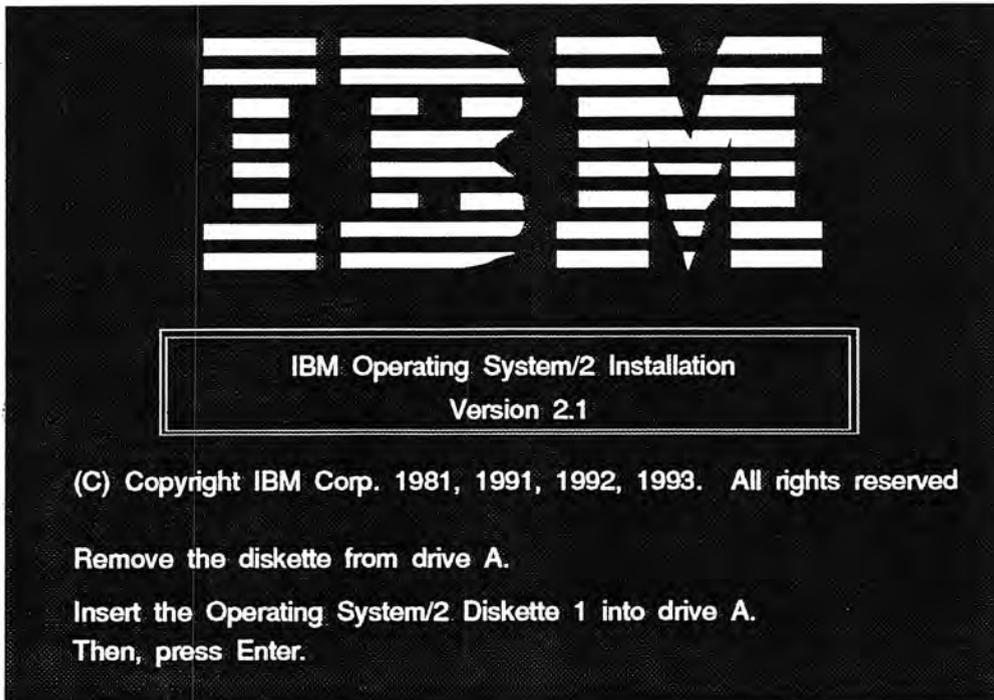
Figure 2-25. Modifications to CONFIG.SYS

Before installing OS/2, you must also modify the **DOS CONFIG.SYS** file. Using any editor, add the following statement:

```
SHELL=C:\DOS\COMMAND.COM /P
```

Note: These statements include the minimum required parameters for dual boot. You can add additional parameters if you wish.

When you install OS/2, the installation program will create **OS/2 AUTOEXEC.BAT** and **CONFIG.SYS** files. Your existing **DOS AUTOEXEC.BAT** and **CONFIG.SYS** files will be moved to the **C:\OS2\SYSTEM** subdirectory and their file extensions will be changed to "DOS."



ps512310

Figure 2-26. Install OS/2 Version 2.1

After the DOS preparations, insert the OS/2 Installation diskette in drive A. If your system is on, press and hold the CTRL, ALT and, the DEL keys. If your system is off, insert the OS/2 Installation diskette in drive A and turn on the computer.

Note: Remember that you must **not** format your hard disk during installation.

After installing OS/2, use the **BOOT** command to switch from one operating system to another. If you are running OS/2 and want to switch to DOS, type:

```
BOOT /DOS
```

from an OS/2 command prompt.

If you are running DOS and want to switch to OS/2, type:

```
C:\OS2\BOOT /OS2
```

Whenever this partition is selected to start, it starts in whichever operating system was last used (DOS or OS/2).

Subtopic Summary

In this subtopic, the student learned how to install the OS/2 Version 2.1 operating system on a partition that already has DOS installed so as to be able to use both systems.

This subtopic illustrated how to:

- Set up the DOS directory structure for dual boot.
- Modify the DOS CONFIG.SYS and AUTOEXEC.BAT files for dual boot.
- Install the OS/2 Version 2.1 operating system for dual boot.

This concludes another subtopic of the topic
"Installing OS/2 Version 2.1".

Manager

Subtopic objectives

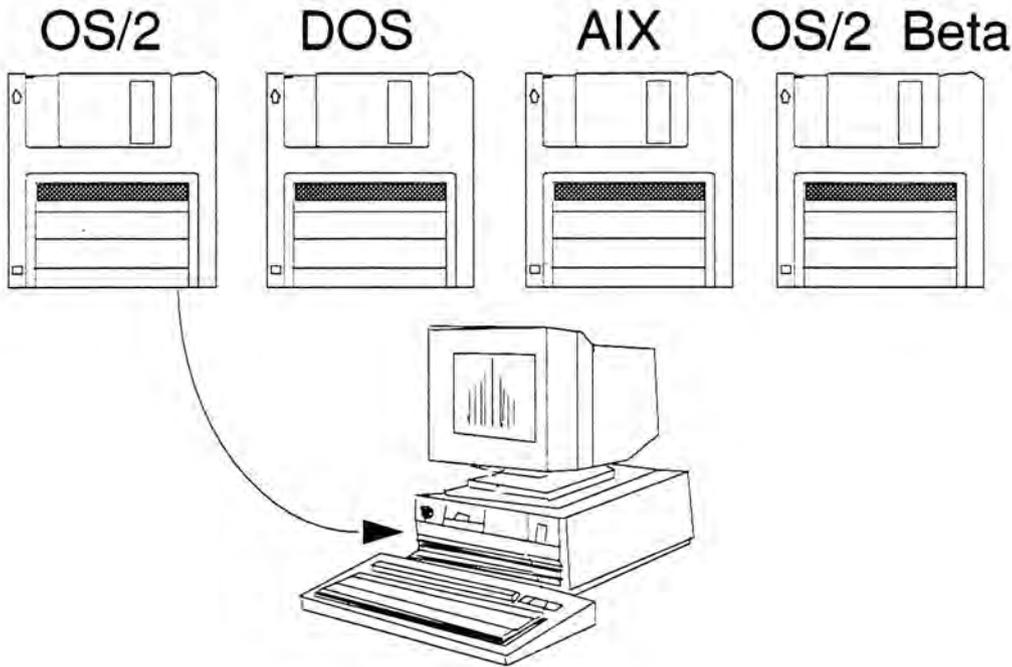
Terminal objective:

After attending this subtopic the student should be able to configure the hard disk so as to be able to install multiple operating systems from which the system can be started.

Enabling objectives:

After attending this subtopic the student should be able to:

- Set up multiple partitions on the hard disk during the installation of OS/2.
- Install the Boot Manager feature.
- Start the computer and use the Boot Manager startup menu to select one of the operating systems.



ps512400

Figure 2-27. Multiple Operating Systems

With the Boot Manager feature of OS/2 2.1, you can have multiple operating systems installed in separate partitions of your hard disk. The hard disk can be set up to accommodate a variety of operating systems. As with dual boot, the operating systems cannot be run at the same time. Unlike dual boot, each operating system will exist in its own partition.

Once the operating systems are installed, Boot Manager is used to select which operating system you want to start.

*BMX - OS/2 program.
assign letter.
like logical.*

Boot Manager
Menu

OS/2 2.1

DOS 5.0

OS/2 Beta

No selection within 30 seconds, boots OS/2 2.1

Press ESC to disable timer

Use ↑ or ↓ to select. Press Enter to boot

ps512402

Figure 2-28. Boot Manager Menu

To manage the selective startup of multiple operating systems on one computer, you use the Boot Manager. From the Boot Manager menu you select which operating system you want to use each time you start your system.

The figure above is an example of a Boot Manager menu that includes three operating systems. The FDISK utility program is used during the installation of OS/2 2.1 to install the Boot Manager feature.

Installation Drive Selection

If you want multiple versions of DOS, OS/2 or other operating systems on the same hard disk, refer to the Installation Guide for information on OS/2 Hard Disk Management before continuing.

If you have multiple partitions set up on your hard disk, select option 2 to verify that the correct partition is active.

OS/2 will be installed on drive C:

Select an option:

1. Accept the drive

2. Specify a different drive or partition

If you select option 2, the FDISK screen is displayed.

Enter Esc=Cancel F3=Exit F1=Help

ps512404

Figure 2-29. Installation Drive Selection

During installation you have the option of installing OS/2 in one partition that takes up the entire hard disk or to separate your hard disk into partitions. If you choose to separate your hard disk into multiple partitions, you select option 2 at the Installation Drive Selection screen.

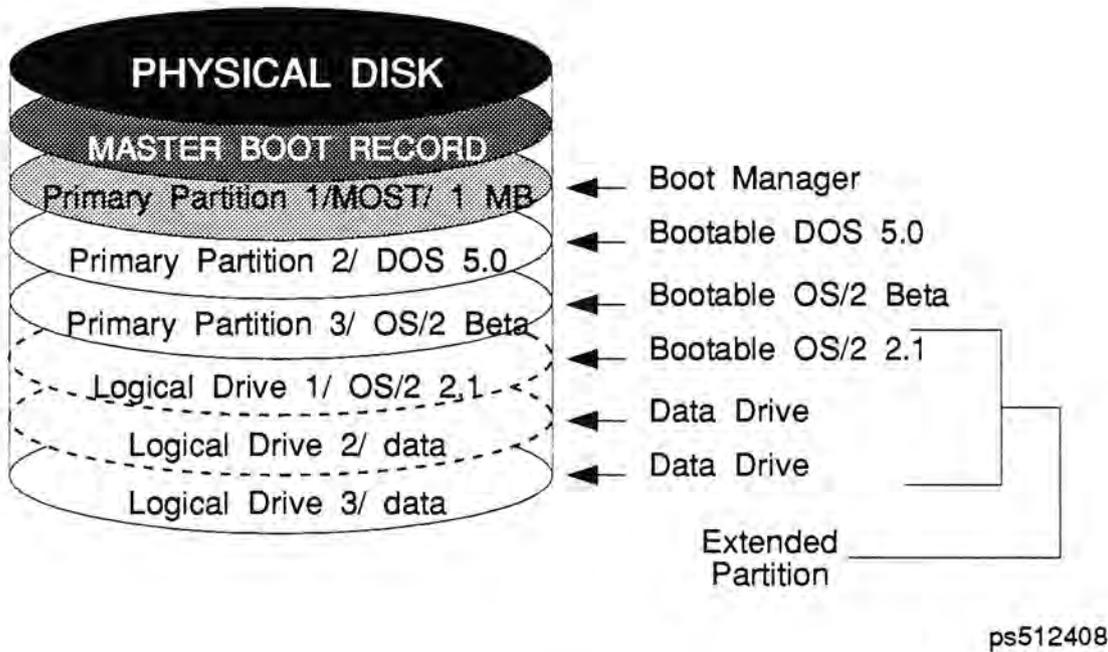


Figure 2-31. Primary Partitions and Logical Drives

The hard disk can be separated into a maximum of four partitions. You can have up to four primary partitions or up to 3 primary partitions and one extended partition.

If you are going to install multiple operating systems on your hard disk, you must create one primary partition to contain the programs that manage the startup of multiple operating systems. This partition is referred to as the Boot Manager partition.

The illustration above, shows an example of a hard disk that is subdivided to include logical drives within an extended partition. Logical drives are typically used to hold programs and data. However, you can install OS/2 2.1 in a logical drive.

Primary partitions share C:

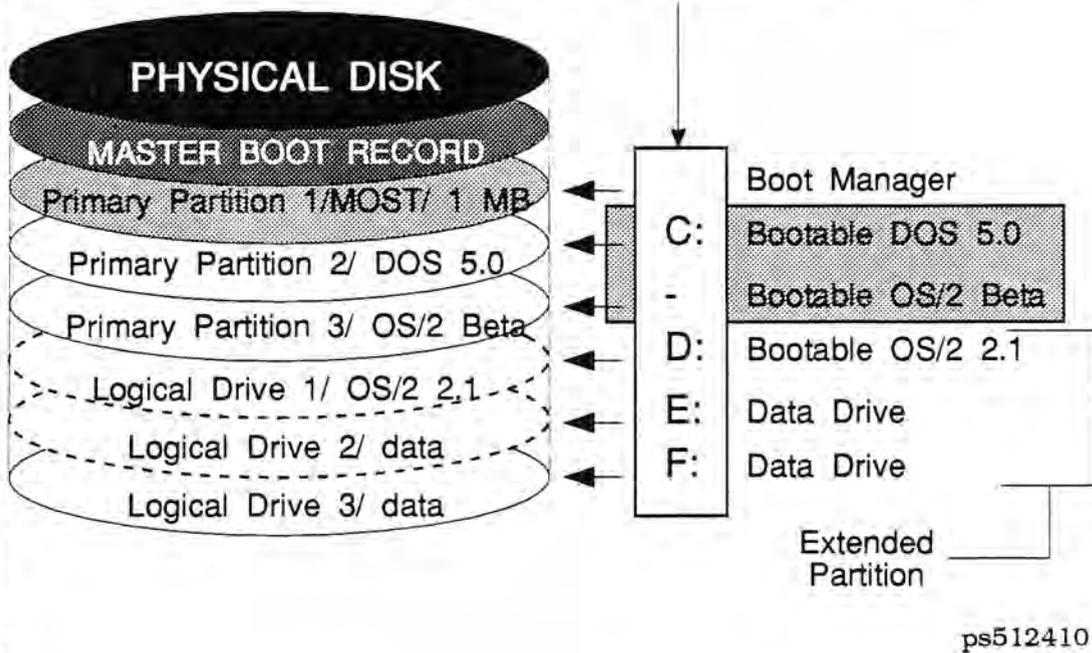


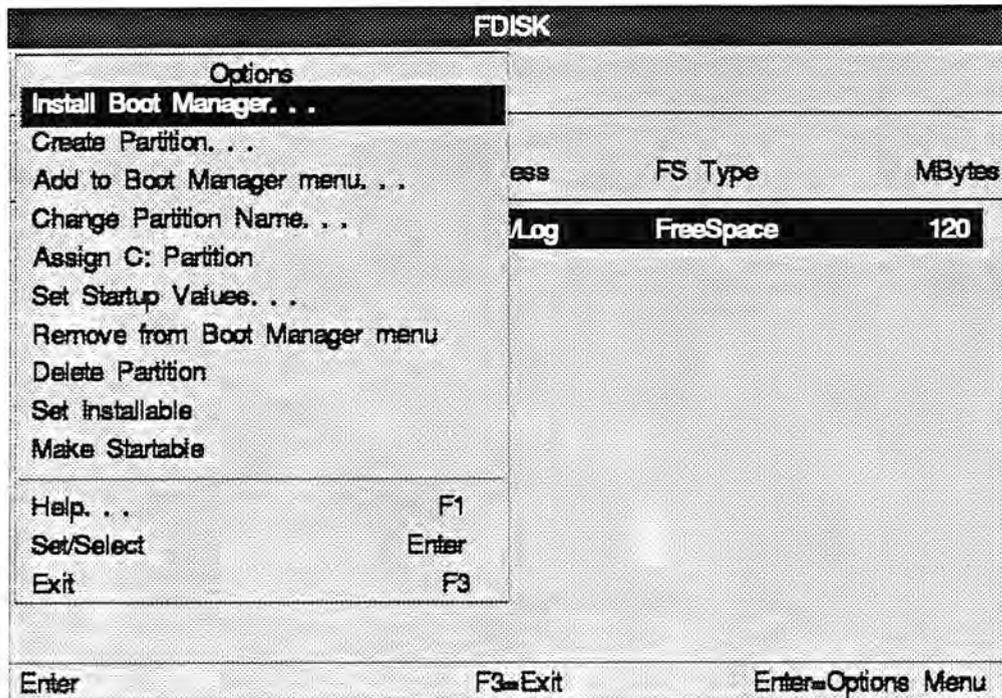
Figure 2-32. Drive Letter Assignments

All of the logical drives exist within one partition, the extended partition. The extended partition is created the first time you create a logical drive.

One of the differences between a logical drive and a primary partition is that each logical drive is assigned a unique drive letter. In contrast, all primary partitions on a hard disk, share the same drive letter. On the first **physical** hard disk on your system, the primary partitions share drive C:. **This means that only one primary partition on a hard disk can be accessed at a time.**

Note: The Boot Manager partition is different from other primary partitions because it is never assigned a drive letter.

The operating system that is active when you start the system performs a process known as **drive mapping**. Drive mapping assigns drive letters to partitions and logical drives. Primary partitions are mapped first and then all logical drives within extended partitions are assigned subsequent drive letters.



ps512414

Figure 2-34. Install Boot Manager

To create a Boot Manager partition you will require 1 MB of Freespace.

1. Make sure that the **FreeSpace** line is highlighted. If it is not, use either the up or down arrow to move the bar and highlight it.
2. Press ENTER to display the options menu.
3. Select **Install Boot Manager** and press ENTER.
4. Depending on how your machine is configured, select either the END of Freespace or the START of the Freespace.

The above picture indicates that the entire drive is freespace, that will not always be the case.

FDISK				
Disk 1				
Partition Name	Information Status	Access	FS Type	MBytes
	Startable	: Primary	BOOT MANAGER	1
	None	: Pri/Log	FreeSpace	119

Enter F3=Exit Enter=Options Menu

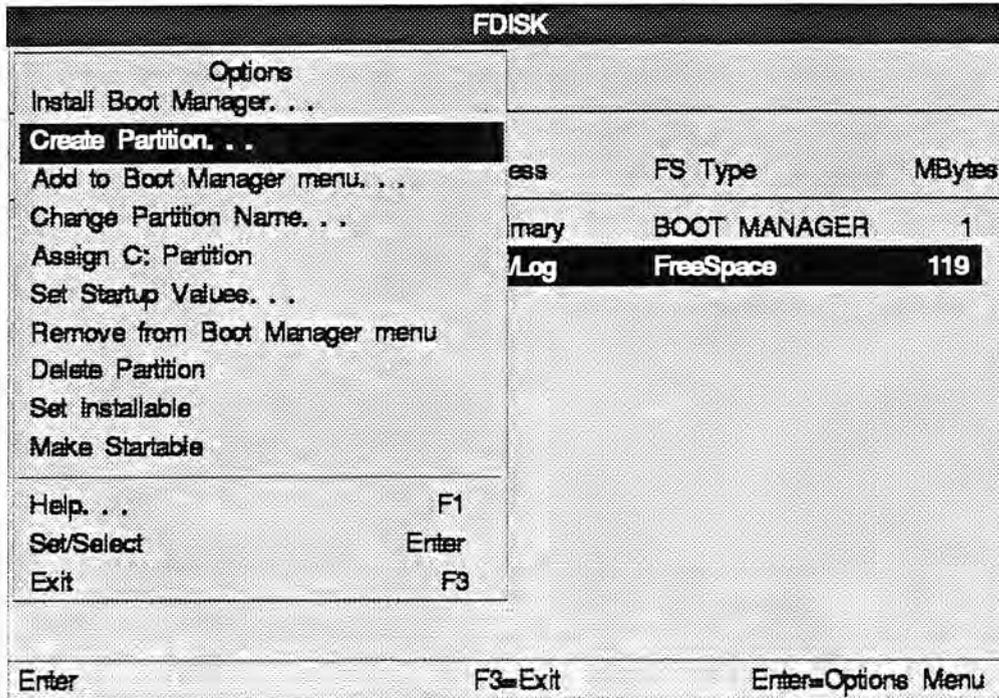
ps512416

Figure 2-35. Fdisk Screen Updated

After you create the partition for the Boot Manager, the FDISK screen changes to reflect the existence of a new partition.

- The partition has no name since it will never appear in the Boot Manager menu.
- The status is **Startable** meaning that it will be the partition that is activated when you start your system.
- File System type is simply "Boot Manager."
- The size in MegaBytes is 1 MB.

Next we create a small primary partition on which to install DOS. Move the black bar with the up or down arrow so as to highlight the **FreeSpace** line. Press ENTER to display the options menu.

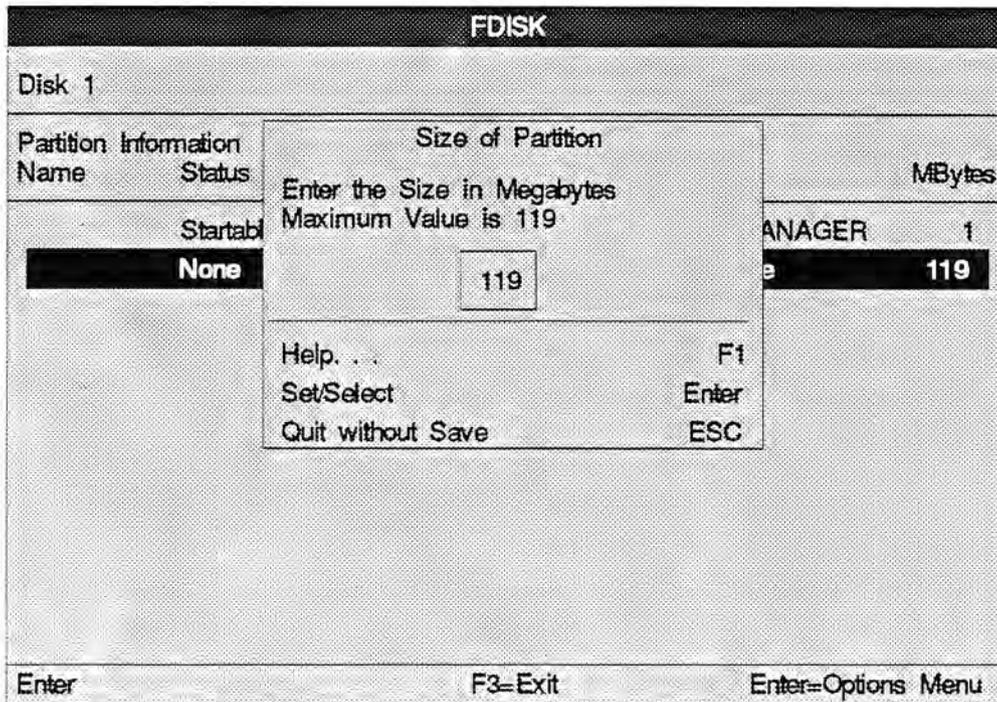


ps512418

Figure 2-36. Create Primary Partition

When you press ENTER, you bring up the **Options** menu. The Options menu displays some options in blue text and some in black. Only the black options can be selected. The blue ones are not available. The availability will change depending on what is being done. For example, in the above panel, the only available options are **Create Partition** and **Set Startup Values**. The option **Create Partition** is highlighted.

Select **Create Partition** and press ENTER.



ps512420

Figure 2-37. Size of the Partition

The next piece of information you are prompted for is the size of the partition you are creating. When the box appears, it has the size of the freespace not yet allocated on your hard disk.

Pressing the ENTER key at this point, would create a partition the size of the remainder of hard disk space.

Typically one types a number in the box to indicate the size of the hard disk. After you type the size, press ENTER.

FDISK		
Disk 1		
Partition Information	Type of Partition	MBytes
Name	Status	
	Primary Partition	
Startable	Extended Logical Drive	MANAGER 1
None		119
Help . . . F1		
Set/Select Enter		
Quit without Save ESC		
Enter	F3=Exit	Enter=Options Menu

ps512422

Figure 2-38. Primary or Extended Partition

The **Type of Partition** prompt requires that you select whether your partition will be a primary or extended partition. Before proceeding, here's a brief recap on the subject of partitions.

The most fundamental division of a hard disk is the partition. A hard disk may be divided into multiple partitions. Each partition is a logical boundary around a physical region of the disk. Partitions, once created, do not change size. Changing the size of a partition requires FDISK and will result in erasing all of its contents. A partition is either a *Primary partition* or an *Extended partition*.

Only one primary partition per hard disk can be active at a time. Thus, only one primary partition is assigned the letter *C* at any time. Any other primary partitions are not mapped, i.e., they will not be given drive letters.

An Extended partition may be divided into multiple logical disk drives. Each drive is accessed with a different letter as if it were a physically separate disk drive. Their drive letters could start from the letter *D* and on through the alphabet.

To select **Primary Partition** simply press ENTER.

FDISK				
Disk 1				
Partition Name	Information Status	Access	FS Type	MBytes
	Startable	: Primary	BOOT MANAGER	1
	None	: Primary	Unformatted	4
	None	: Pri/Log	FreeSpace	115
Enter F3=Exit Enter=Options Menu				

ps512424

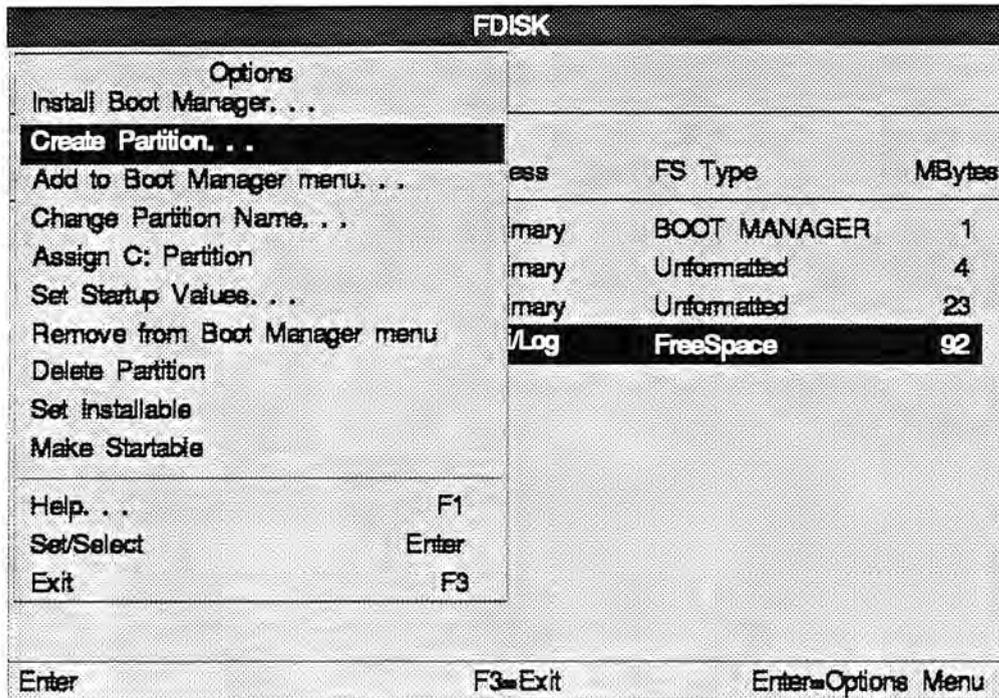
Figure 2-39. Second Primary Partition

As you can see in the illustration above, the FDISK screen has been updated once again to reflect a new primary partition. FDISK indicates that it is a primary partition, that it is unformatted and that it is 4 MB in size.

Notice also the C:. Since it is the only primary partition in the configuration, so far (aside from the Boot Manager partition), it is made active by default.

A hard disk can hold a maximum of four primary partitions. We have now used two.

In the example we've been following, we will create another primary partition which will be used to install OS/2 Version 1.3. Assume that the same steps that were used before, to create the 4 MB primary partition shown above, were repeated. The one difference is that the size of the partition is adjusted for OS/2 1.3.



ps512426

Figure 2-40. Create Extended Partition

On the space that remains in this example, we will create an **extended partition** with two logical drives.

Once again the bar is moved over the area labelled as **FreeSpace** and the ENTER key is pressed. The **Options** menu appears. To create the first of two logical drives in the extended partition, the black bar is moved so as to highlight **Create Partition** and ENTER is pressed.

FDISK			
Disk 1			
Partition Name	Information Status	Type of Partition	MBytes
	Startable	Primary Partition	
	None	Extended Logical Drive	MANAGER 1
	None		ed 4
	None		ed 23
	None	Help. . . F1	e 92
		Set/Select Enter	
		Quit without Save ESC	
Enter F3=Exit Enter=Options Menu			

ps512428

Figure 2-41. Extended Logical Drive

The user is now required to select between **Primary Partition** and **Extended Logical Drive**.

To indicate **Extended Logical Drive**, the bar must be moved so as to highlight that option. Once **Extended Logical Drive** is highlighted, the ENTER key is pressed.

FDISK				
Disk 1				
Partition Name	Information Status	Access	FS Type	MBytes
	Startable	: Primary	BOOT MANAGER	1
	None	C: Primary	Unformatted	4
	None	: Primary	Unformatted	23
	None	D: Logical	Unformatted	33
		: Pri/Log	FreeSpace	59
Enter F3=Exit Enter=Options Menu				

ps512430

Figure 2-42. Logical Drive Letter Assigned

The FDISK screen is updated to indicate the creation of our first logical drive.

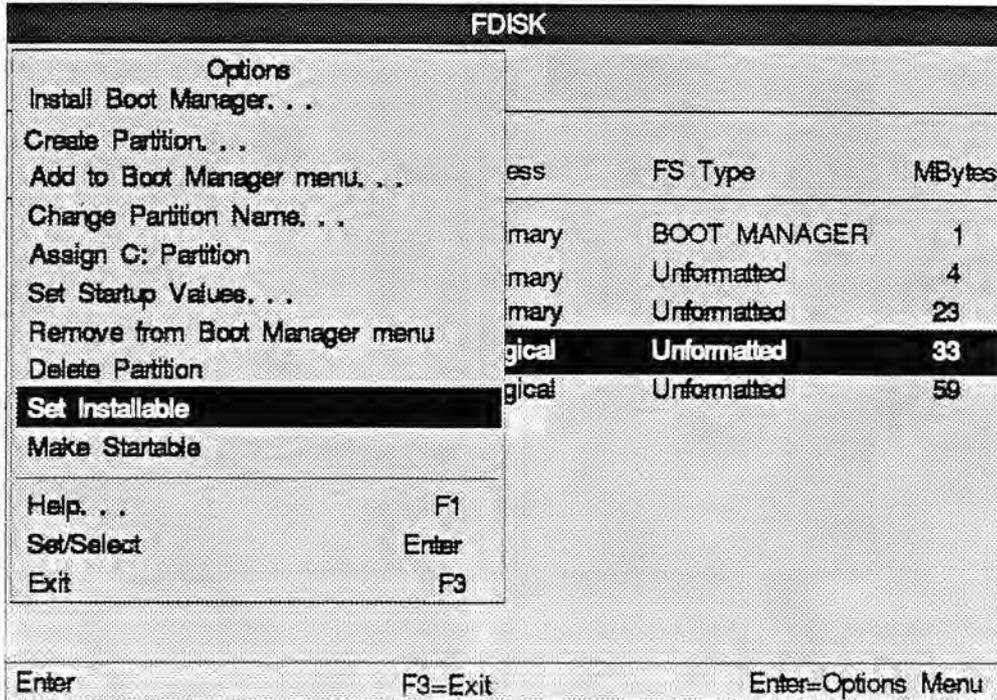
The new line indicates a new drive which has been assigned the drive letter D: . It is shown to be a logical drive, unformatted, and 33 MegaBytes in size.

We have now utilized our maximum of four (4) partitions.

- One for the Boot Manager.
- One for the first Primary Partition (4 MB).
- One for the second Primary Partition (23 MB).
- One for the Extended Partition.

There are still 59 MB of free disk space, however. Remember that an **Extended Partition** can have multiple **Logical Drives** within it.

To finish our example, assume the bulk of the remaining hard disk space will be used for a logical (data) drive within the extended partition. The line which indicates **FreeSpace** is highlighted and ENTER is pressed.



ps512432

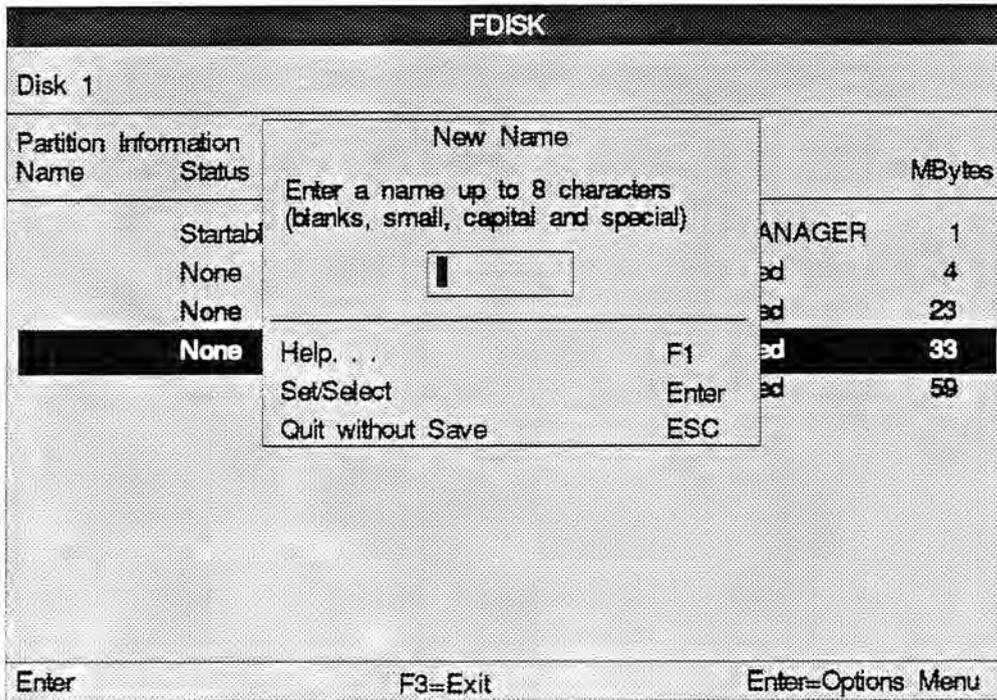
Figure 2-43. Options Menu Choices

In the **Options** menu, move the black bar to the **Set Installable** option and press ENTER.

Set Installable is used to mark a partition or logical drive as the target for installation. For example, before you install the OS/2 Version 2.1 operating system, you set one primary (or logical drive as in this case) as installable before you install the operating system.

Later, when you install other operating systems, you mark **their** partitions as installable before you install the operating system that will go on those other partitions.

After you install OS/2 2.1, the status of this partition is changed from **Installable** to **Bootable**



ps512434

Figure 2-44. Assign Boot Manager Menu Name

Because the Boot Manager partition has been installed and we have indicated that this logical drive is to be *Installable*, the installation program assumes that it is to be added to the Boot Manager menu. To appear on the menu, the selectable drive requires a name. This name is one that should be meaningful to you.

For the purpose of this example, we will call it **OS/2 2.1**. As indicated in the window, it can be any name that is eight characters (including blanks) in length.

Topic Summary

In this subtopic the student learned how to configure the hard disk to have multiple operating systems from which the system can be started.

This subtopic illustrated how to:

- Set up multiple partitions on the hard disk during the installation of OS/2.
- Install the Boot Manager feature.
- Start the computer and use the Boot Manager startup menu to select one of the operating systems.

This concludes another subtopic of the topic
"Installing OS/2 Version 2.1".

ctive Install

ubtopic objectives

Terminal objective:

After attending this subtopic the student will be able to selectively add options to an existing OS/2 2.1 system.

Enabling objectives:

After attending this subtopic the student should be able to:

- Start the Selective Install option.
- Select options to install from the OS/2 Setup and Installation panel.

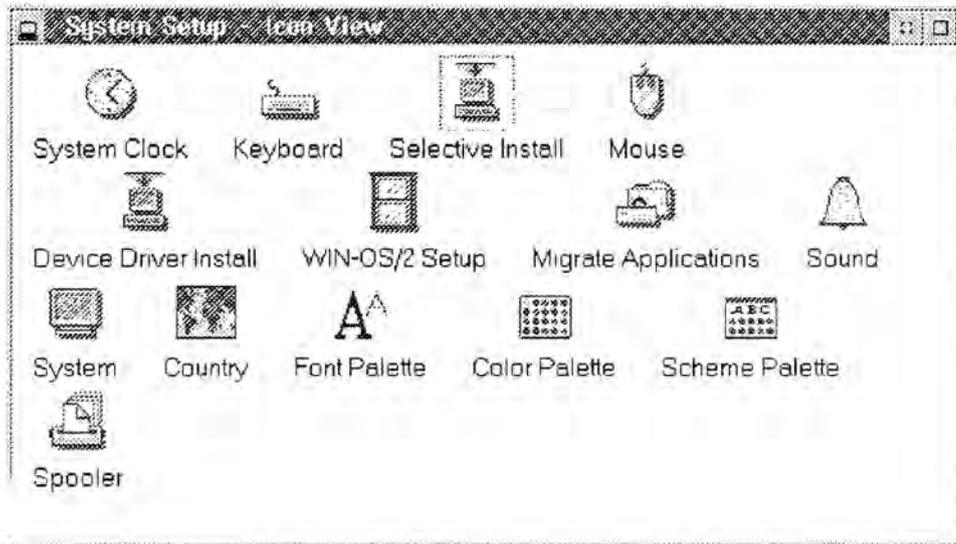


Figure 2-46. Selective Install

1. Open the **System Setup** object
2. Start the **Selective Install** program.

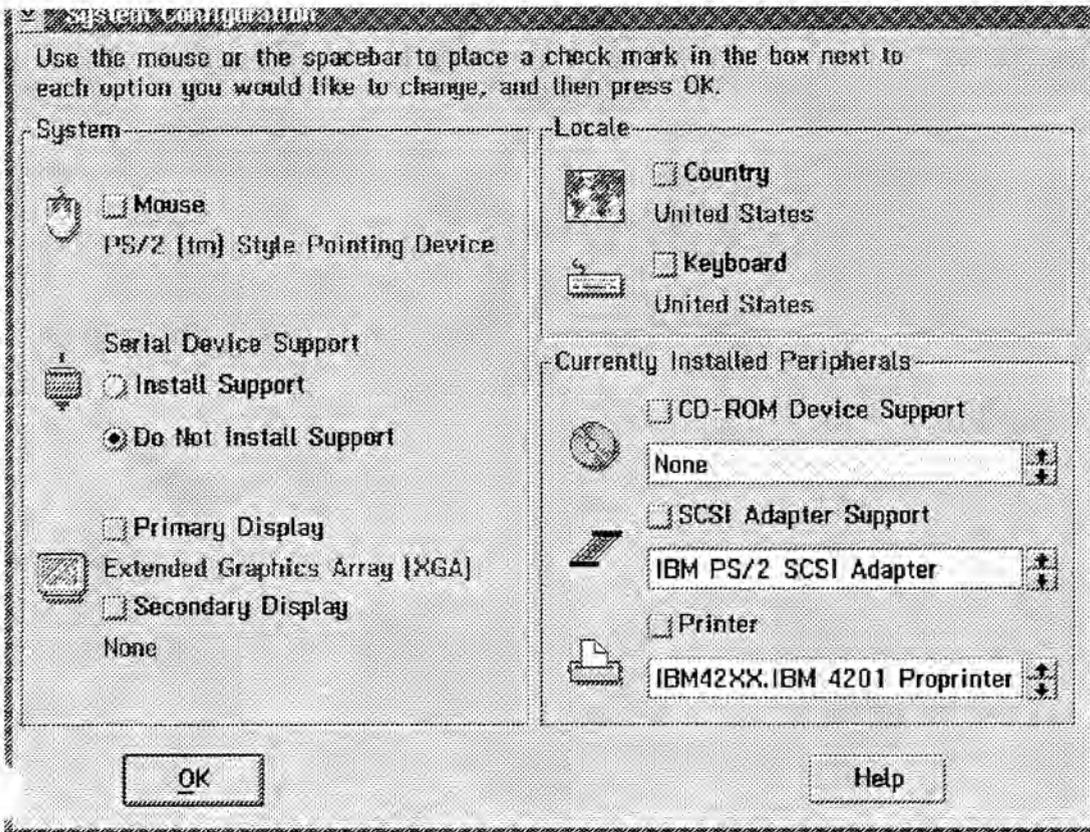


Figure 2-47. System Configuration

During the initial installation your system had certain configuration. Should you determine that you want to alter this configuration, for instance the mouse, this panel will permit you to do so.

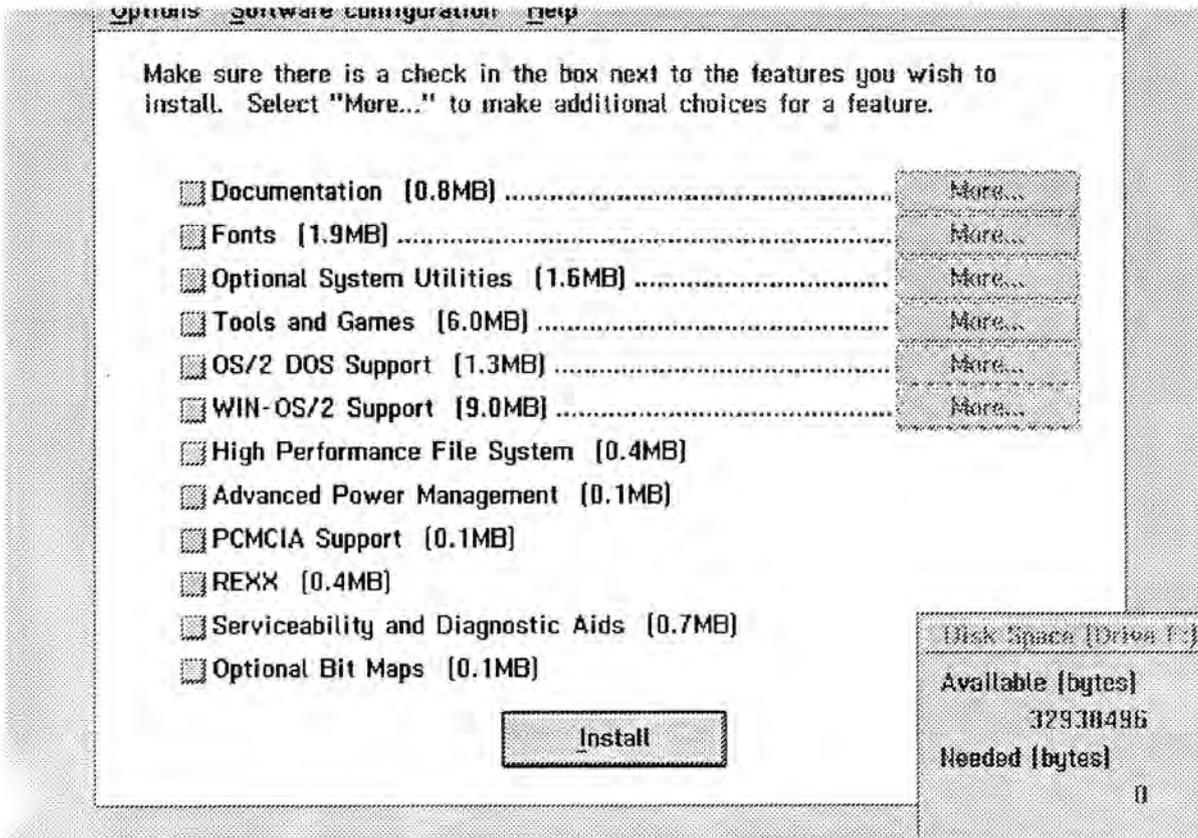


Figure 2-48. Selective Installation of Features

During the initial installation you select certain features to install in OS/2. Should you determine that other features, not previously installed, are required you can then do so from this panel.

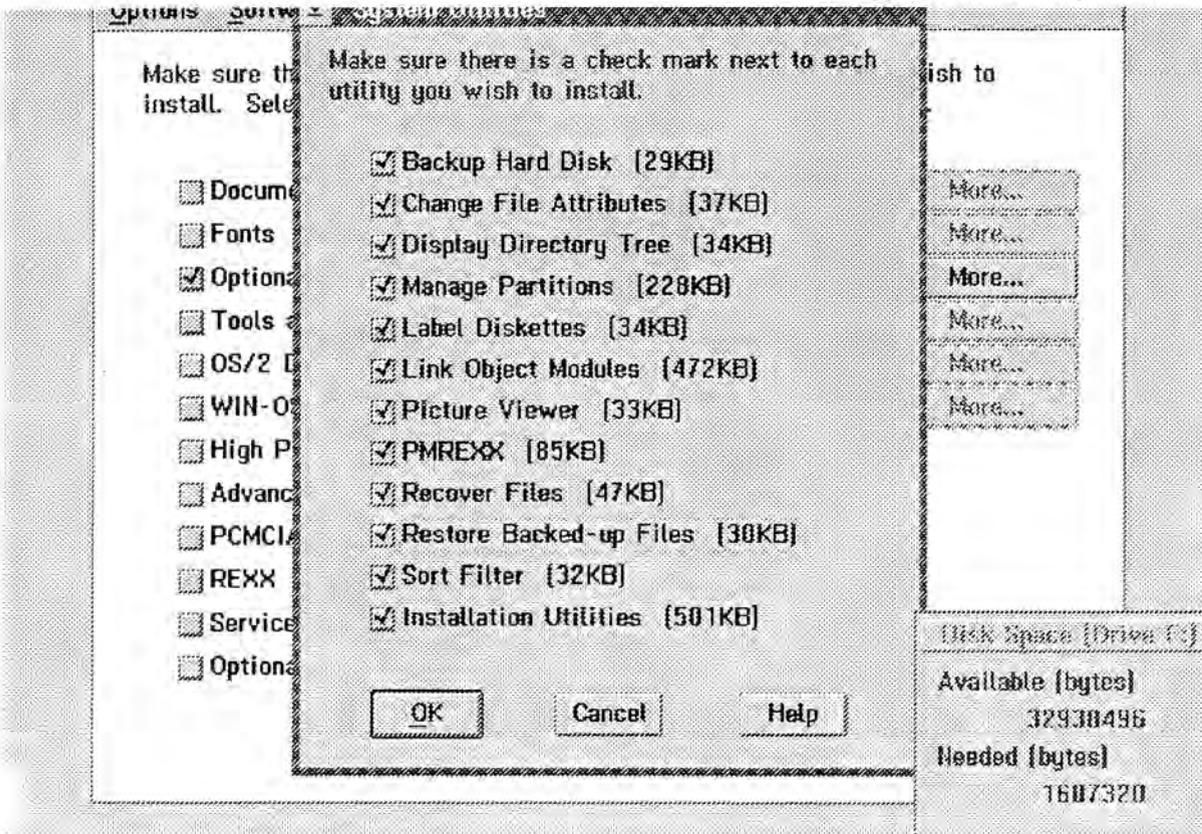


Figure 2-49. Selecting Options

This picture will appear when one selects the **MORE** push button for *Optional System Utilities*.

Subtopic Summary

In this subtopic the student learned how to use the selective install of OS/2 2.1.

This subtopic illustrated how to:

- Install various options of OS/2 2.1 after the initial installation of OS/2.

This concludes another subtopic of the topic
"Installing OS/2 Version 2.1".

Alternative Installation

Subtopic objectives

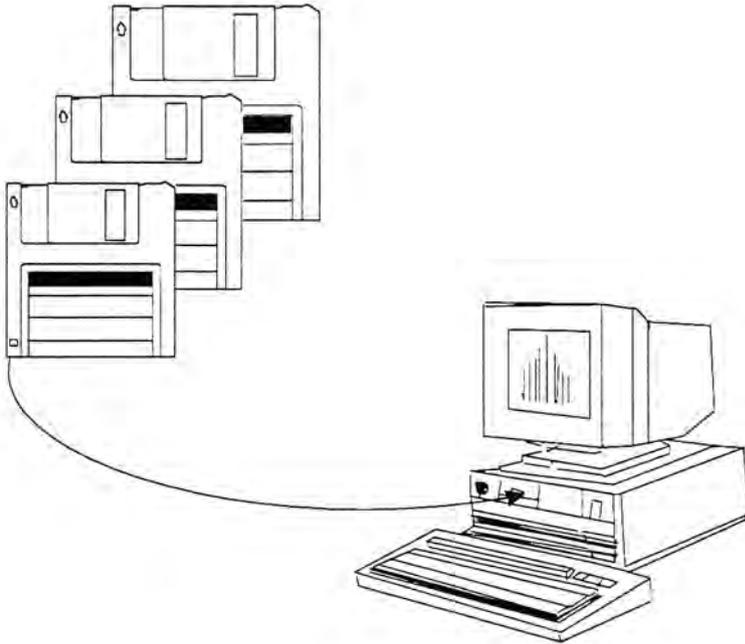
Terminal objective:

After attending this subtopic the student should be able to briefly describe the environment that has created the need and ability for central distribution and a general description of how it can be done.

Enabling objectives:

Upon completion of this topic the student should be able to describe:

- How to create a response file to install OS/2 2.1



ps512600

Figure 2-50. Regular Installation

A regular installation is one in which the process is started from drive A: by inserting the "Installation Diskette" in drive A:, starting or rebooting the system and proceeding by feeding diskettes, in turn, into the A: drive.

Redirected I/O refers to capability of OS/2 2.1 to use drive letters that are not connected to local drives but to drives, directories, or subdirectories that are located on a remote workstation.

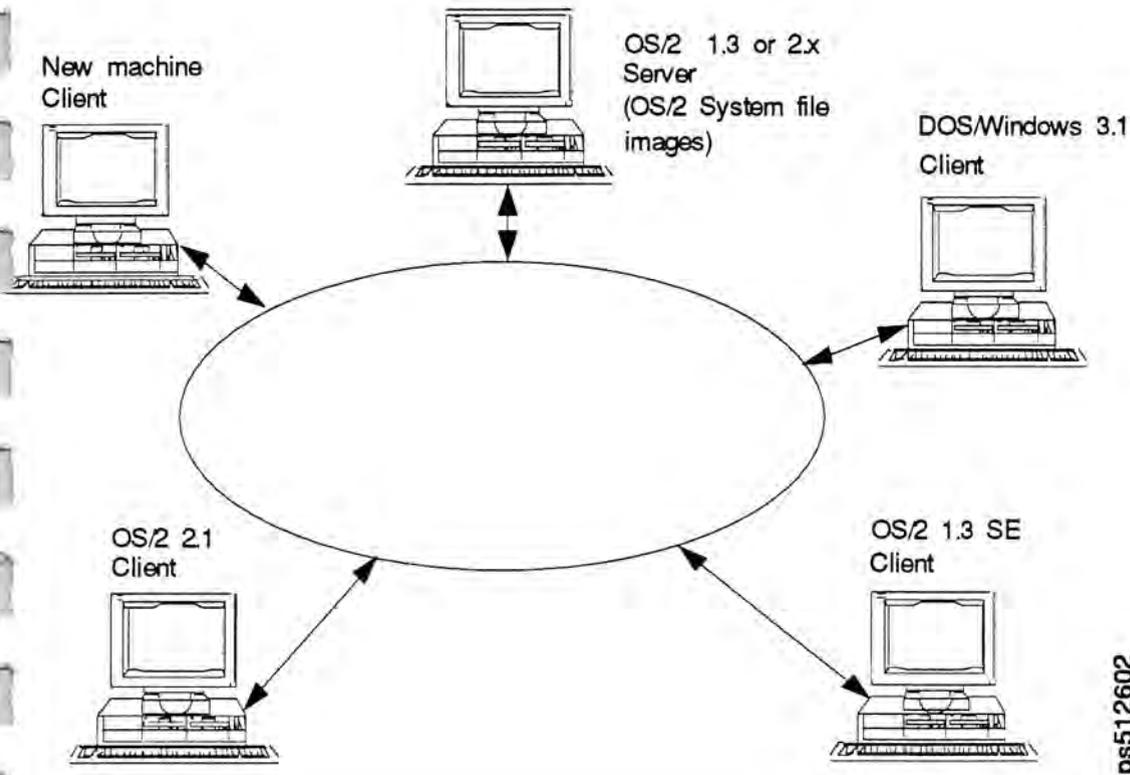


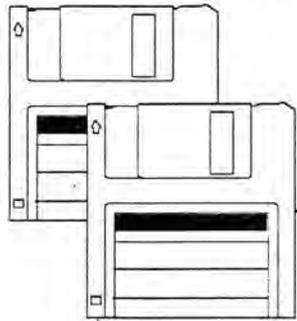
Figure 2-51. Redirected Drive

A workstation must be assigned as the server that distributes the OS/2 2.1 files. Refer to the documentation *GG24-3780 Remote Installation and Maintenance for OS/2* for specific details.

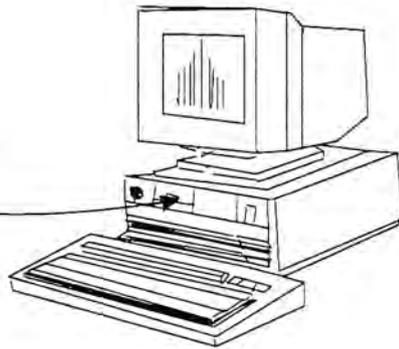
the server can be either an

- IBM LAN Client/Server
- TCP/IP Client /Server
- Novell Netware Client/Server

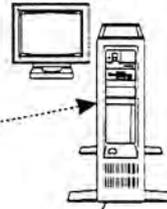
Installation Diskette



LT Diskette



Server



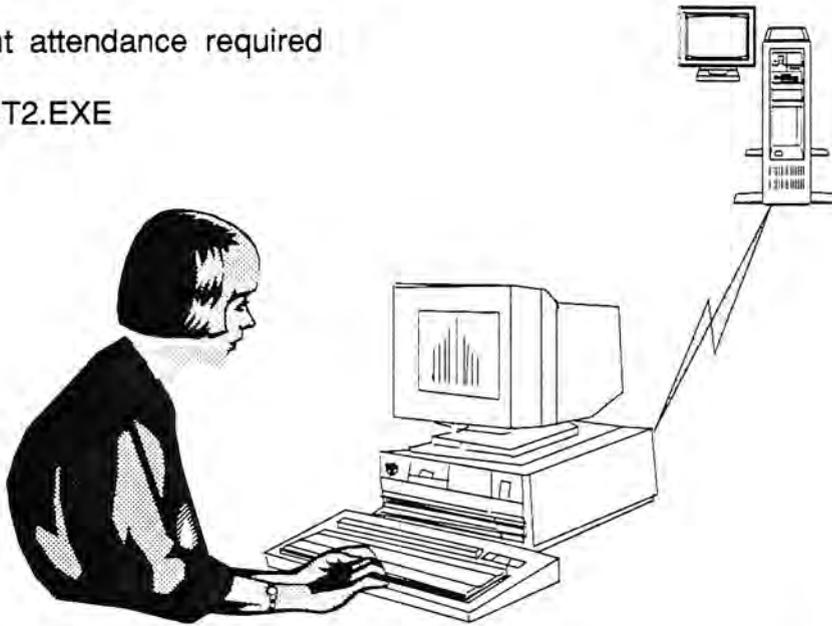
ps512604

Figure 2-52. LT Diskette

The LT Diskette to be created contains the absolute minimum necessary files, but must contain:

- LAN driver support
- Code to establish protocol with the server

- No Response File
- Constant attendance required
- SYSINST2.EXE



ps512608

Figure 2-53. Overview Without Response File

The regular installation process requires the insertion of diskettes and the answering of prompts. Previously you learned how you can eliminate the need to insert diskettes.

The attended, dialog driven installation is a special version of a regular installation. It uses `SYSINST2.EXE`, asks the user questions that must be answered before the installation can proceed, but removes the need to feed diskettes.

- Response File
- Minimal intervention
- RSPINST.EXE

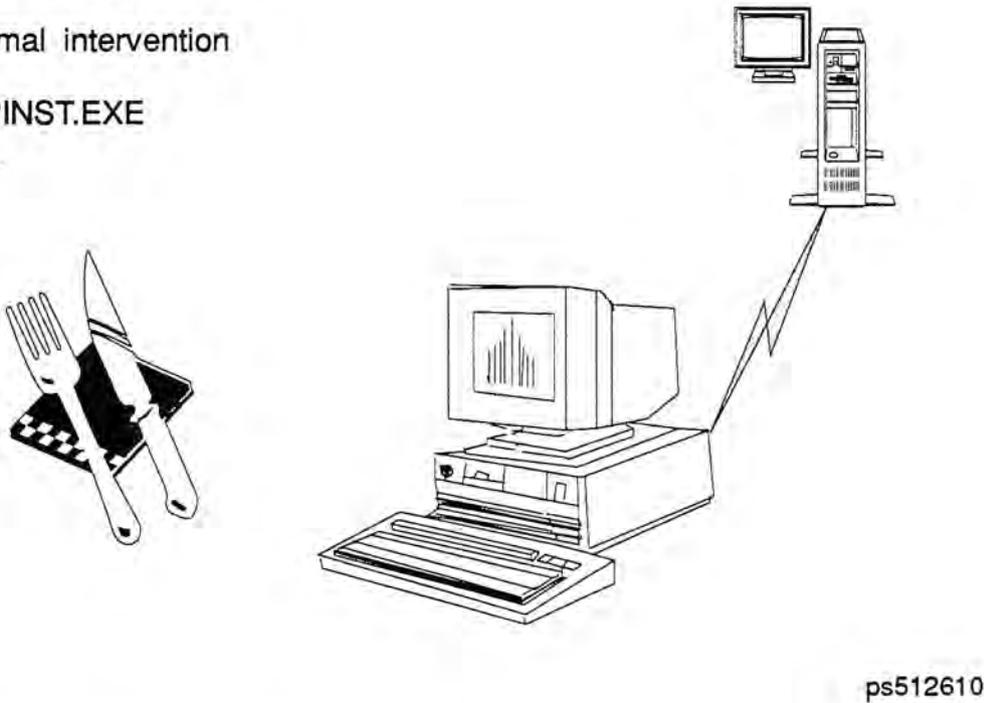


Figure 2-54. Overview With Response File

OS/2 provides a so-called response file `SAMPLE.RSP`. It can be used to edit an installation template for different machines. There is also a response file called `USER.RSP`, that contains all the answers you gave at diskette installation time. After installation, these files can be found in the subdirectory :

```
\OS2\INSTALL
```

Both the regular installation and the response file installation use `SYSINST2.EXE`. `SYSINST2.EXE` will always search for the executable `RSPINST.EXE`. If the system finds it, it will executed it. Otherwise the system will perform an attended installation. contents, and installs without further attendance required.

If you do not have a code server, it is still possible to use a response file. This will avoid you from entering the same choices over and over again when installing different machines.

To perform a diskette response file installation, do the following :

- Copy the file `xxxxxxx.RSP` (x = filename `SAMPLE` or `USER`) to Diskette 1.
- On Diskette 1, rename it `OS2SE21.RSP`.
- Copy `RSPINST.EXE` to that same diskette.
- Start the diskette installation with the regular Installation diskette.
- when prompted for Diskette 1, insert the copy.

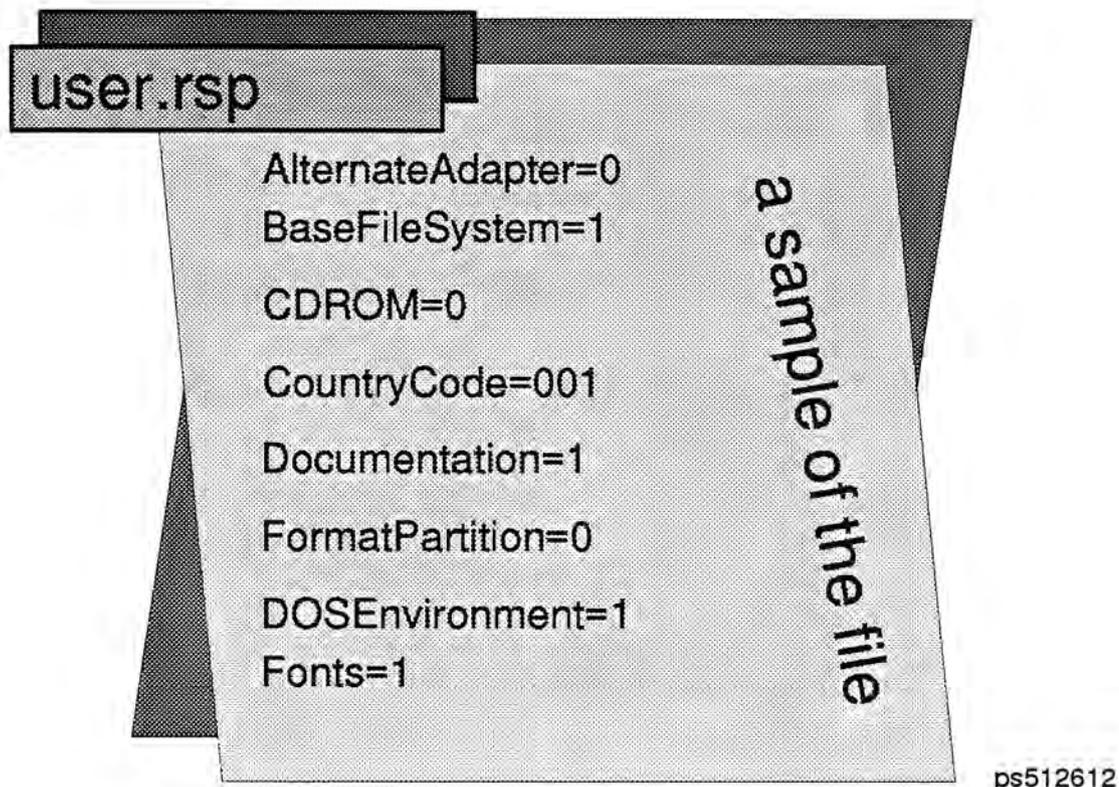


Figure 2-55. Sample Response File

Included is an example of a response file after installation.

(User.rsp):

AlternateAdapter = 0	specifies there no other display adapter
BaseFileSystem = 1	specifies the install partition is HPFS
CDROM = 1	specifies the installation of all CD IFS files
CountryCode = 001	specifies US country code info to be installed
Documentation = 1	specifies all documentation to be installed
DOSEnvironment = 1	specifies DOS and WIN-OS2 environment
Fonts = 1	specifies all fonts to be installed

Topic Summary

In this subtopic the student learned about the response file as an alternative installation method.

Enabling objectives:

This topic enabled the student to create and test their response file

This concludes the final subtopic of the topic
"Installing OS/2 Version 2.1".

Topic Summary

In this topic the student learned how to install the OS/2 Version 2.1 operating system such that OS/2 provides the most efficient support for the user's work on that particular system.

The student was taught how to plan an OS/2 installation then execute whichever one of the installation types listed below is appropriate and be aware of the remote installation feature.

- Basic Installation
- Dual Boot Installation
- Boot Manager Installation
- Selective Installation of Options
- Creation of a simple Response File

This concludes the topic
"Installing OS/2 Version 2.1".

TOPIC 3: The Design of OS/2

Topic objectives

Terminal objective:

After attending this topic the student should be able to list the features of the 80386 and higher micro-processors that OS/2 2.1 has been designed to exploit.

Enabling objectives:

Upon completion of this topic the student should be able to describe key features of the 80386:

- Pipelining
- 32-bit memory management
- Multitasking
- Privilege level
- Virtual 86 Mode

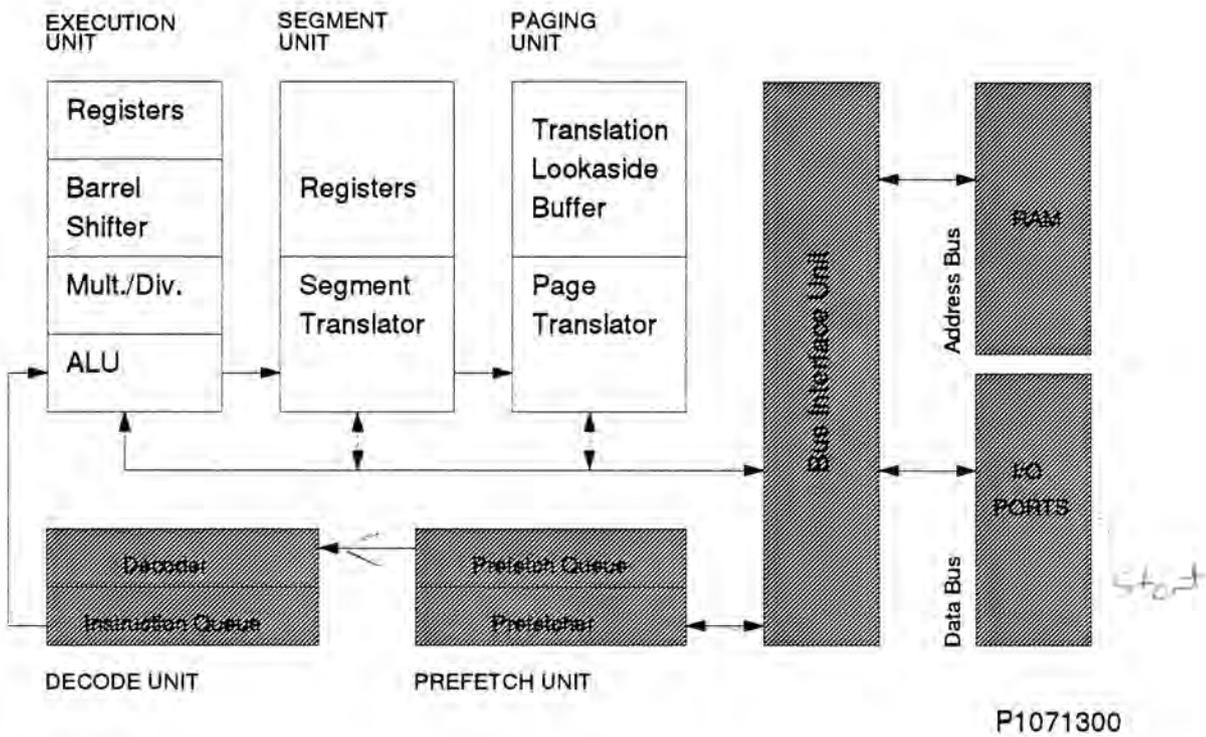


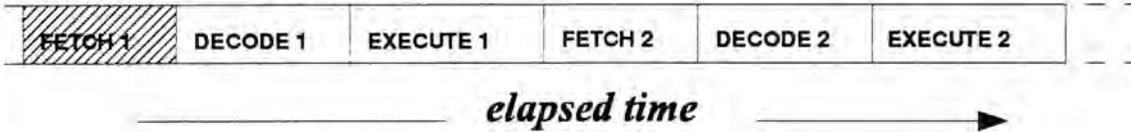
Figure 3-1. Microprocessor Structure

The 80386 architecture is made up of six functional units that operate in parallel. The units of the 80386 are:

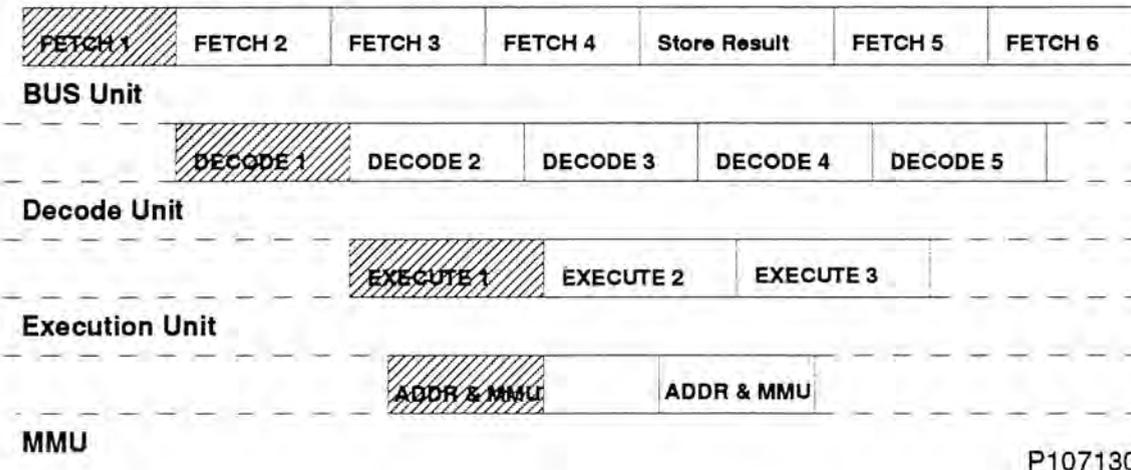
- Bus Interface Unit.
- Code Prefetch Unit.
- Instruction Decode Unit.
- Execution Unit.
- Segmentation Unit.
- Paging Unit.

32 bit address Bus

TYPICAL PROCESSOR



80386 MICRO PROCESSOR



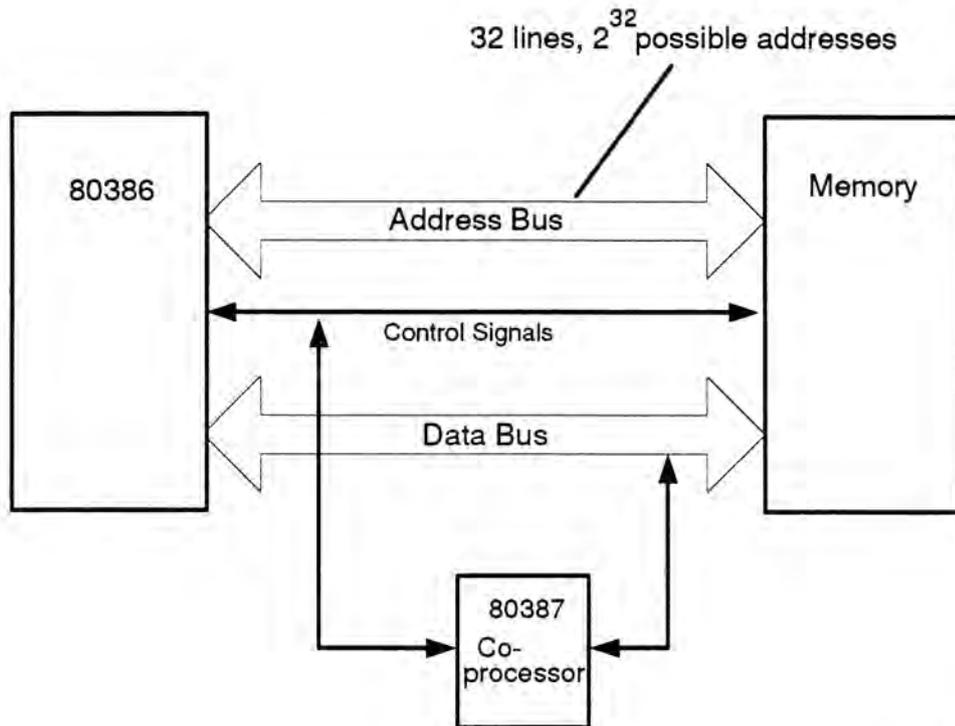
P1071302

Figure 3-2. Instruction Pipelining

Several instructions are performed simultaneously;

- fetching
- decoding
- execution
- memory management
- bus accesses

This parallel operation is called **pipelined** instruction processing. Each instruction is executed in stages and the processing of several instructions at different stages may overlap as shown in the figure above.



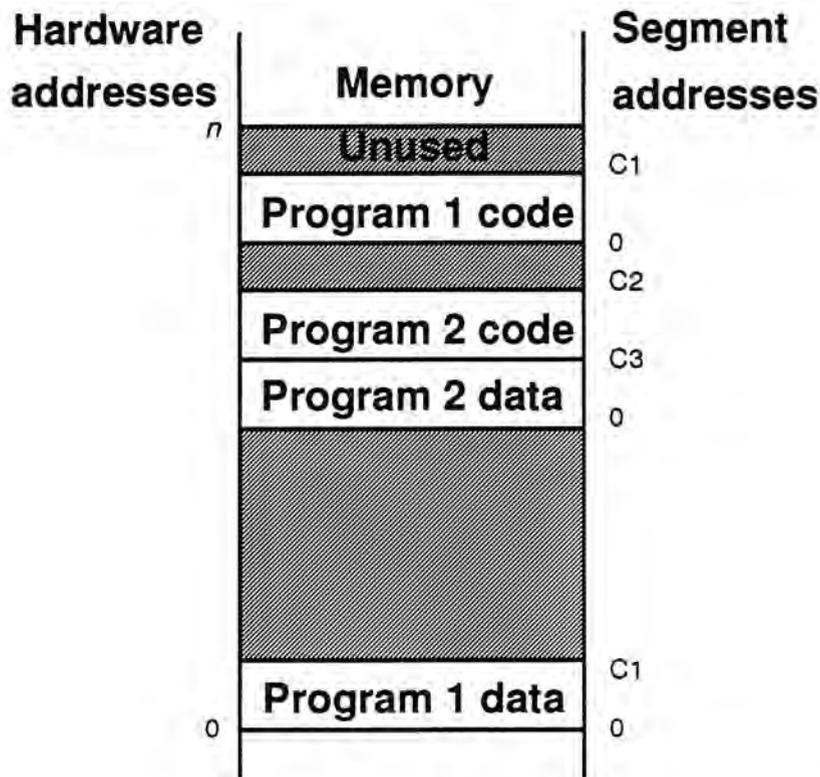
P1071304

Figure 3-3. Memory Interface

The interface between the hardware and the 80386 Central Processing Unit (CPU) is a set of address lines that go out from the processor to memory. The CPU places an address on the bus and memory responds by returning the value at the address or by accepting a new value for that address.

The address bus of the 80386 contains 32 lines. This capacity and the fact that the digital computer is binary in nature, means that a system with 32 address lines can reference anywhere from byte 0 to byte 2^{32} (2 to the 32nd power) in memory.¹ This linear addressing capability is consistent with the design of the hardware memory. The memory in the hardware allows access beginning at byte 0 and on to the last byte of memory in the system. This linear approach is known as the **flat** memory model.

¹ Figure and explanation by Ross P. Nelson, The 80386 Book.



P1071306

ds51

Figure 3-4. Segmented Memory.

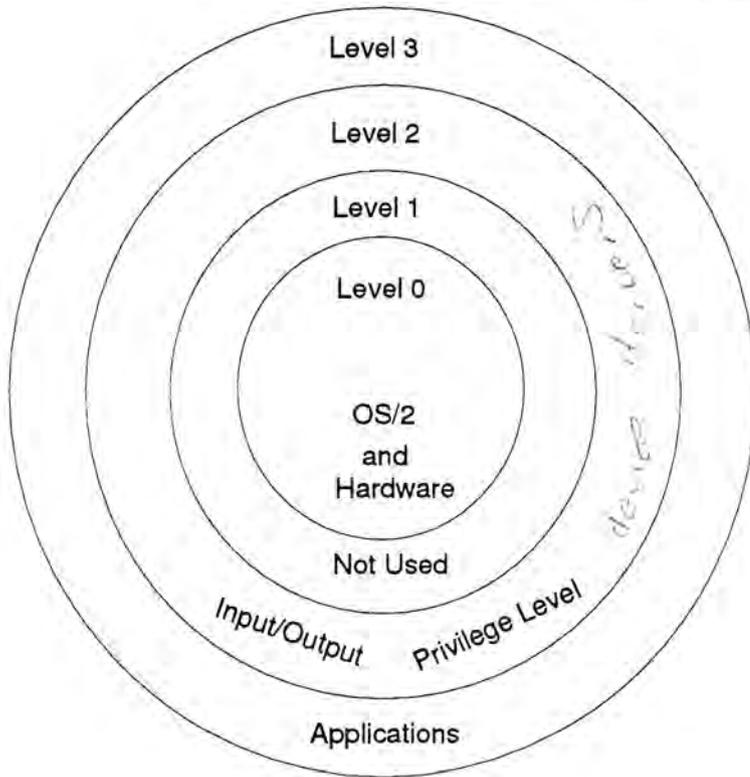
The 80386 has a memory model different from that of the hardware and of the bus design. The memory model of the 80386 is one in which memory is divided into chunks or **segments**.

A program can only access data contained in those segments. Within each segment, addressing is linear and an application program can access byte 0, byte 1, byte 2, and on to the limit of the segment.

Addressing is relative to the start of the segment and the hardware address associated with software address 0 (zero) is internal to the processor and hidden from the program.

Multiple programs are able to coexist within the computer's physical memory because in a multitasking environment, this breaking up of the memory into segments, isolates processes from one another.¹

Note: For our present purposes, the word "segment" is intended to denote a memory object and not any particular allocation size.



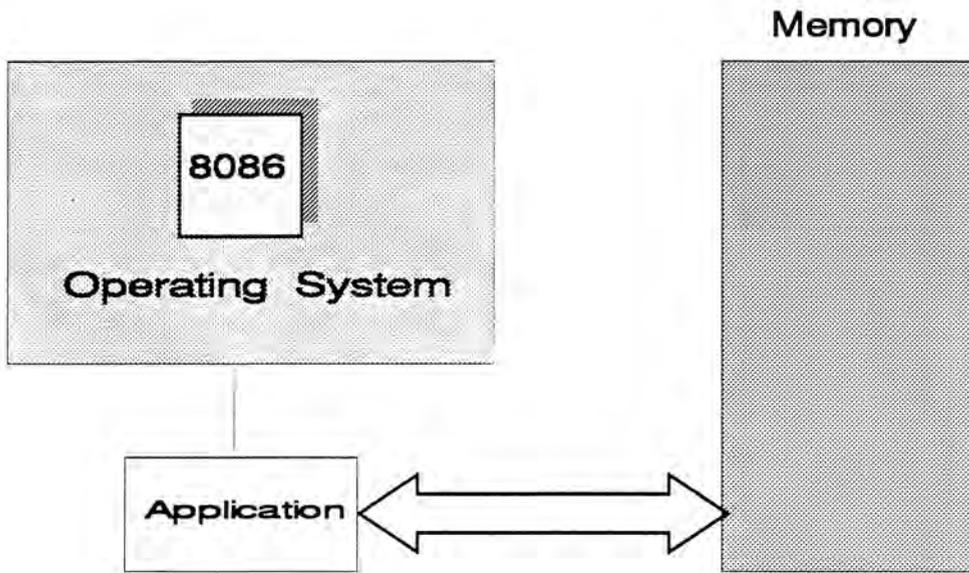
P1071308

Figure 3-5. Privilege Levels

The 80386 protection mechanisms are based on a hierarchy of privilege. There are four privileges ranging from level 0, the most privileged, to level 3, the least privileged. The four levels may be visualized as concentric rings with the most privileged level at the center. All data and code segments are assigned a privilege level. A task can execute under only one level at any given moment.

A task executing at one level cannot access data at a more privileged level (i.e., level 3 cannot access data at level 1) nor can it invoke a procedure at a less privileged level (i.e., level 1 cannot invoke level 3).

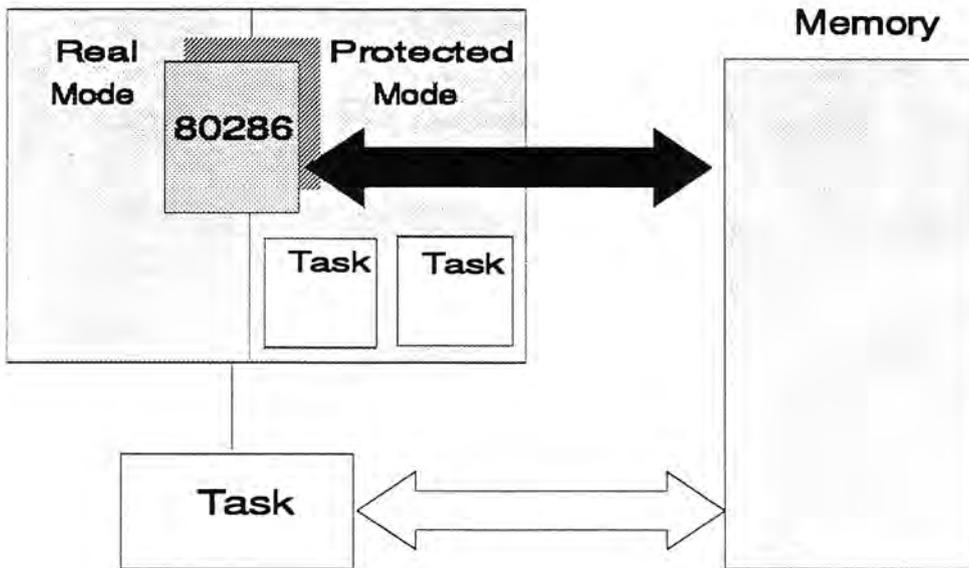
A memory object's privilege level is defined in the descriptor.



ps514008

Figure 3-6. The 8086 Environment.

An application developed for the 8086 runs in a single task, unprotected environment. The 8086 does not inhibit the application from accessing the hardware in any way.



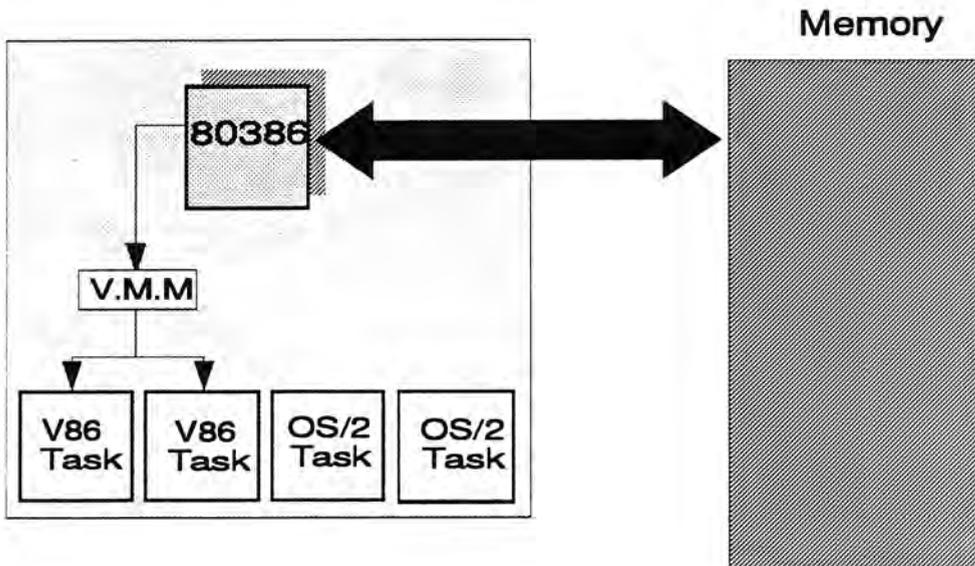
ps514010

Figure 3-7. The Real Mode Environment.

In protected mode, only the processor can access all memory or even issue instructions. Application program code is inhibited from accessing the hardware memory directly and is constrained to addressing a subset of the address space.

The program code for DOS applications does not behave as described above. To support DOS applications, the 80286 processor was designed with the capability to operate in **Real mode**. The 80286 could be switched to operate as a simulated DOS system in a compatibility process known as the DOS box. In Real mode, the processor behaves much like a fast 8086.

Programs that run in the DOS box can not be prevented from accessing the hardware out of turn because in Real mode, all programs share a common address space. The only way for Real mode applications to inhabit a system which operates Protected mode programs is to suspend Real mode operation when it is in the background.



ps514012

Figure 3-8. Virtual 8086 Mode.

The 80386 can execute 8086 programs in one of two modes:

- Real mode as already described, and
- in a sub-mode of the 80386's protected environment.

This special environment is actually a part of the **protected mode** environment provided by the 80386. Therefore, a protected environment now exists for 8086 programs. A **Virtual 8086 Mode**.

Attributes of a **V86** task:

- A V86 task runs at privilege level 3.
- A V86 task executes concurrently with protected 80386 tasks and other V86 tasks.
- A V86 task can utilize the 80386's paging and virtual memory capabilities.

Pentium

Dual Pipelines

8Kb cache for data

8Kb cache for code

each cache has its own TLB

64 bit data bus

pages can be extended to 2 MB and 4 MB

ps514016

Figure 3-9. The Pentium

The application instruction set of the Pentium processor includes the complete Intel 80486 instruction set with extensions to accommodate some additional functionality of the Pentium. All software within for the Intel 80386 and Intel 80486 will run on the Pentium without modification.

As shown above, the Pentium processor has several enhancements to increase performance.

Topic Summary

In this topic the student was taught the features of the 80386 and higher microprocessors that OS/2 2.1 has been designed to exploit.

The student should now be able to describe the following features of the 80386:

- Pipelining
- 32-bit memory management
- Multitasking
- Privilege level
- Virtual 86 Mode

This concludes the third topic
"The Design of OS/2".

TOPIC 4: OS/2 2.x Internals

Topic objectives

Terminal objective:

After attending this topic the student should have a general understanding of the implementation of the 80386's memory management in OS/2 V2.x.

Enabling objectives:

Upon completion of this topic the student should be able to highlight the internals of OS/2 2.x's:

- Memory management scheme
- Input output privilege level
- Dynamic Linking
- Process management
- Scheduler



Memory Management

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to explain OS/2 2.x's memory management scheme.

Enabling objectives:

After attending this subtopic the student should be able to explain:

- Memory addressing.
- Virtual memory management
- Concept of "thunking"
- Segment swapping

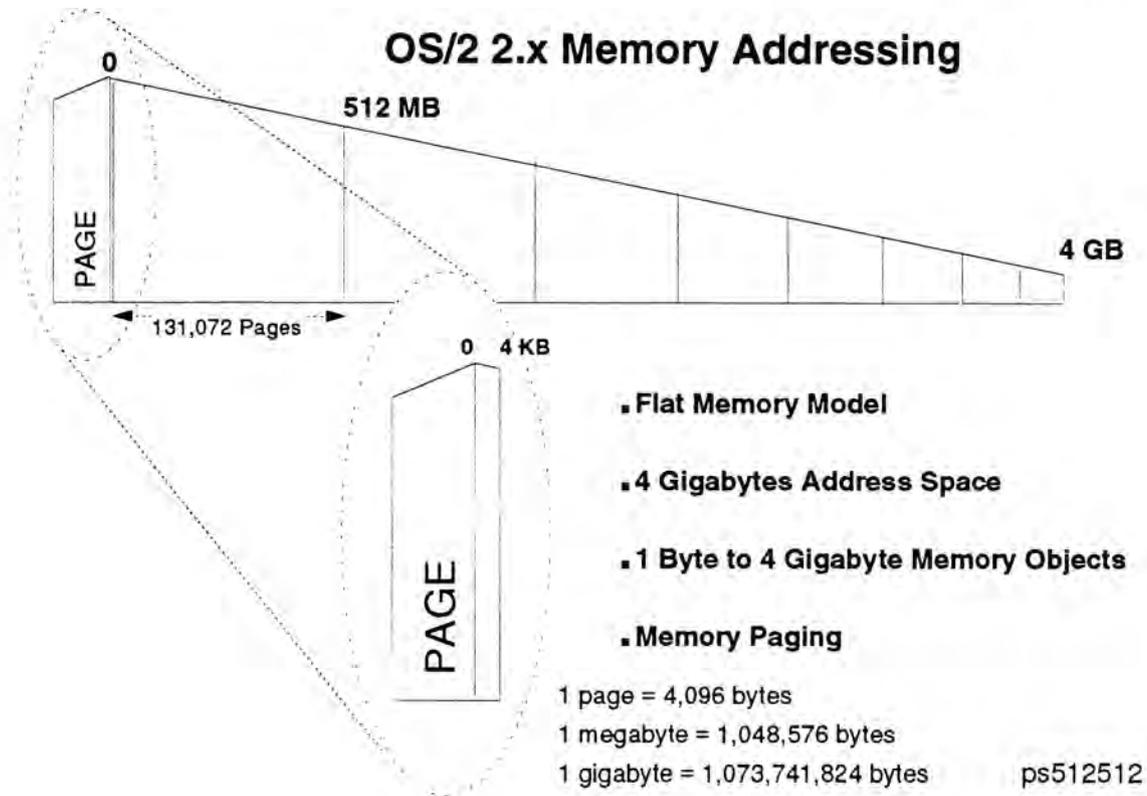


Figure 4-1. OS/2 2.x Flat Memory Model

In OS/2 2.x the name for a range of memory is "memory object".

OS/2 2.x utilizes the 32-bit addressing capabilities of the 80386. This addressing scheme allows 2^{32} possible addresses which equates to 4 Gigabytes of address range.

The memory model used by this version of OS/2 is the **Flat Memory Model**. "Flat" refers to the fact that the memory is conceptually seen as one large, linear address space of 4 Gigabytes.

If OS/2 2.x handled over-commitment the same way previous versions did, this same 4 Gigabyte size would potentially be used as the size of the swap segment causing severe problems due to the size. All available memory, both real and secondary, might not be enough to accommodate it.

A 4 Kilobyte memory object, therefore, is now the unit of memory on which all swapping and addressing is based. This 4 Kilobyte memory object is adopted from the mainframe environment and is known as a "PAGE".

Virtual Memory Management

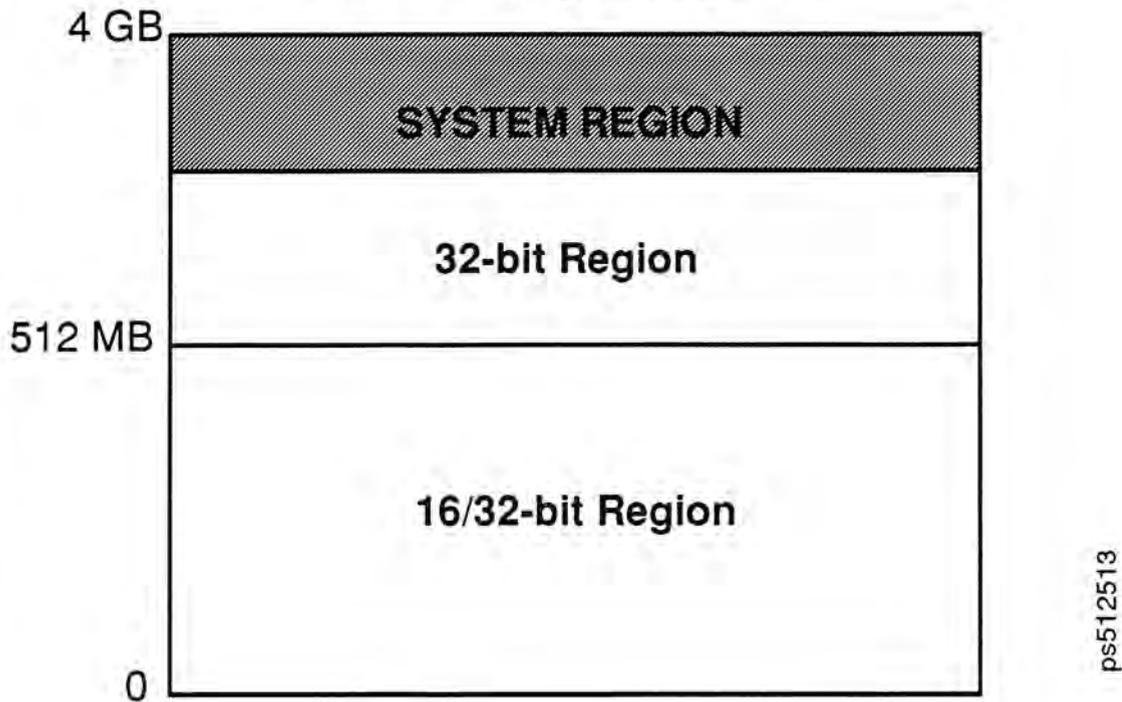


Figure 4-2. Virtual memory management

Each process has its own, distinct, address space. Theoretically, the space is 4 GB but the maximum is defined at system initialization and is somewhat smaller than 4 GB. It is smaller due to the system region memory objects. The operating system reserves the topmost layer to itself. Each process has a maximum address space of 512 MB. The OS/2 2.x implementation of the 32-bit 80386 has this limitation to accommodate 16-bit applications.

The address space below the system region is divided into two regions. The 16/32-bit region, which is found below the 512 MB boundary, accommodates applications that use the LDT addressing scheme.

Above 512 MB is the process address space addressable using **only** the 0:32 (zero base : 32-bit offset) memory scheme.

Virtual Memory Management - Process Region

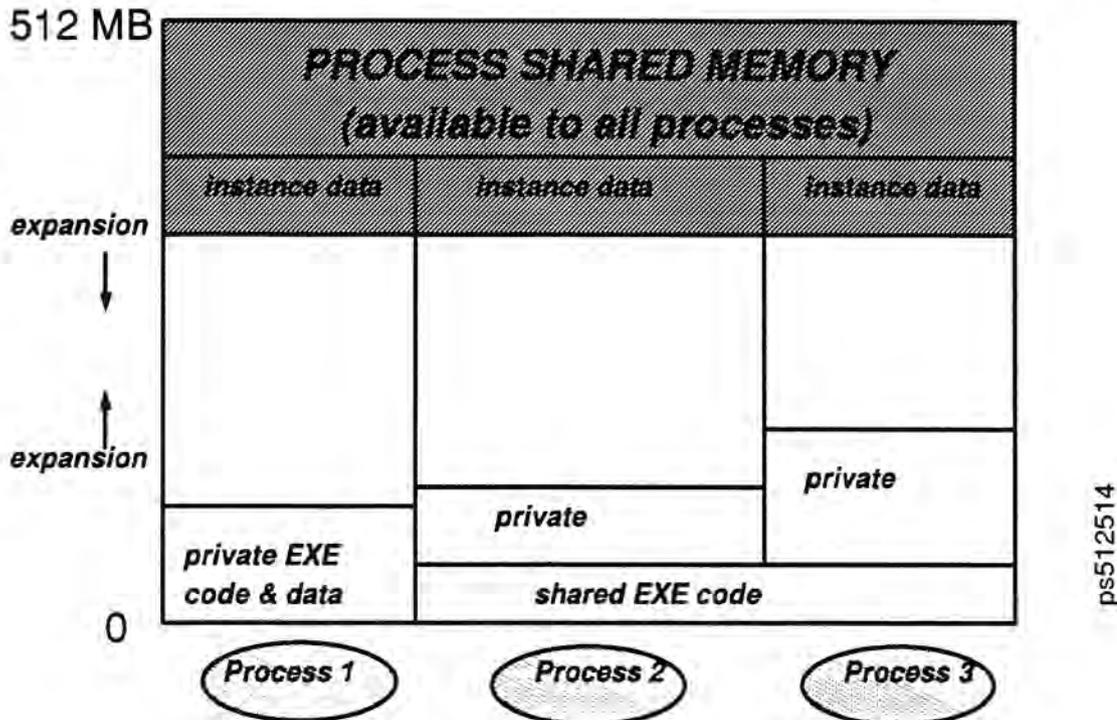


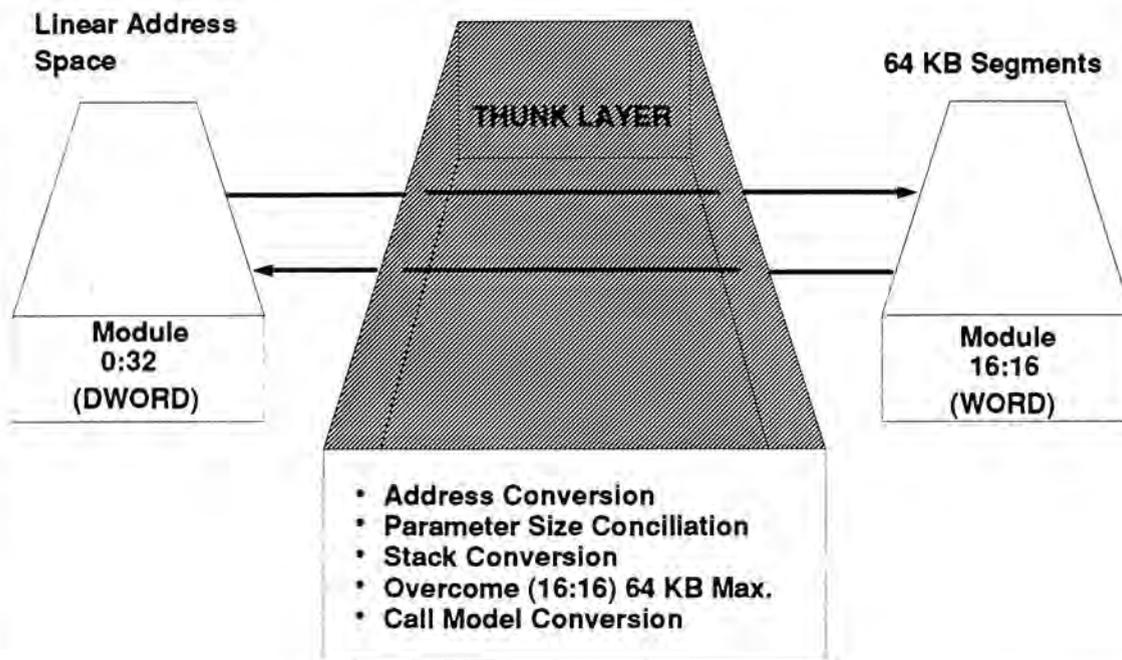
Figure 4-3. Process Region

The mixed 16/32-bit region of each process address space is further divided by OS/2 2.x into two regions:

- A **private** region that contains the EXE code and the process's private data.
- A **shared** region that is used to hold shared memory objects such as DLL code and shared data areas.

The **shared process region** is automatically reserved at the topmost part of the address space of all executing processes and expands downward in memory as new elements are loaded by the process.

The **private process region** expands upward in memory.



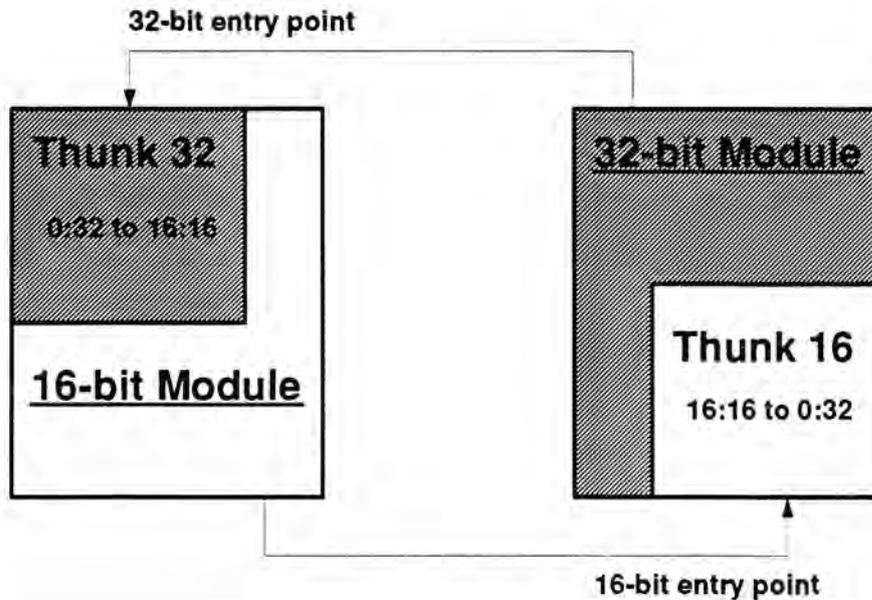
ps512515

Figure 4-4. Mixed Environment Considerations

One of the major concerns in implementing this 32-bit version of OS/2 was ensuring backward compatibility with 16-bit applications. The kernel uses 0:32 addressing but few, if any, applications currently use true 0:32. However, as time passes, there will be 0:32 applications and some considerations of compatibility must be taken into account:

- Running 16-bit applications in a 0:32 environment.
- 0:32 applications utilizing 16:16 procedures.
- 16:16 applications utilizing 0:32 procedures.
- Large memory objects (over 64 KB).

All of the above are the responsibility of special routines called **thunks** which are designed to properly handle the differences.



ps512516

Figure 4-5. Thunks

Source modules are compiled to produce object modules. As there are different entry points for each module (as illustrated in Figure 3-8), it is possible to mix 16-bit and 32-bit function calls in an application.

Within each module is a thunk layer that is packaged with the module. The layer contains the supporting code in 16-bit APIs for a 32-bit entry point and the supporting code in 32-bit APIs for a 16-bit entry point.

Thunks are available for:

- Executable programs (.EXE)
- Import libraries (.LIB)
- Dynamic link libraries (.DLL)
- Presentation Manager messages.

Managing Pages



Fixed

Discardable

Swappable

ps512517

Figure 4-6. Managing Pages

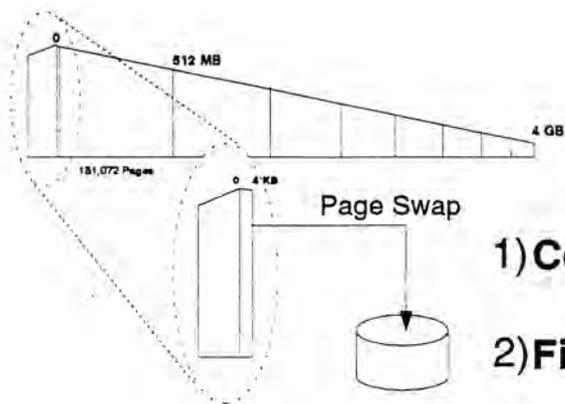
Pages can have the following types:

- Fixed
These pages are permanently resident in memory.
- Discardable
It is possible to reload these pages from either an EXE or DLL file. When memory becomes over-committed, space used for discardable pages can be freed up, and when the page is required again they are reloaded from the original file.
- Swappable

When there is a shortage of memory, these pages can be swapped to the hard drive.

The operating system needs more information over and above those contained in the Page directories and page tables to manage the paging process. OS/2 2.x builds three arrays of data structures that represents:

1. Committed pages in the process and system address spaces
2. Pages in memory
3. Pages held in the 'pager'



1) Count free pages

2) Find least recently used pages

3) SWAP LRU to disk

4) Swap pages to designated slots

5) If swapper.dat too small, add 512 KB

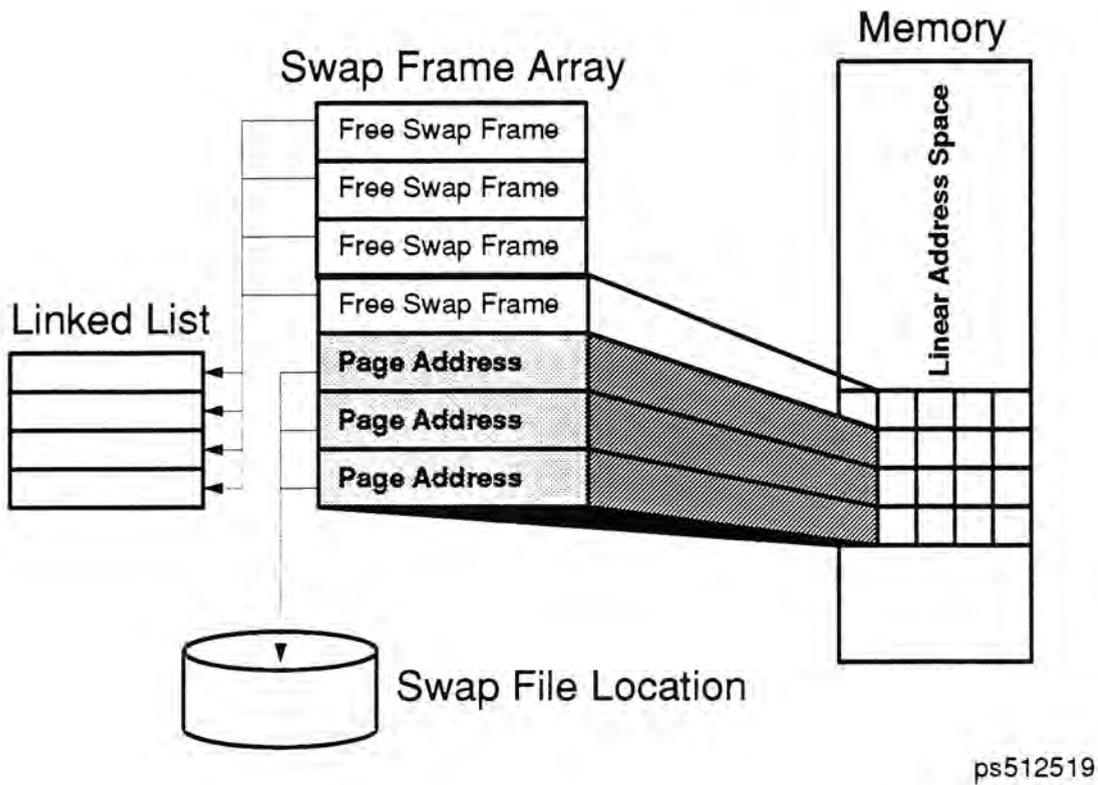
6) If swapper.dat too large, reduce by 512 KB

ps512518

Figure 4-7. Page Swapping

Swapping that is based on pages instead of segments is implemented somewhat differently in OS/2 2.x. The swapping algorithm is much simpler and easier to maintain and is designed to improve performance.

The most apparent difference is the SWAPPER.DAT file. It is managed dynamically. It will not simply grow indefinitely until there is insufficient disk space to accommodate it. When an application is terminated, the disk space being used by its swapped pages is recovered and the SWAPPER.DAT file actually shrinks. The operating system determines when to shrink it and by how much in terms of 512 KB.



ps512519

Figure 4-8. Page Swapping and the Swap File

Space in the SWAPPER.DAT file is managed on the basis of a **swap frame array** where specific slots are reserved for each swapped page.

A page being swapped out to disk for a second or subsequent time, is always returned to its assigned slot unless the entire swap file is reorganized to reduce its size. In that case, the entire swap frame array is rearranged.

The system maintains a linked or chained list (a list in which each data element contains information for locating the next element) of free swap frame slots. The linked list improves file access speed when swapping is required.

The SWAPPER.DAT file is created at system initialization with a size based on available disk space and RAM memory.

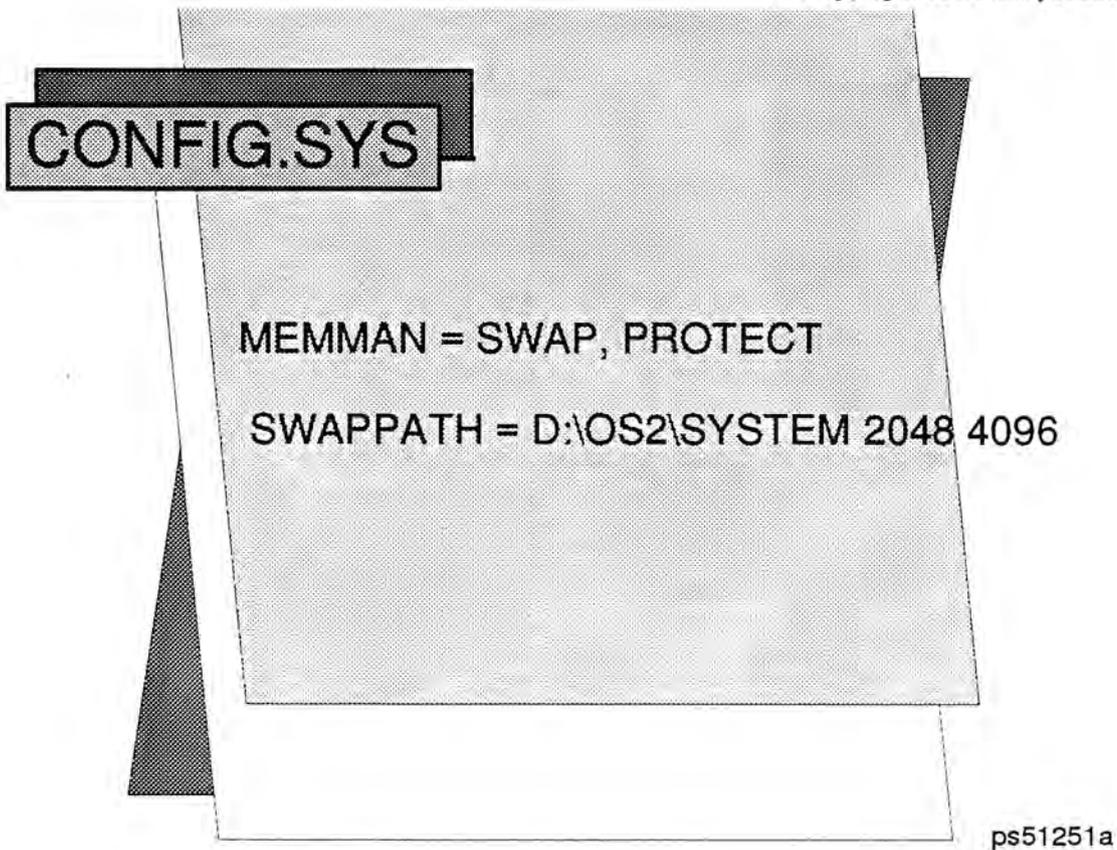


Figure 4-9. Configuration file parameters

The *MEMMAN* parameter, selects memory management options for the OS/2 environment.

The swap file is used to temporarily store data that the system has removed from memory to satisfy a request for memory. The default path of the location of this file can be altered to another sub-directory or another partition or even another physical hard drive. After the path, the next value is the *minfree* parameter. The third value is the *initial* size of the file upon system startup.

- *Minfree*

This parameter specifies the minimum free space that can remain on the partition before you will receive a warning that the swap file has increased to a size that leaves less than this amount of free space on the partition.

- *Initial*

The initial size indicates how big the swapper.dat file is to be upon starting the system.

Subtopic Summary

In this subtopic the student learned the highlights of OS/2 2.x's memory management scheme with respect to the following:

- Memory addressing
- Virtual memory management
- Concept of "thunking"
- Segment swapping

This concludes this subtopic of the topic
"OS/2 2.x Internals".

commit
- don't use
app's reads and commits all
- the memory

put jar swap file

on the most used partition

on the least used drive.

Input Output Privilege Level

Subtopic objectives

Terminal objective:

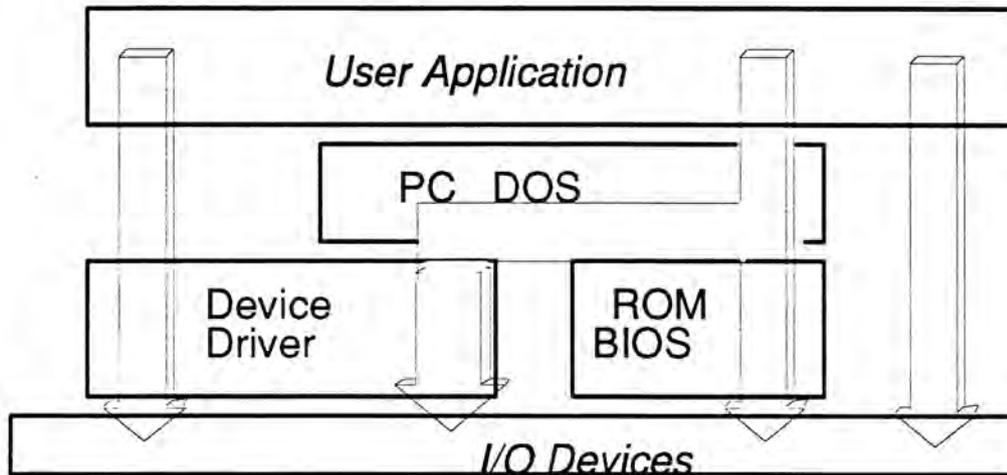
After attending this subtopic the student should be able to explain the purpose of the IOPL statement in the OS/2 configuration file.

Enabling objectives:

After attending this subtopic the student should be able to explain the

- Difference between IOPL = yes and IOPL = no
- Purpose of the Input Output Privilege Level parameter

Structure of PC DOS



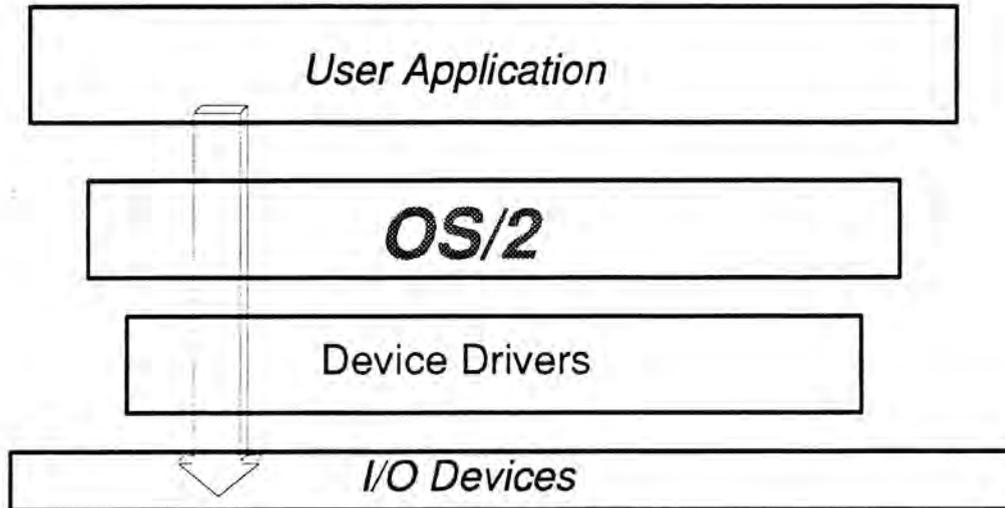
ps512521

Figure 4-10. Structure of PC DOS

In a single tasking environment, such as DOS, applications do not share resources with other applications. Orderly access to a device is not a problem because a device can be used by only one application at a time. As such, you will find the following types of DOS applications:

- A DOS application can communicate to an I/O device by going through PC DOS which in turn will go to the appropriate BIOS/Device Driver which in turn goes to the appropriate hardware.
- A DOS application can contain certain software such that it avoids PC DOS and communicates to either BIOS or Device drivers directly which in turn will talk to the appropriate device driver.
- A DOS application can contain certain software such that it totally avoids PC DOS, BIOS, and Device Drivers for a specific hardware and communicates to it directly.

Structure of OS/2



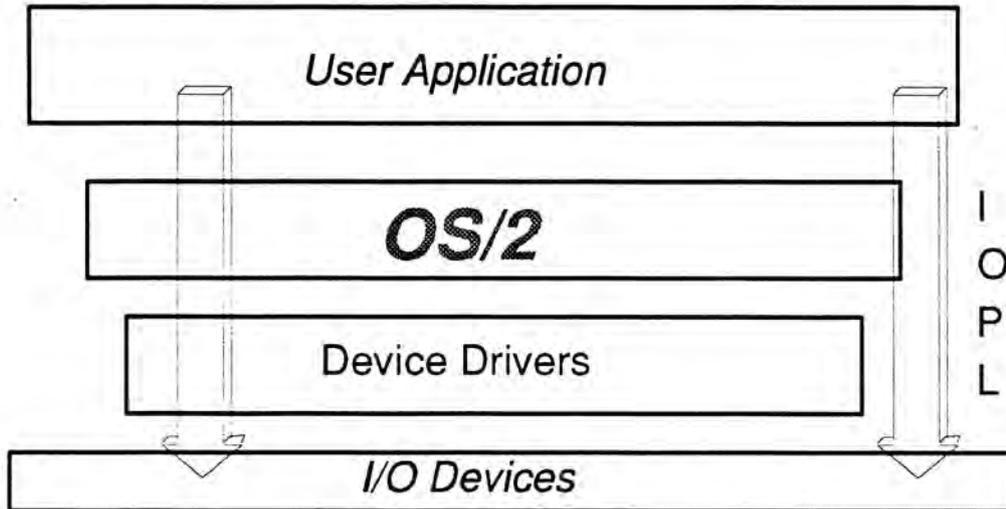
ps512522

Figure 4-11. Structure of OS/2

Because OS/2 is a multitasking operating system, DOS mode and OS/2 mode applications share the same resources. Applications running in the OS/2 environment that directly access a device, therefore, may impact the system or another application. Because of this, applications in the OS/2 environment should access devices through the device drivers.

A device driver guarantees orderly access to a device by manipulating the device on behalf of all applications sending data to or receiving data from the device. To maintain the integrity of the system, therefore, applications should access the device through the device driver.

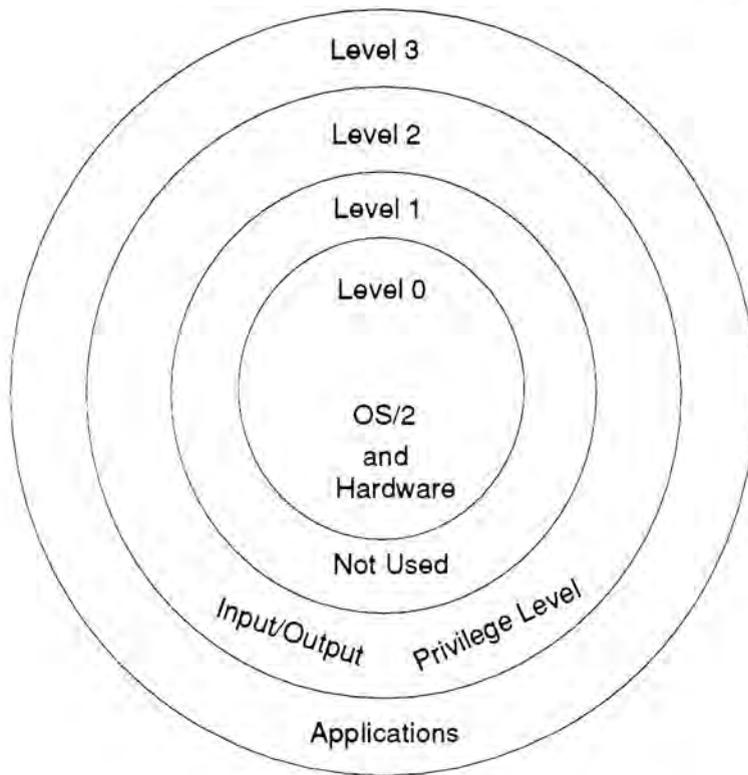
Allowing IOPL



ps512524

Figure 4-12. Permitting IOPL

Programs that are granted I/O privilege run at privilege level 2. A program assigned privileged level 2 (such as a subsystem of an application) that needs to communicate directly with a specific device, is permitted to send or receive instructions to or from that device.



ps512523

Figure 4-13. Existing Privilege Levels

This drawing was first introduced in the 80386 architecture lecture.

Privileged level 0 OS/2 and device drivers

Privileged level 1 Reserved

Privileged level 2 Special purpose routines other than device drivers requiring I/O privileged

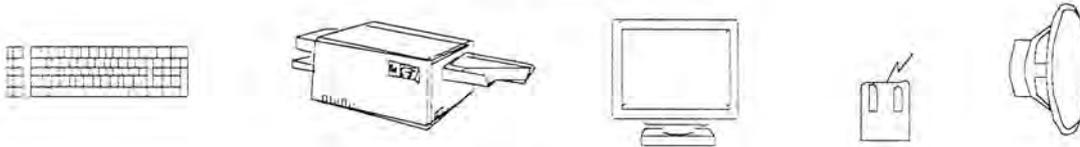
Privileged level 3 User applications

some DLLS

Recall, from the 80386 presentation that at any special level in the hierarchy, only programs at that level or a more privileged level can access data at the specified level. The 80386 processor does not allow applications to execute IN/OUT instructions. Applications run at privileged level 3 thus cannot execute these instructions. However, OS/2 allows applications to execute these instructions from the IOPL code segments at privileged level. Thus for an application to move data to and from a hardware port, it will require an IOPL code segment.

I/O Devices

Character Devices



Block Devices



ps512525

Figure 4-14. Sample I/O Devices

Here is a list of some of the device drivers that come with OS/2. Some of these devices are referred to as Character devices and the others as Block devices. Character devices are considered simple enough to permit IOPL to access them.

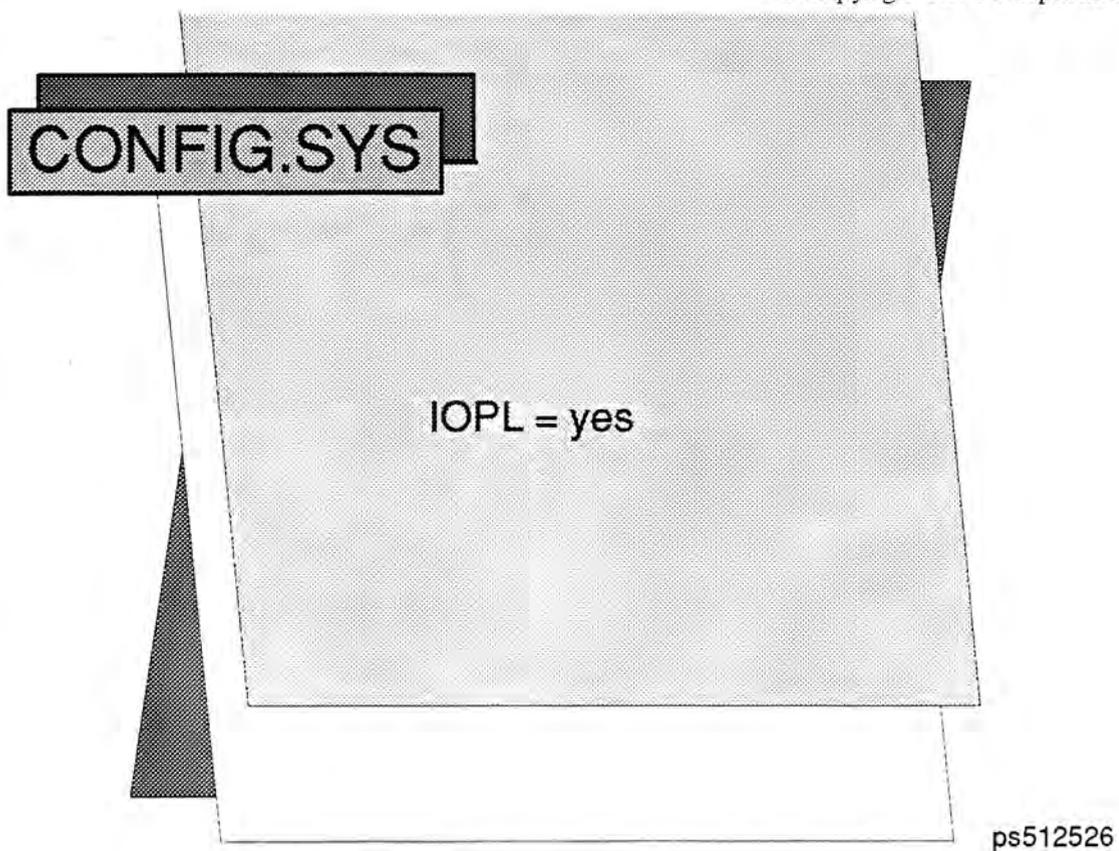


Figure 4-15. Configuration file parameter

Allows I/O privilege to be granted to requesting processes in OS/2 sessions.

This parameter if set to **No** will not permit any OS/2 application to use their IOPL code segments.

Subtopic Summary

In this subtopic the student learned the purpose of the IOPL parameter found in the OS/2 configuration file.

This concludes this subtopic of the topic
"OS/2 2.x Internals".

Dynamic Linking

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to explain the purpose of dynamic linking.

Enabling objectives:

After attending this subtopic the student should be able to explain the

- Basic concepts of Dynamic Linking
- Importance of the LIBPATH parameter

Dynamic Linking

***A method that allows preloading of segments
before application execution OR on demand***

ps512531

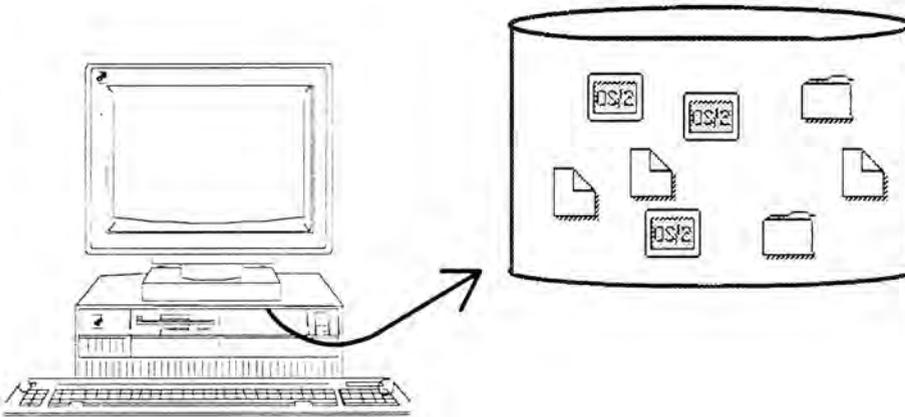
Figure 4-16. Definition

Under DOS, an executable file will contain all of the object code files, or modules that make up the application. This includes error handling and exception handling routines that may not be regularly be required. When a program is invoked, its entire execution module is loaded into memory. This works well but can result in large executable modules. In a single tasking system this is not often a problem, but in multi application systems where memory over-commitment is the norm these large modules can be a problem.

OS/2 provides facilities that allow a programmer to specify what part of the code is to be loaded at the start of execution. If the other functions are required then they are loaded into memory on demand. This method of linking, where not all code segments are loaded into memory at the start of program execution is called *dynamic linking*. Instead of grouping together all of the executable code needed by an application into one executable file, OS/2 with Dynamic Linking can keep track of the other code segments and bring them into memory when required.

Dynamic linking allows applications running in the system to make efficient use of memory by delaying the resolution of external references. This feature enables an applications to call a subroutine that is not part of its executable file, thus making the size of the executable file smaller. The subroutine called by the application can also be called by other applications. In addition, an application can decide whether or not to call a subroutine, depending upon events that occur at run time.

Dynamic Linking



ps512532

Figure 4-17. Files, processes, directories

The icons represented in the above picture, are processes, files and, directories. These are stored in either the hard drive of your system or in a diskette.

Dynamic Linking

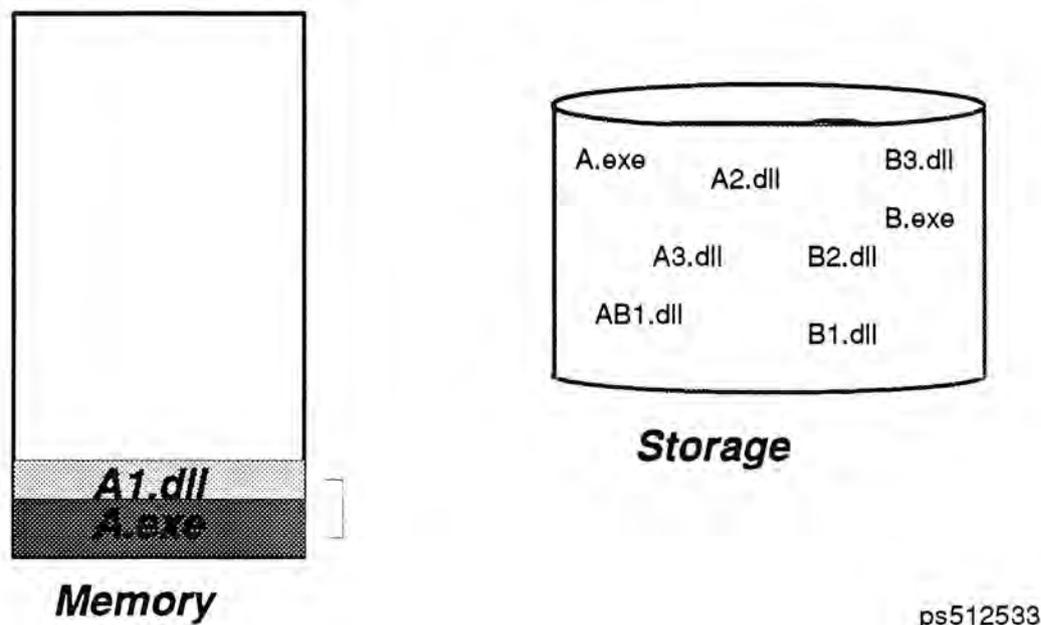


Figure 4-18. Starting Process A

DLLs are bound to an application when the application loads, or later, during execution of an application. When an application loads a DLL on demand, this reduces the need to use system memory until absolutely necessary. An application can release a DLL and thus free up the memory it used.

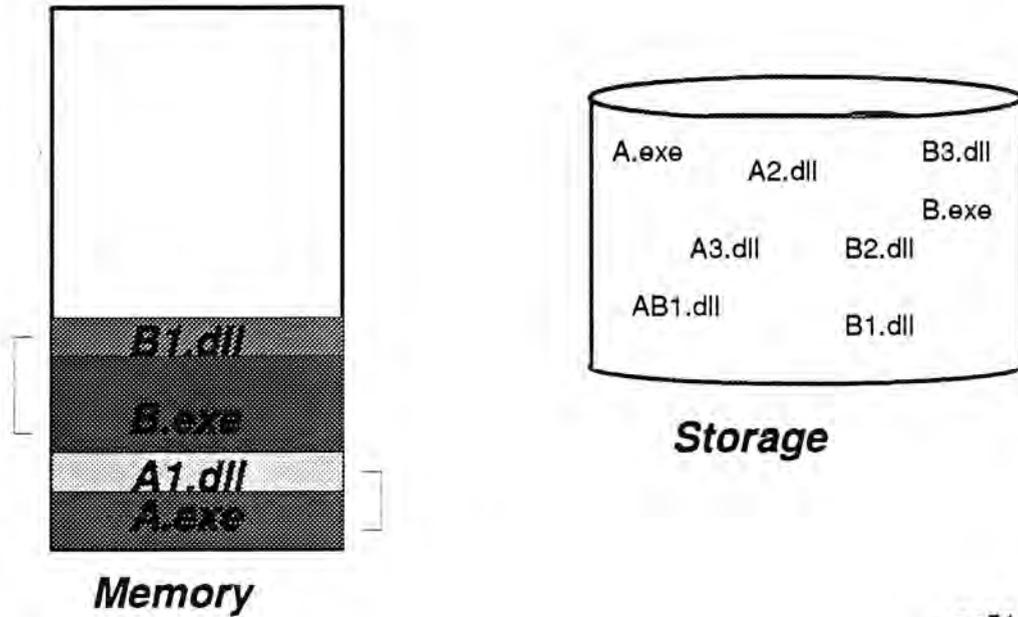
Assume that the storage contains 2 programs: **A.exe** and **B.exe**.

Also assume that program **A** has 4 DLLs: **A1.dll**, **A2.dll**, **A3.dll** and, **AB1.dll**.

Assume that program **B** has 4 DLLs: **B1.dll**, **B2.dll**, **B3.dll** and, **AB1.dll**.

Upon starting program **A** the program has been written to bring in **A.exe** as well as **A1.dll** into memory.

Dynamic Linking



ps512534

Figure 4-19. Starting Process B

An application interacts with a DLL as it would with any other part of the operating system.

Upon starting program **B** the program has been written to bring in **B.exe** as well as **B1.dll** into memory.

At this time both programs **A** and **B** have been started.

Dynamic Linking

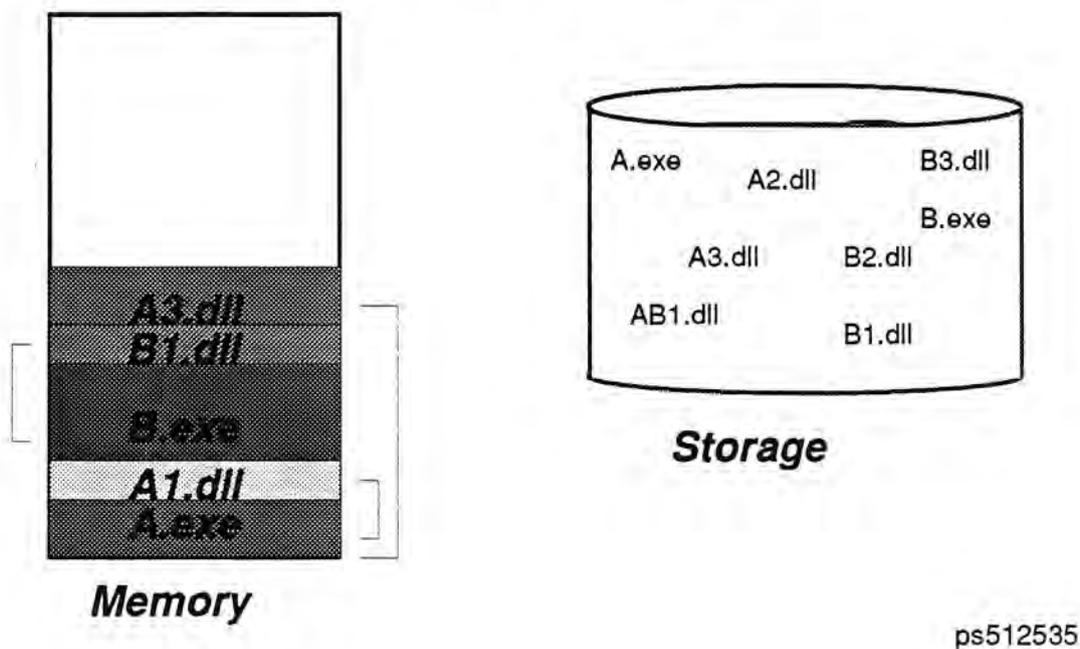
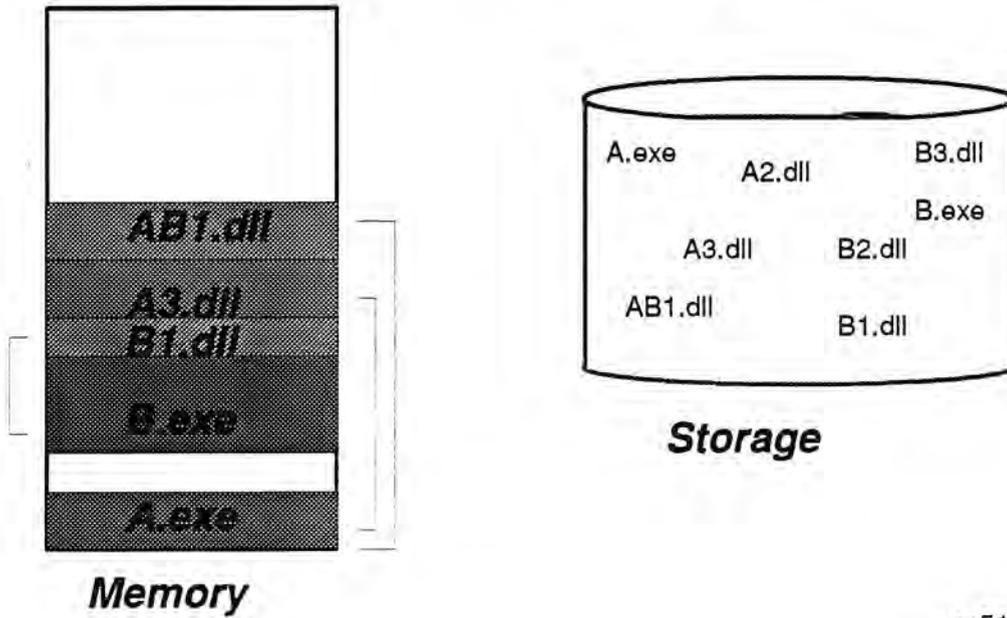


Figure 4-20. Using Process A

As the user is using program A, assume that they decide to use a particular option of the program. Assume, this option has been programmed in **A3.dll**. Seeing as this option is not currently in memory the system is responsible for getting it in memory.

Dynamic Linking

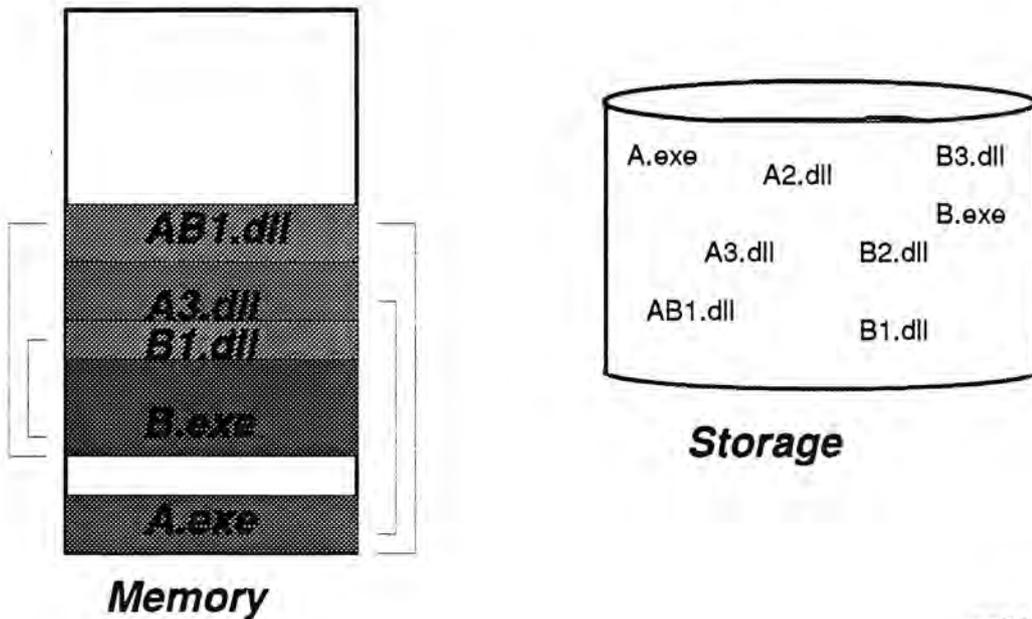


ps512536

Figure 4-21. Using Process A

As the user is using the program **A** assume that they decide to use another option of this program. This option has been programmed in **AB1.dll**. Seeing as this option is not currently in memory the system is responsible for getting it in memory.

Dynamic Linking

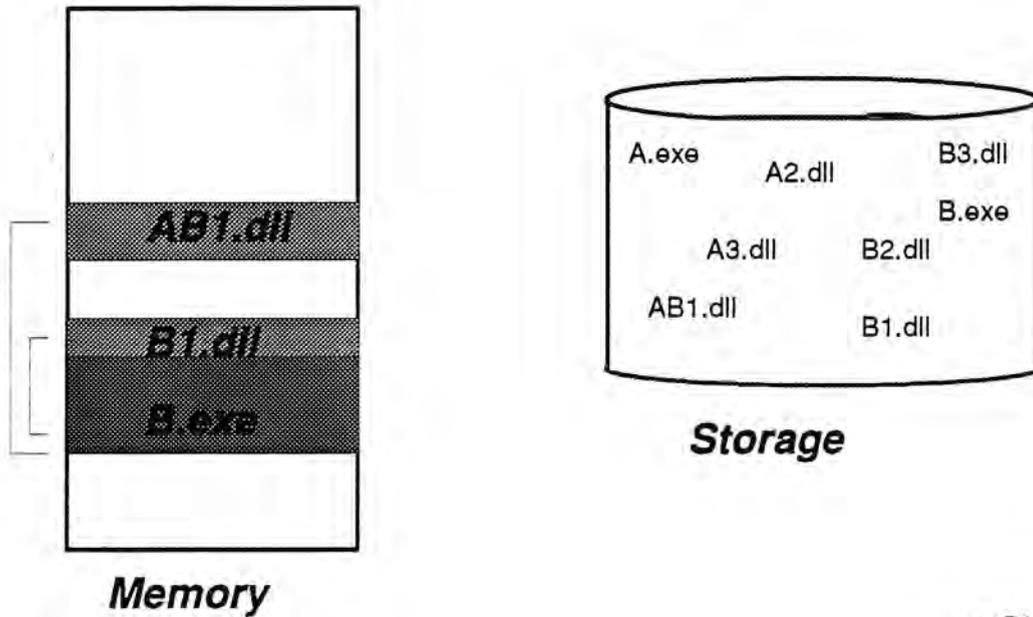


ps512537

Figure 4-22. Sharing a common DLL

The ability is there, to share common code. For instance, program **B** may have been written such that one of its options is identical to that of program **A**. Rather than duplicating the code the same piece of code can be used by more than one program. The example shown here is **AB1.dll**. This can be used by both programs and need not be duplicated in memory, thus saving on memory requirements.

Dynamic Linking



ps512538

Figure 4-23. Ending Process A

The user can end program A. Although that program was responsible for bringing in **AB1.dll** it will not be 'erased' until program **B** is completed using it.

Dynamic Linking

Provides general access for many processes

Each DLL need only be loaded once

Prevents redundancy

ps512539

Figure 4-24. Dynamic Linking

Consider an application that contains extensive help and tutorials. If this application was statically linked, then these sections of code would always be in memory even if not used. If the Tutorial was placed in a DLL, then only when/if it was required would that section of code actually be brought into memory.

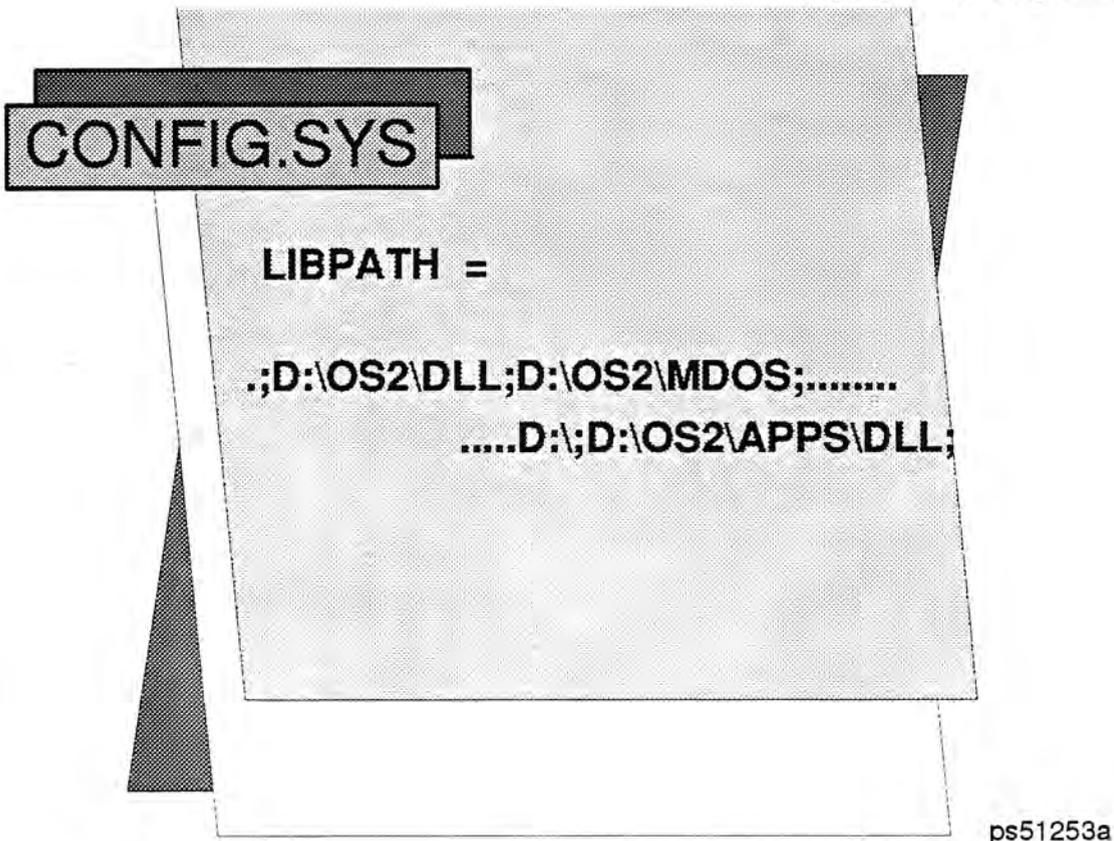


Figure 4-25. LIBPATH statement

This statement identifies the locations of dynamic link libraries for OS/2 programs.

This statement will be created by the installation of OS/2 2.x. As one installs other programs, this statement will get updated either by the User or by the Installation program of the new program.

The LIBPATH is used to identify a set of directories to be searched when the OS/2 operating system loads dynamic link libraries.

Subtopic Summary

In this subtopic the student learned the existence of Dynamic Link Libraries and their purpose.

This concludes this subtopic of the topic
"OS/2 2.x Internals".

Process Management

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to explain how the OS/2 manages your processes and explain the scheduler functions.

Enabling objectives:

After attending this subtopic the student should be able to explain the

- Basic concepts of threads, processes and the OS/2 scheduler
- Explain the purpose of THREADS, MAXWAIT and, the PRIORITY parameter in the OS/2 configuration file



Terminology - Process

Represents an application that is currently started

Can consist of one or more threads

Is a unit of resource ownership

OS/2 will manage resources required by the process

ps512541

Figure 4-26. Terminology - Process

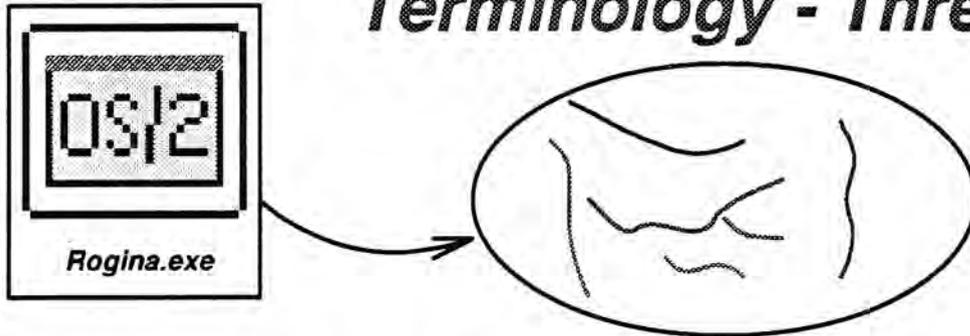
A process is the basic unit of programming and resource sharing in OS/2. A process responds to a program and is created when a program is loaded. A program consists of a series of instructions.

When started, each process is assigned a unique process identifier (PID) by OS/2.

The system maintains many resources on a per process basis. The primary resources contained in a process are its memory and the threads of execution.

STAT - MAIL

Terminology - Thread



Is the smallest unit of execution

Can access resources by its Parent Process

Maximum of 4K threads per system

***Its Parent Process is responsible for acquiring
the needed resources***

ps512542

Figure 4-27. Terminology - Threads

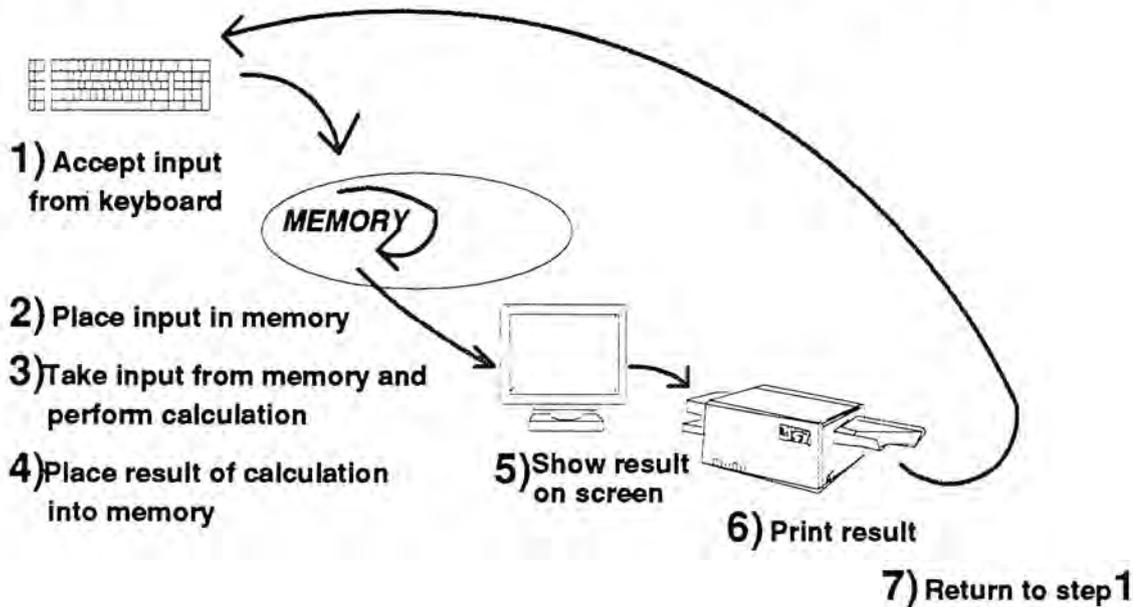
An application consists of a series of instructions. As the processor passes through these instructions, it creates a **thread** of execution. A **process** is a collection of one or more threads that own resources, such as memory, open files, devices or connections to DLLs. Threads share all the resources owned by the process that creates them.

Threads are dispatchable units within OS/2 processes. Processes do not really run, but threads do. A thread provides, within a process, a piece of code with instructions for a specific function.

The ability to divide an application into multiple asynchronous threads, which can communicate rapidly through a common data space is an extremely powerful feature of OS/2.

4095 threads
per OS/2
system.

A Process with a SINGLE thread

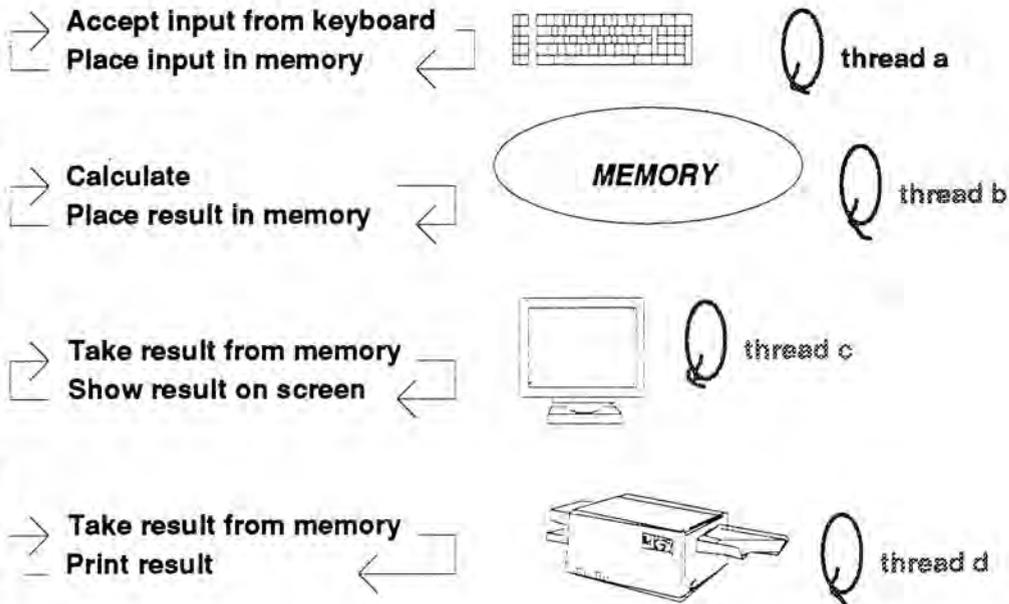


ps512543

Figure 4-28. A process using a single thread

The above example indicates how a process would react if there were only one thread to perform all functions. Each step can only be performed one at a time one after the other. The user after entering input through the keyboard would have to wait until steps 2 - 6 were completed before being given 'control' again.

Process using multiple threads



ps512544

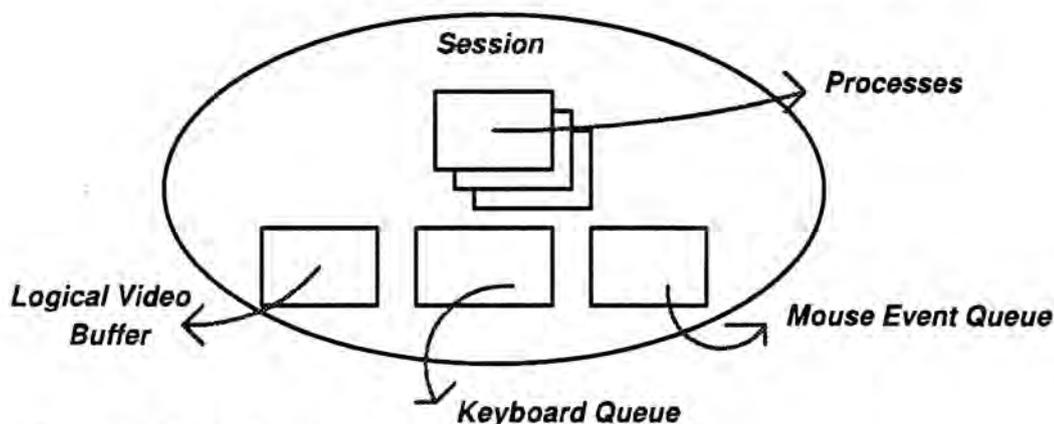
Figure 4-29. A process using multiple threads

Multiple threads promote greater overlapping of I/O requests. A multithreaded system is able to be more interactive with the user than a single-threaded one, due to the greater level of concurrency achieved.

Programs with multiple threads often dedicate a single thread to servicing requests from the user interface (keyboard, mouse etc..) while other threads are actually performing the work requested by the user.

Multiple threads better support an environment where parallel applications can execute with a far better performance than is possible in a single thread process.

Terminology - Session



Unit of user interface

Represents logical devices-

Display, Keyboard, Mouse and related processes

Is managed by the OS/2 session manager

ps512545

Figure 4-30. Terminology - Session

Sessions are managed by the session manager of OS/2. Each session contains a **logical display**, a **logical keyboard**, a **logical mouse**, and any processes that share that logical user I/O devices.

The process that is running in the *foreground* has the input focus of the user input devices. Only one process can be in the foreground at a time.

Processes that are not in the foreground are said to be in the *background*.

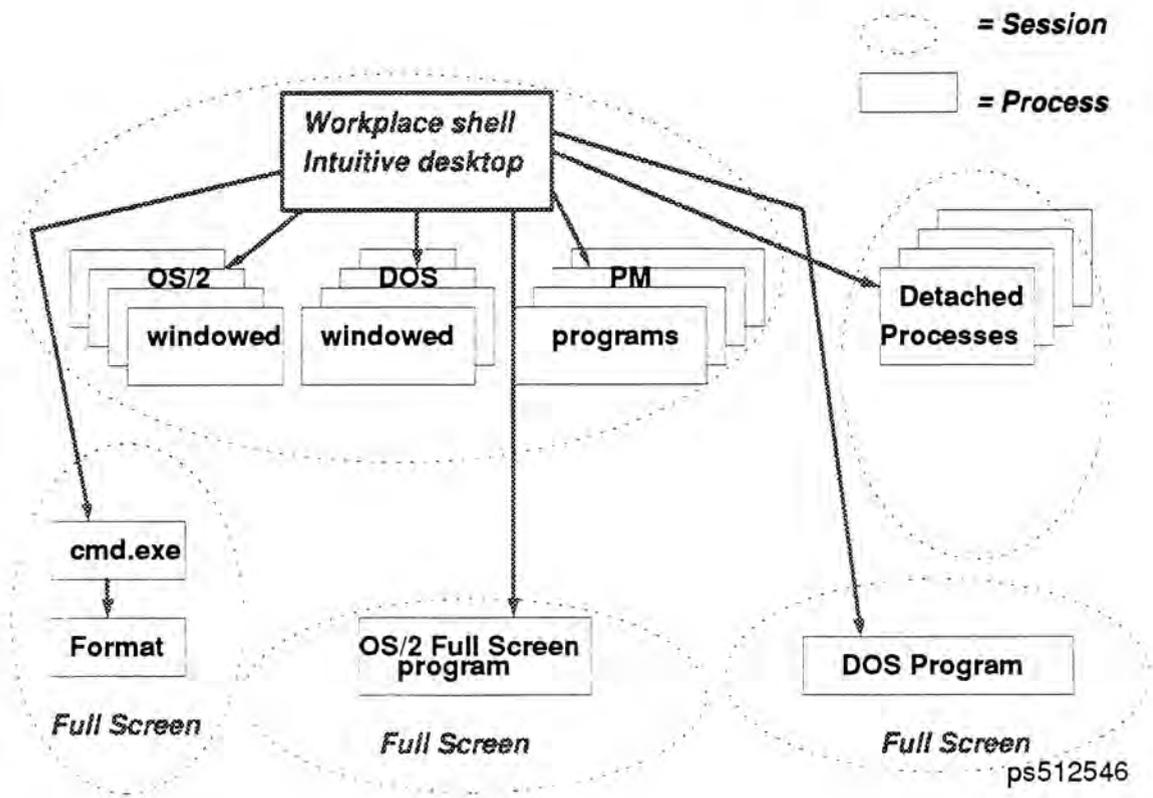
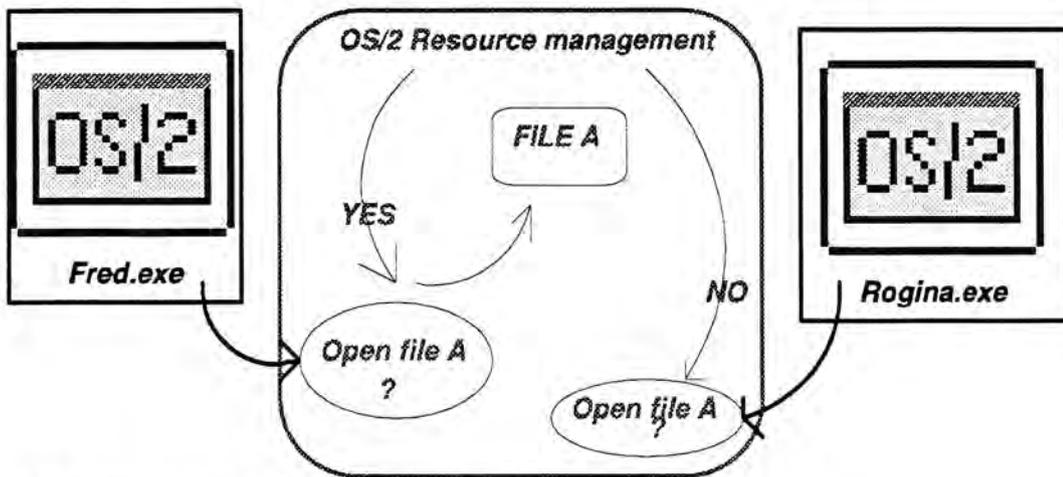


Figure 4-31. Sessions

The appearance of full screen sessions to the user is like having multiple full-screen console. In this case, console, refers to a screen, keyboard and a, mouse.

A single **detached session** contains background processes that have been detached. Processes running in this session, run without the user I/O devices. Generally, programs that are usually asleep and wakeup occasionally to perform some minimal duty run in this session.

Resource Management



OS/2 controls access to each resource

**OS/2 manages information about the resources
for each process**

ps512547

Figure 4-32. Resource Management

*Simpler
page mechanism
of windows*

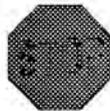
Execution Control

Multitasking is provided by a

Time slicing ,



Pre-emptive



Scheduler



Scheduling is done on a per thread basis

All threads in the system compete for processor time

ps512548

Figure 4-33. Execution Control

Several threads can be ready to execute at the same time, but only one thread at a time can have access to the processor. Access to the processor is managed by the system scheduler, which assigns each thread a priority. The thread that has the highest priority, and that is ready to run, is allocated to the processor.

If a thread is being processed and another thread with a higher priority becomes ready to run, the system stops processing the thread with the lowest priority and allocates the processor to the thread with the highest priority.

Handwritten notes:
SCHEDULER
PRIORITY

Scheduling



An active thread is made inactive:

at the end of its timeslice

if it must wait for an event to occur

if a higher priority thread is ready

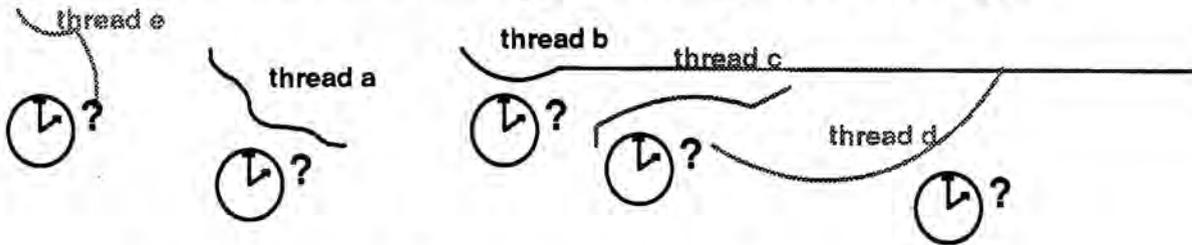
ps512549

Figure 4-34. Scheduling

Each thread has its own execution priority. When a thread is ready to run, at some moment in time, it will be given a certain amount of time to work with the processor to complete its job. If a thread is given a time slice with the processor and must wait for an event to occur before continuing its work, the remaining timeslice (if any) will not be wasted.

41 prioritized class
32 per class

Scheduling - Dispatching



The choice of which thread to activate

depends on its priority class

and on its priority level

ps51254a

Figure 4-35. Scheduling - Dispatching

High priority threads that are ready to run will be dispatched before low priority threads that are ready to run.

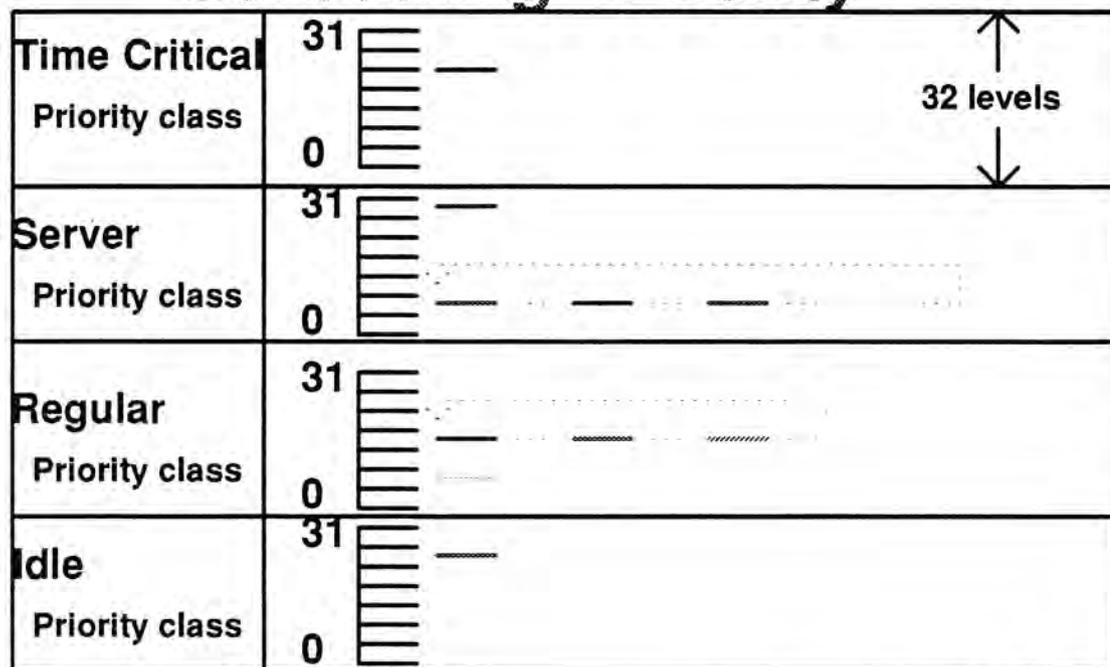
A thread will belong in one of 4 classes of priority:

- Time Critical
- Server
- Regular
- Idle

Within each of these 4 classes there are 32 levels of priorities. The highest priority in a class is level 31.

The scheduler determines how to distribute execution control among the running threads. It uses time slicing to distribute the execution controls.

Scheduling - Priority



ps51254b

Figure 4-36. Scheduling - Priority

Threads in the *Time Critical* class are ones that must react quickly to events outside the system.

Threads in the *Server* class are used for programs that run in a server environment that need to execute before regular priority class processes on the server.

Idle priority class threads will only run when there is nothing to run in either of the above mentioned classes.

OS/2 implements a multilevel priority scheduler with **dynamic priority capabilities** and **round robin scheduling**.

Most threads in the system are in the regular priority class.

Threads sharing the same level in the same class will have the scheduler take care of them in a round robin fashion and will give each one of them a timeslice.

To ensure that the system is responsive to the user requests, all the threads in the foreground process receive a boost in priority. The actual thread, in the foreground process, that performs the user I/O receives an additional boost. This priority increase is known as a **foreground boost**.

CONFIG.SYS

THREADS = 256

PRIORITY = DYNAMIC

MAXWAIT = 3

TIMESLICE = 32, 248

*4096
0x221*

1000000

*Threads = 64
til 200
dont
want
much
priority*

ps51254c

Figure 4-37. Configuration parameters

- **Threads**

This parameter controls the number of threads that can be created simultaneously in the system.

- **Priority**

Dynamic priority, is the scheduler's ability to adjust the priorities of threads in the regular class to ensure that all threads get a chance to run. It also ensures that the system provides as interactive a response to the user as possible. These adjustments are called **priority boosts**. This is another parameter used, but not 'seen' in the CONFIG.SYS file.

This value is not normally in the config.sys file. When changing its default values, then the statement must be placed in the config.sys file.

- **Maxwait**

This is the amount of time, in seconds, that a regular class thread, that is ready to run and has not yet done so, has to wait before the scheduler will boost it.

- **Timeslice**

This parameter controls the length of the timeslice that the scheduler uses. The first number is the represents the normal length of a threads timeslice in milli-

seconds. The second number is the maximum length of the timeslice. This is a parameter used, but not 'seen' in the CONFIG.SYS file.

When a thread receives either an I/O boost or a starvation boost, the threads priority is adjusted. The time slice is also adjusted to a minimum timeslice.

This value is not normally in the config.sys file. When changing its' default values, then the statement must be placed in the config.sys file.

Topic Summary

In this subtopic the student learned the differences between Threads and Processes. The student also was introduced to how the OS/2 schedules the threads for processor time.

This concludes this subtopic of the topic
"OS/2 2.x Internals".

Topic Summary

In this topic the student learned highlights of how OS/2 internally functions with regards to its memory and process management schemes.

- Memory management scheme.
- Input Output privilege level.
- Dynamic Linking.
- Process management.
- The scheduler.

This concludes the topic
"OS/2 2.x Internals".

TOPIC 5: File Systems

Topic objective:

Terminal objective:

After attending this topic the student should be able to distinguish between a FAT file system and an HPFS file system.

Enabling objectives:

Upon completion of this topic the student should be able to:

- Explain the key concept of sectors, tracks and, clusters
- Explain the key differences between the FAT , HPFS and the Enhanced FAT file systems
- Change their system config.sys file to support the HPFS file system
- Format a partition as HPFS

Prerequisite knowledge

No previous knowledge is required prior to attending this topic.

Topic objective:

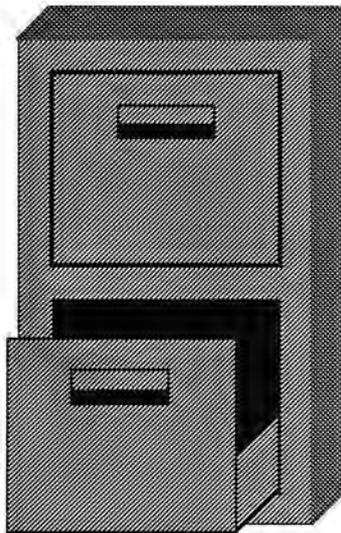
Terminal objective:

After attending this subtopic, the student should be able to distinguish what is meant by a FAT file system.

Enabling objectives:

Upon completion of this topic the student should be able to:

- Explain the key concept of sectors, tracks and, clusters
- Explain the key concept of the FAT file system



1. **FAT File System**
2. **High Performance File System (HPFS)**

ps512811

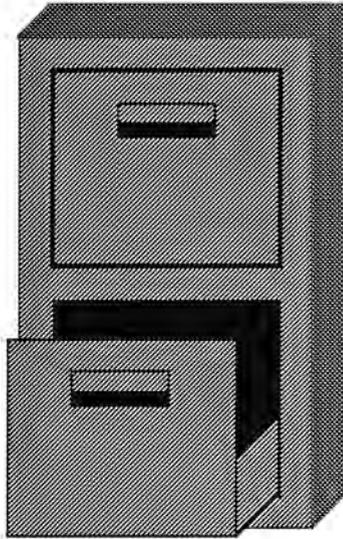
Figure 5-1. The 2 available file systems

OS/2 2.x provides two disk file systems: the File Allocation Table (FAT) system used by DOS 3.3 (and later) and previous releases of OS/2, and the High Performance File System (HPFS). Either file system can be specified during the OS/2 installation process. Both file systems can coexist on the same disk drive by partitioning the drive with more than one partition and formatting each partition separately for the file system desired.

By using the CHKDSK utility, the user can quickly see which file system is managing a storage device.

Formatting a partition, associates the File System Driver (FSD) with it. When an application makes a file system function call, OS/2 directs the request to the appropriate file system managing the device: an Installable File System or the FAT file system if no Installable File System is loaded or attached to the device.

FAT



Supports:

- DOS
- OS/2
- Up to 7 fixed disks
- Partitions C - Z

ps512812

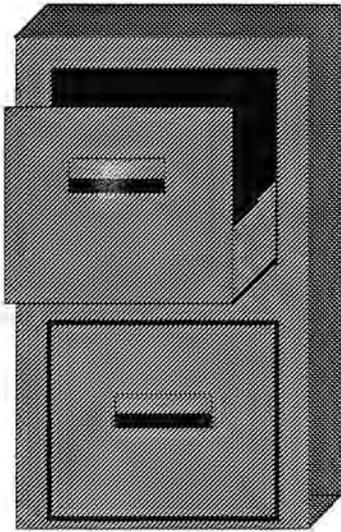
Figure 5-2. The FAT File System

The FAT file system revolves around the File Allocation Table. Each logical volume has its own FAT. This serves 2 purposes: it contains the allocation information for each file on the volume in the form of linked clusters and, it indicates which allocation units are available for a file that is being created or extended.

When FAT was designed, it was for diskette management which at that time was no more than 1 MB in size. On such disks, the FAT was small enough to be held in memory at all times, thus allowing very fast random access to any part of any file. When used on large drives, the FAT starts being too large to be held entirely in memory and has to be brought in, in pieces. With time, the available clusters get dispersed all across sectors thus causing fragmentation problems. The use of large clusters on fixed disks can also result in a lot of dead, unused, unavailable free space.

The fundamental data structures used by the FAT file system are not well suited for large random access devices.

HPFS



supported by OS/2

supports 2 GB files

supports partitions

**C through Z
each can be 512 GB**

ps512813

Figure 5-3. The HPFS File System

HPFS is an installable file system (IFS) designed to provide better performance than the existing FAT file system. HPFS was designed to provide fast access to very large disk volumes.

With the restrictions of the FAT file system, when the HPFS file system was designed, it was designed to address 3 key issues:

1. It is a way of organizing data on a random access block storage device
2. It is a software module that translates file oriented requests from an application program into more primitive requests that a device driver can understand
3. It is a practical illustration of an OS/2 feature known as an Installable File System

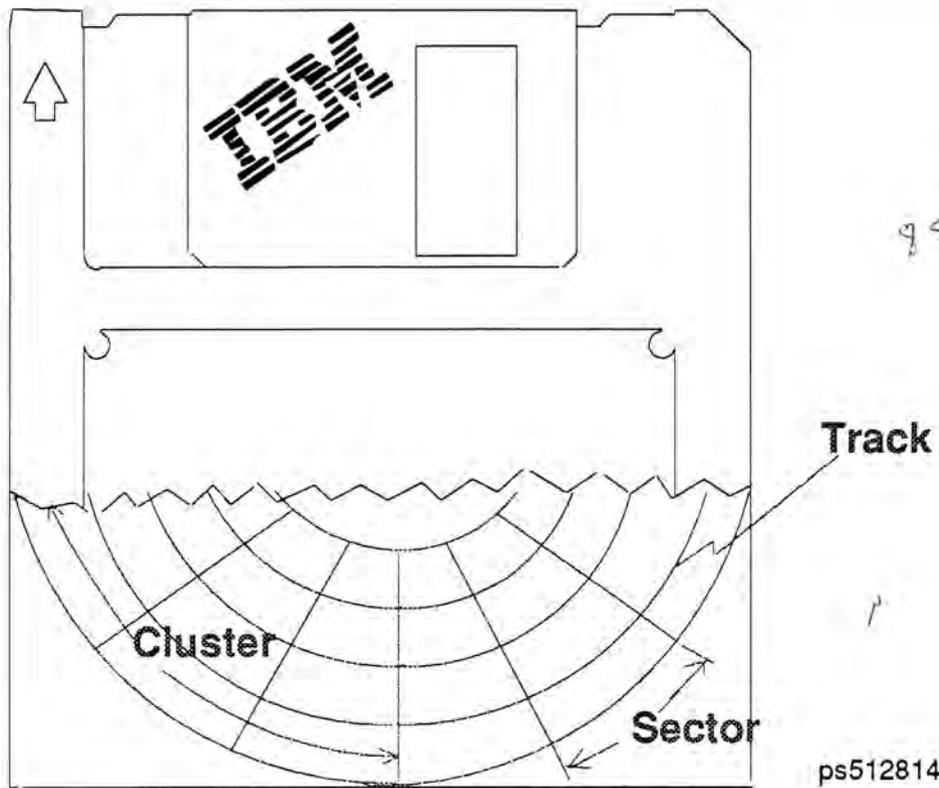


Figure 5-4. Structure of the disk

The structure of a disk is similar to that of a diskette. Visualize your new unused hard drive as initially blank. The Format command divides the disk/drive into concentric **tracks** and then further divides each track into units called **sectors**. Files are stored on your disk by recording the information on the sectors.

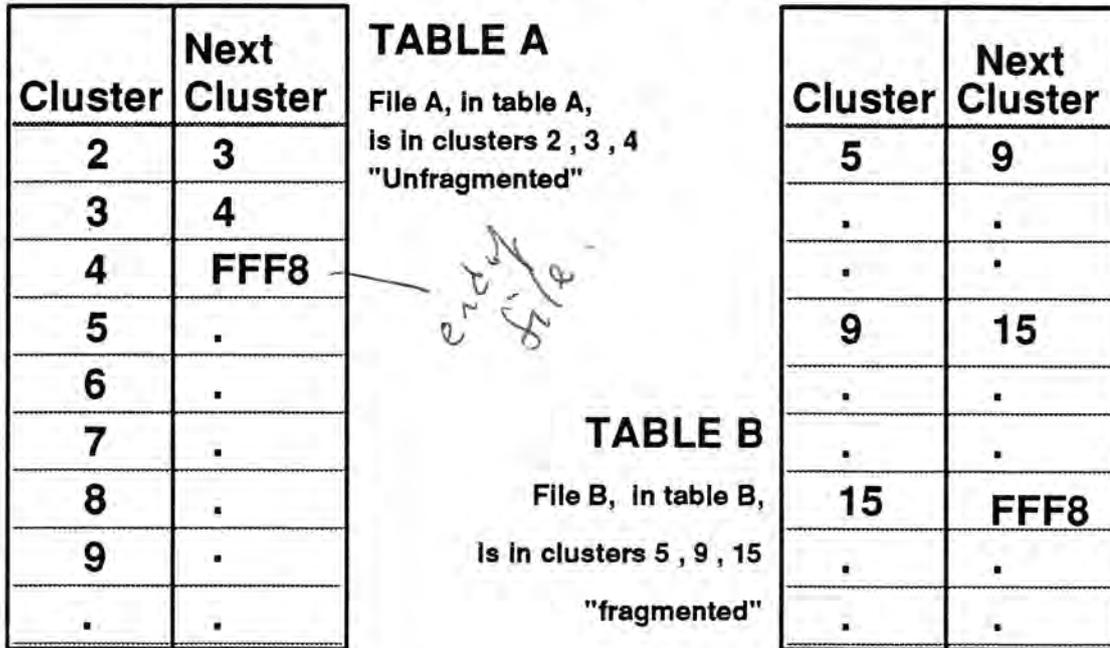
Files are stored on the drive in groups called **clusters** or **allocation units**. Depending on the file size, the file may fit into a single cluster or may require many clusters. Attempts are made to store a file's cluster in consecutive disk locations. However, if this is not possible, then the file will become fragmented. Each cluster will contain multiple disk sectors.

An **allocation unit** defines the smallest amount of disk space that can be allocated for a file.

To keep track of each file's cluster(s), a table is created on the drive called the **File Allocation Table**. This table contains every cluster available on the drive and indicates, for each cluster, whether:

- It is in use
- It is available
- It has been marked as unusable

File Allocation Table



ps512815

Figure 5-5. The File Allocation table

The File Allocation Table is used to locate each file's clusters on disk, storing the list of cluster numbers that make up each file. When a file is required, the system follows the file's cluster chain to determine where the file is situated. The starting cluster of a file is indicated by the file's directory entry.

A directory contains a list of file names and key information for each file in that directory. For every file you create on a FAT file system partition, the FAT file system creates a directory entry that contains 32 bytes of information. The starting cluster of a file is one of the items listed for each file. The FAT file system stores directories on the partition in one or more clusters, just as if they were files. Every time you need to access a file, the FAT file system must first locate the directory that file is situated in. To read that directory, it must be brought into memory. Once in memory, the it can then be determine if the required file is in that directory.

FS File System

Subtopic objectives:

Terminal objective:

After attending this subtopic the student should be able to distinguish between a FAT file system and an HPFS file system.

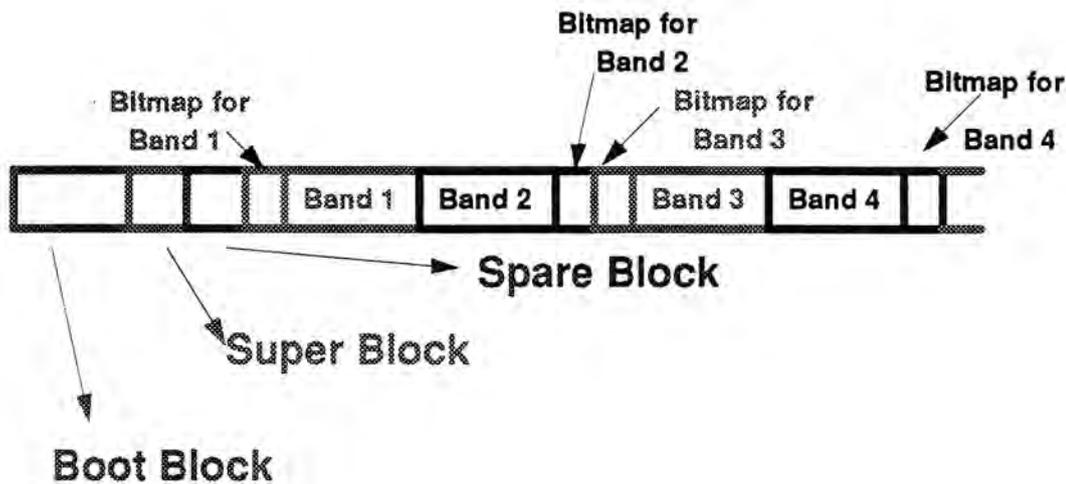
Enabling objectives:

Upon completion of this topic the student should be able to:

- Explain the key concepts of the HPFS file systems
- Change their system config.sys file to support the HPFS file system
- Format a partition as HPFS

Severe fragmentation
ON F.A.T.

HPFS Volume Structure



ps512821

Figure 5-6. HPFS Volume Structure

There are very few fixed structures in an HPFS Volume. There is a Boot Block, a Super Block, a Spare Block, a bit map for each band and, bands.

The Boot Block contains the volume name, a 32-bit volume ID and, a disk bootstrap program. The Super Block is only modifiable by the disk maintenance utilities. the Spare Block contains various flags and pointers. It is infrequently modified.

The remainder of the disk is divided into 8 MB bands. Each band has its own free space bitmap in which a bit represents each sector. The bitmaps are located between alternate bands thus the maximum contiguous space that can be allocated to a file is 16 MB. One band, located toward the seek center of the disk, is called the Directory Block Band.

Every file or directory on an HPFS volume is anchored on a system object called an **FNODE**. Each Fnode occupies a single sector and contains control and access history information used internally by the file system, extended attributes, the length and first 15 characters of the name of the associated file or directory and, an allocation structure. Each Fnode is always stored near the file or directory that it represents.

The allocation structure can take on different forms: dependant on the size and the contiguity of the file or directory.

How HPFS locates a file

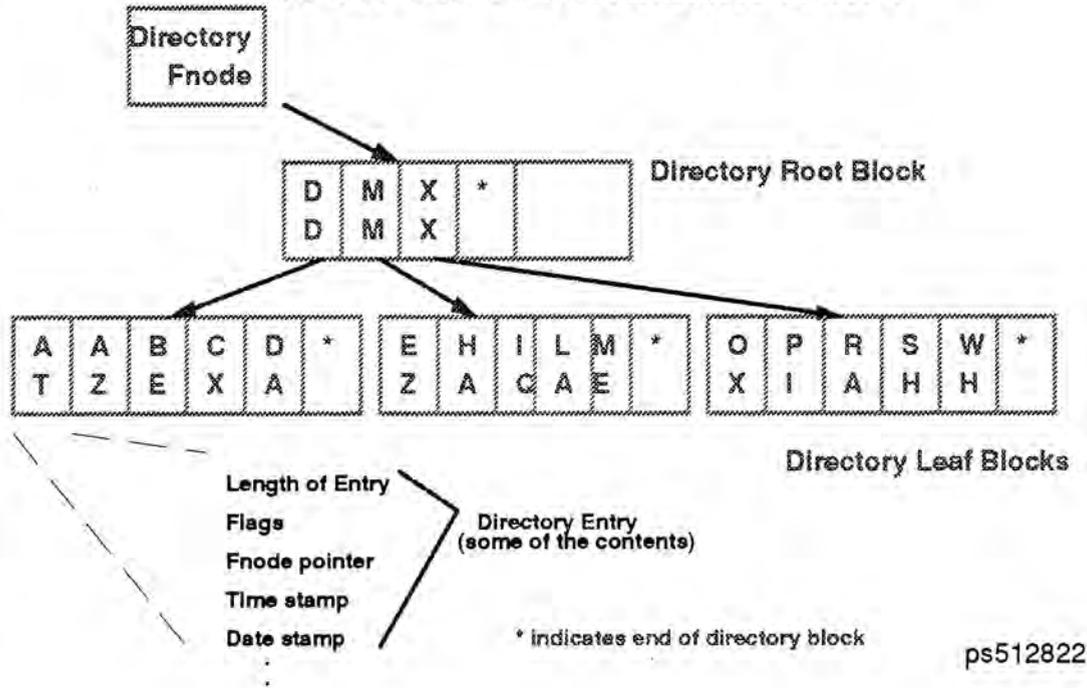


Figure 5-7. Directory Structure

Directories can grow to any size and are built up from 2 KB directory blocks which are allocated as four consecutive sectors on the disk. The file system will attempt to allocate directory blocks in the directory band, which is located near the seek center of the disk. Once that directory band is full, the directory blocks are allocated wherever space is available. Each 2 KB directory block contains from one to many directory entries. A directory entry will contain information about the entry.

The entries in a directory block are sorted by the binary lexical order of their name fields. The last entry in a directory block is a dummy record that indicates end of the block.

When searching for a specific name, the file system goes through the directory block until it either finds a match or finds a name that is lexically greater than the target. When found, the file system then extracts the B-Tree pointer from the entry.

HPFS Features

- **Caching**
- **Multi-threaded I/O**
- **Write-behind logic**
- **Optional write-through**
- **Strategic allocation of directory structures**

ps512823

Figure 5-8. Features of HPFS

- *Caching of Directories and Data*
It caches all directories; It has a "look aside" where the last 10 recently used directories are kept in cache.
- *Multi-Threaded I/O*
HPFS was written for a multi thread environment. Multiple threads are used for I/O , Caching , look ahead
- *Write-behind logic (also known as Lazy Write)*
This indicates that when writing data to the disk, if Lazy write is ON, this will actually write the data to cache instead of the disk. That data will be marked as 'dirty'. HPFS has a separate thread whose only job is to check the cache and when ever there is idle time in the system that thread will then do the actual writing to the disk from cache. This results in I/O overlapping of I/O.
- *Optional Write Through*
One can ensure that data will be written to the disk no matter what size it is.
- *Strategic Allocation of directory structures*
The root directory is now placed in the logical seek center of the disk.

HPFS Features

- **Highly contiguous file allocation**
- **Enhanced recoverability**
- **Extended attribute support**
- **Long file name support**
- **Bootability**

ps512824

Figure 5-9. Features of HPFS

- ***Highly Contiguous File Allocation***

HPFS defines bitmaps of contiguous strings of sectors and those are assigned to a file. HPFS attempts to keep contiguous disk space to allow for a file to grow contiguously. If there is no more room in that contiguous disk space, HPFS will attempt to find another band of contiguous disk space to allocate to that file and thus decrease the chances of fragmentation.

- ***Enhanced Recoverability***

If you were to get a power failure and were in the middle of updating your file, you can specify in your Config.sys file that upon powering up, you want the system to automatically run check disk against any HPFS partitions. You can also specify for the system to do a recovery as a result of the check disk.

- ***Hot Defect Mapping***

As data is written from an application to a fixed disk, if a sector error is found, HPFS will mark in the control block for that directory, that this particular sector is 'bad'. HPFS will not use that sector. That sector will always be marked as 'bad', this will prevent future attempts to write to that sector.

The same applies for the reading of data. If a read detect error is found, that sector will be marked as 'bad'. One can then attempt a recovery of that data. Application will also get an error that a bad sector has been detected in it's data.

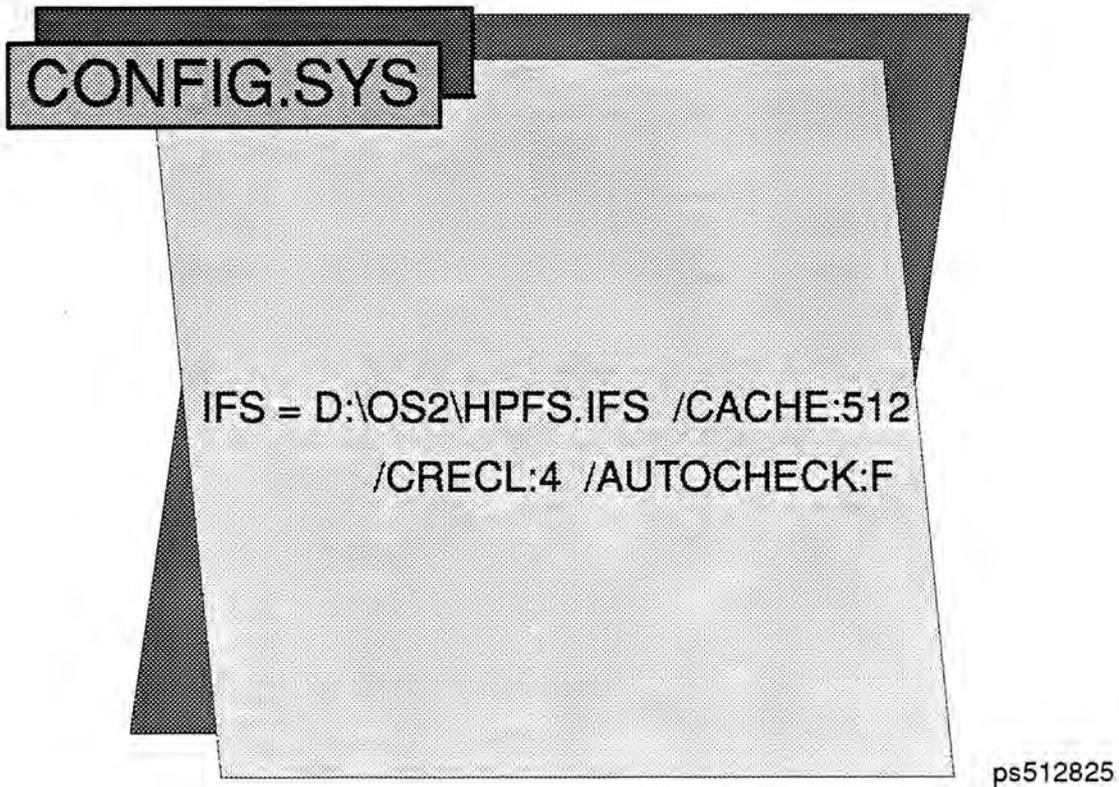


Figure 5-10. The IFS statement

The default cache size of 64 is in KB.

IFS stands for Installable File System.

CONFIG.SYS

```
RUN = D:\OS2\CACHE.EXE  
      /Lazy:on /Maxage:time  
      /Diskidle:time /Buffer:time
```

ps512826

Figure 5-11. The Run statement for HPFS

- **Maxage**
When the data in a cache block is more than the MAXAGE milliseconds out of date (with respect to the disk), the block is queued for writing to the disk. The default is 5000 milliseconds.
- **DiskIdle**
Sets the amount of time (in milliseconds) that a disk must be idle before it can accept data from cache memory.
- **BufIdle**
Sets the amount of time (in milliseconds) that the cache buffer can be idle before the data it contains must be written to a disk.

Formatting an HPFS partition

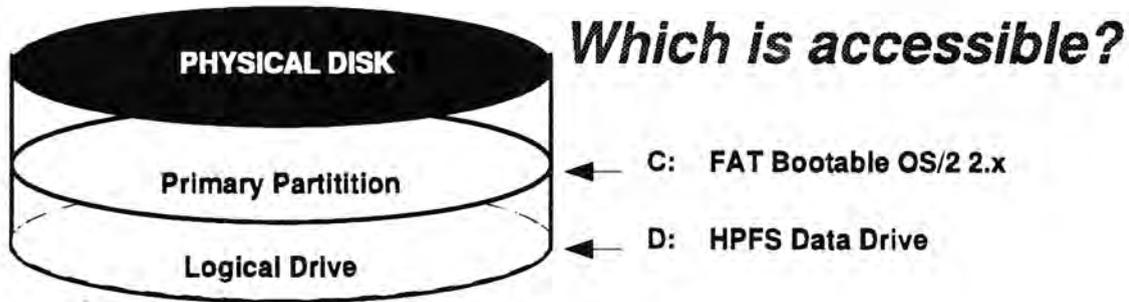
FORMAT F: /FS:HPFS

ps512827

Figure 5-12. Formatting for HPFS

FS represents, File System. If you require a partition to be formatted as FAT then the command would be;

- **FORMAT D: /FS:FAT**



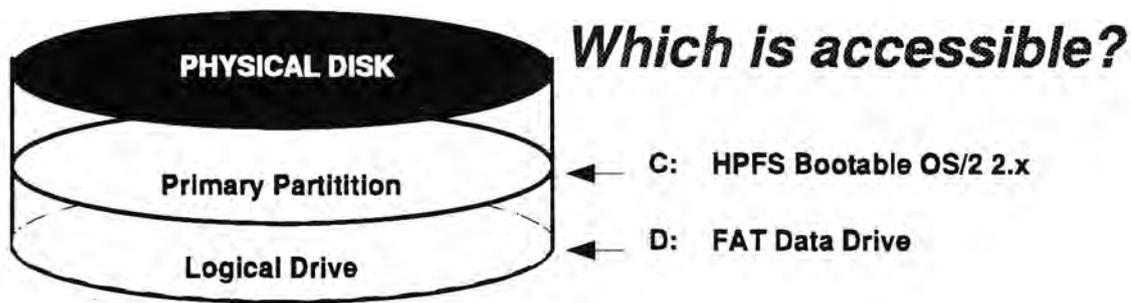
	FAT Partition C	HPFS Partition D
<i>accessible by OS/2 1.2 and above?</i>	yes	yes
<i>MVDM can get files?</i>	yes	yes
<i>Native DOS can get files?</i>	yes	no

ps512828

Figure 5-13. "C" as FAT and "D" as HPFS

*always put the HPFS drive
at the end.*

*FAT
HPFS
Boot mark*



	<i>HPFS Partition C</i>	<i>FAT Partition D</i>
<i>accessible by OS/2 1.2 and above?</i>	<i>yes</i>	<i>yes</i>
<i>MVDM can get files?</i>	<i>yes</i>	<i>yes</i>
<i>Native DOS can get files?</i>	<i>no</i>	<i>no</i>

ps512829

Figure 5-14. "C" as HPFS and "D" as FAT

At an OS/2 prompt, what would you see...?

[F:] dir

*The volume label in drive F is HPFS. The volume
serial number is 650E:121*

Directory of F:

<i>01-02-92</i>	<i>5:23p</i>	<i><DIR></i>	<i>0.</i>	<i>0.</i>
<i>01-02-92</i>	<i>5:43p</i>	<i><DIR></i>	<i>0.</i>	<i>0..</i>
<i>01-02-92</i>	<i>6:10p</i>	<i>127</i>	<i>0.</i>	<i>longfilename</i>
<i>01-02-92</i>	<i>8:13p</i>	<i>1590</i>	<i>0.</i>	<i>OS.extension</i>
<i>01-03-92</i>	<i>9:34a</i>	<i><DIR></i>	<i>0.</i>	<i>WORKSHOP</i>
<i>01-03-92</i>	<i>4:23p</i>	<i><DIR></i>	<i>0.</i>	<i>0.Xy[]Z=+.Se</i>
<i>01-03-92</i>	<i>5:30p</i>	<i>647</i>	<i>0.</i>	<i>ReadMe.Dat</i>

ps51282a

Figure 5-15. "DIR" from an OS/2 command prompt

At a DOS prompt, what would you see...?

[F:] dir

*The volume label in drive F is HPFS. The volume
serial number is 650E:121*

Directory of F:

<i>.</i>	<i><DIR></i>	<i>01-02-92</i>	<i>5:23p</i>	
<i>..</i>	<i><DIR></i>	<i>01-02-92</i>	<i>5:43p</i>	
<i>ReadMe</i>	<i>DAT</i>	<i>647</i>	<i>01-03-92</i>	<i>5:23p</i>
<i>WORKSHOP</i>	<i><DIR></i>	<i>01-03-92</i>	<i>5:23p</i>	

ps51282b

Figure 5-16. "DIR" from a DOS command prompt

Notice, that only the files and directories that meet the FAT naming convention is shown.

Enhanced (FAT) File System

Subtopic objectives:

Terminal objective:

After attending this subtopic the student should be able to explain the key concepts of the Enhanced FAT File System.

Enabling objectives:

Upon completion of this topic the student should be able to:

- Explain the key concepts of the Enhanced FAT File System

OS/2 FAT File System Features

- **Functionally equivalent to prior releases to FAT**
- **Disk layout is unchanged**
- **Improved diskcache**
- **Lazy-write**
- **Read-ahead**

ps512831

Figure 5-17. OS/2 2.x FAT File System

FAT caching in OS/2 2.x is the same as HPFS.

Superfat

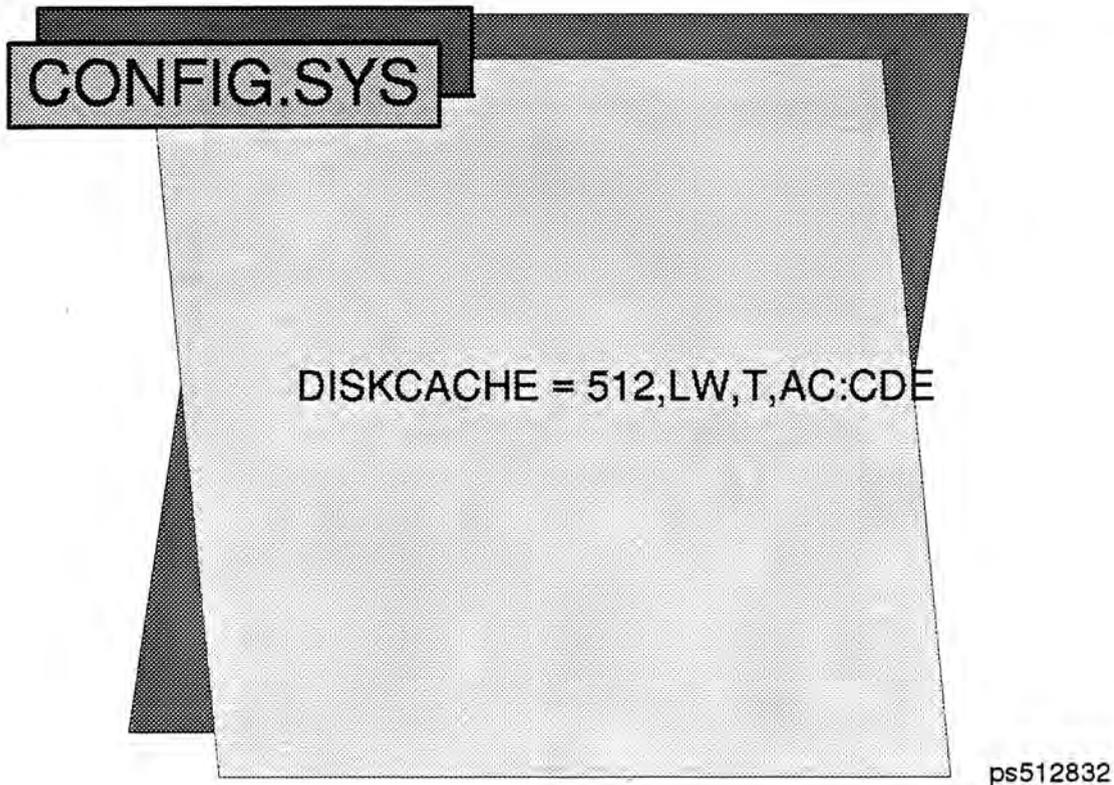


Figure 5-18. OS/2 2.x FAT DISKCACHE

Diskcache specifies the number of blocks of storage to allocate for control information and for use by the Disk Cache. Disk cache uses a portion of memory as additional hard disk buffer.

The **first** value specifies a number from 64 to 14400 which indicates the number of 1 KB blocks of storage to be used for control information and programs.

LW is the second value. It specifies if the contents in cache memory are written directly to the hard drive or only when the hard drive is idle.

T indicates the threshold size for the number of sectors that will be placed into cahce. the default is 4.

The **Autocheck** is the final option. It specifies which drives, upon startup, that the operating system will check to determine if it had been left in an inconsistent state at poweroff.

Topic Summary

In this topic the student learned

- The key concept of sectors, tracks and, clusters
- The key differences between the FAT , HPFS and the Enhanced FAT file systems
- How to format a partition for HPFS

This concludes the topic
"File Systems in OS/2 2.x."

TOPIC 6: The DOS Environment

Topic objective:

Terminal objective:

After attending this topic the student should be able to tune the OS/2 2.x DOS Environment and enable a Virtual DOS Machine Boot.

Enabling objectives:

Upon completion of this topic the student should be able to:

- Use DOS Settings to optimize a DOS application running under OS/2 2.1's DOS environment
- Enable a DOS Virtual Machine Boot

OS/2 pent product.

Clear & Simple Inc

P.O. Box 130

W. Simsbury, CT

06092

203-658-1204

ISM only -SR28-4641-00

Instructor: Jim Docherty.

VDM Architecture

Subtopic objectives

Terminal objective:

After attending this subtopic the student should know what components go into the creation, execution and termination of a VDM.

Enabling objectives:

After attending this subtopic the student should be able to describe the following features of the OS/2 MVDM kernel.

- VDM Address Space Management.
- Virtual DOS Machine Manager (VDMM).
- VDM Initialization.
- Virtual Device Drivers.

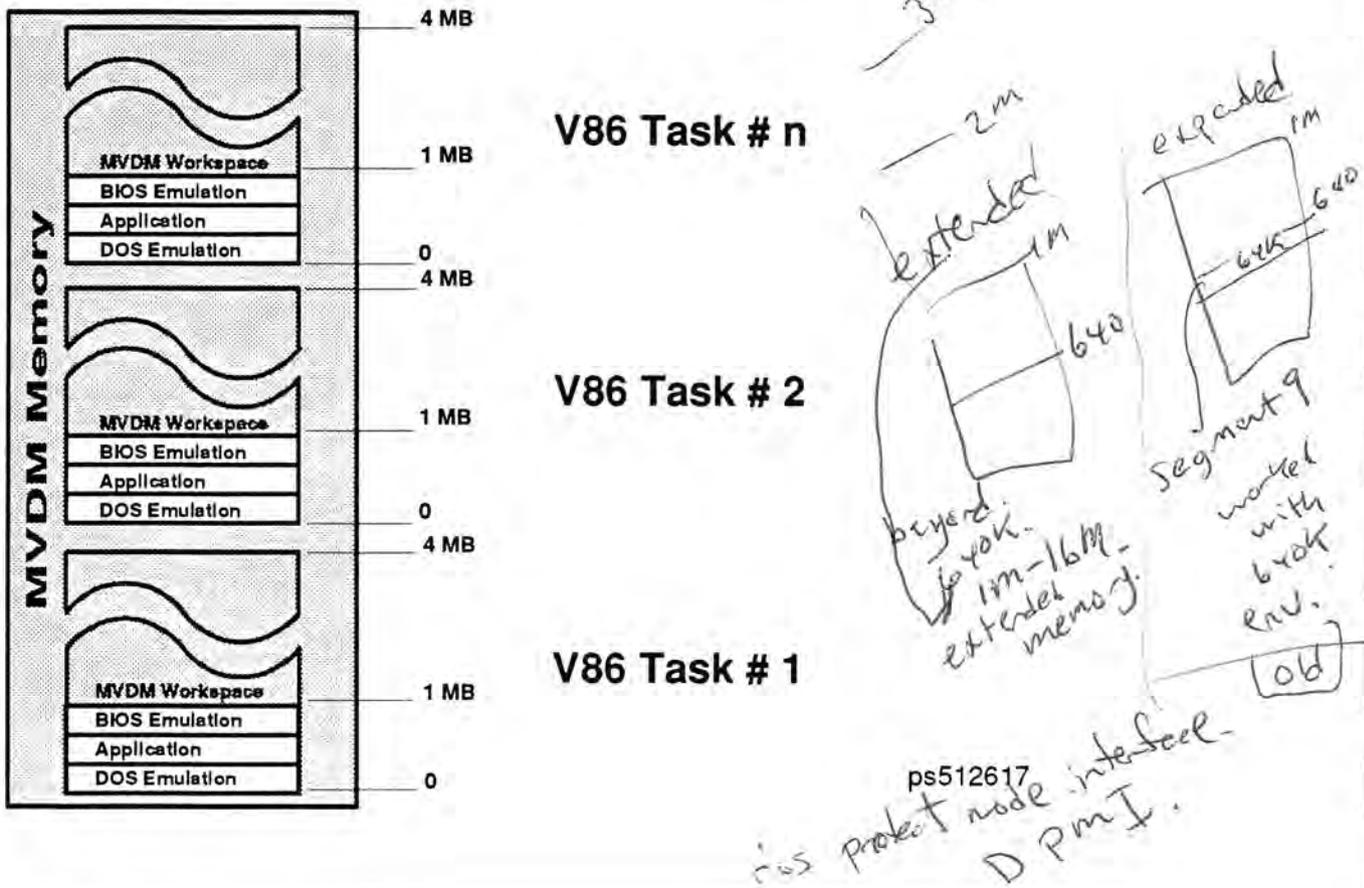


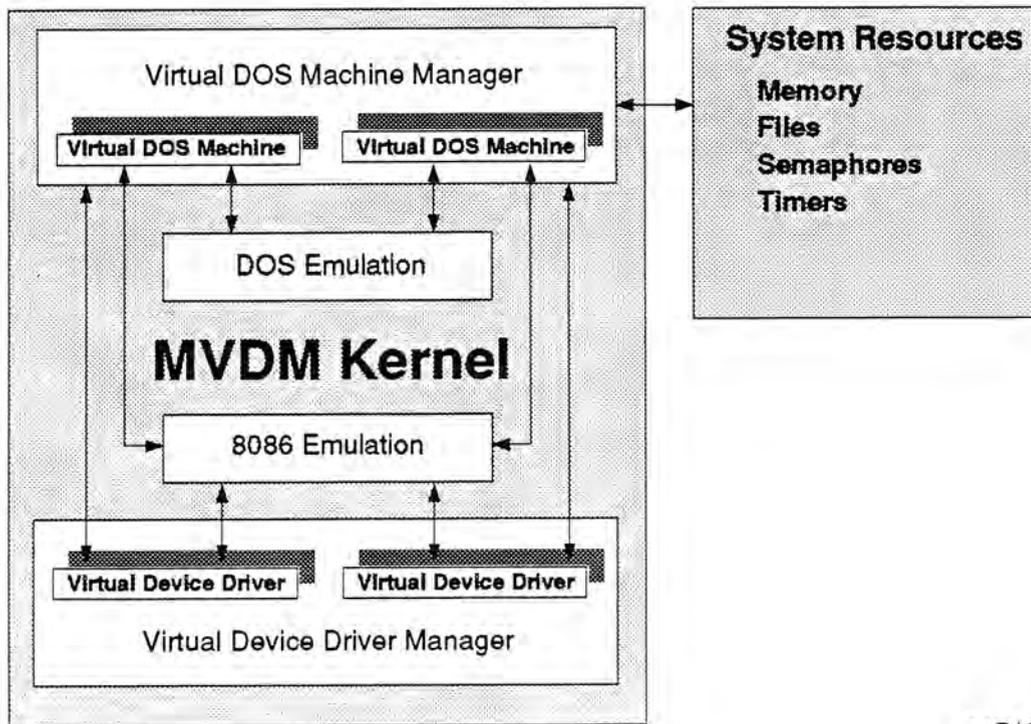
Figure 6-1. VDM Address Space Management

Each VDM task executes in the first megabyte of the linear address space so as to allow the physical addresses used within the DOS applications to be mapped directly to the process address space of the VDM.

DOS system areas such as ROM BIOS and the interrupt vector table are mapped from physical memory into the VDM's address space by the virtual device driver **VBIOS.SYS**.

DOS Emulation is implemented by running a very small portion of the DOS emulation kernel in V86 mode and a much larger portion of code in protected mode outside the VDM. In OS/2 V2.1, physical device drivers are loaded above 1 Mb and only the DOS Emulation kernel resides below 1 MB. Any user-installed OS/2 device drivers will not affect the amount of application space available to a DOS application running in a VDM.

In this way, MVDM architecture makes available to DOS applications the maximum amount of memory. In fact, up to 630KB is free for multiple DOS sessions. This represents an increase of about 100KB over memory available to the single DOS session that was available under OS/2 1.x.



ps512612

Figure 6-2. MVDM Kernel Components

The Multiple Virtual DOS Machine (MVDM) kernel controls the state and operation of concurrent VDMs and is composed of four major components.

- **Virtual DOS Machine Manager (VDMM)**
Initiates VDMs and communicates with DOS applications. Manages system resources for all active VDMs. Responsible for loading and initializing all virtual device drivers in conjunction with the Virtual Device Driver Manager.
- **Virtual Device Driver Manager (VDDM)**
Loads, initializes and communicates with virtual device drivers. The virtual device drivers are required to virtualize the hardware.
- **8086 Emulation**
Manages communication between 8086 instruction streams and virtual device drivers.
- **DOS Emulation**
Emulates the function and operation of the DOS operating system on a per-VDM basis. Each VDM emulates an entirely independent session of DOS.

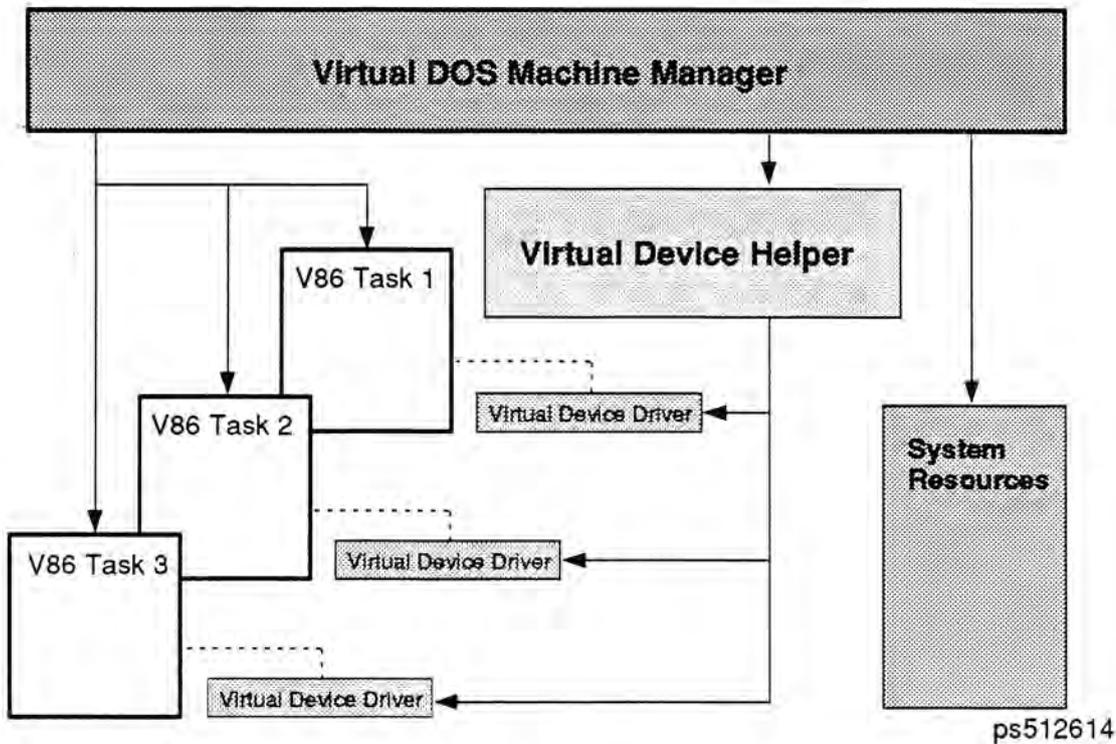
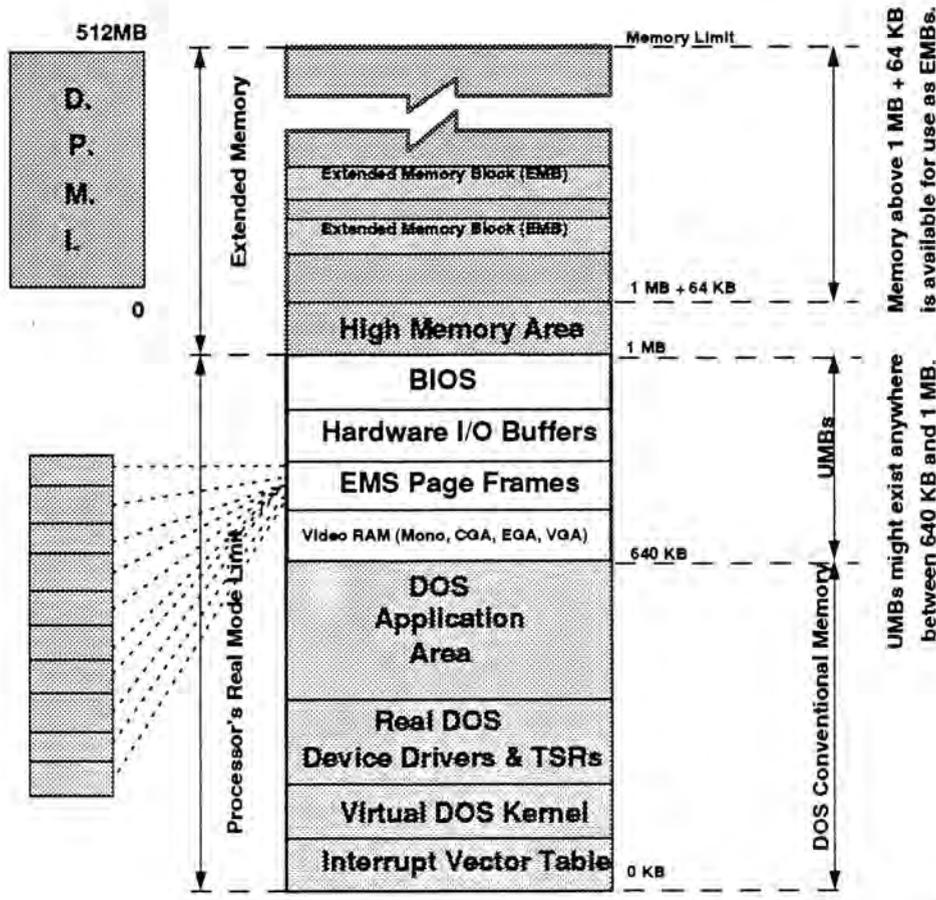


Figure 6-3. Virtual Device Drivers

Virtual device drivers are installable modules responsible for virtualizing the hardware and ROM BIOS aspects of the DOS environment for Virtual DOS Machines. A virtual device driver manages shared access to hardware I/O devices for multiple VDMs. This sharing allows an application, running in a VDM, to act as though it exercised sole control over I/O devices.

A virtual device driver typically performs I/O through a physical device driver using a direct call interface. However, a virtual device driver may directly access an I/O control device. This technique is used by the virtual video device driver, VVIDEO.SYS, for performance reasons. A virtual device driver may simulate hardware interrupts into one or many VDM processes.



ps512623

Figure 6-4. Memory : Conventional, Expanded, Extended

The diagram above shows some of the types of memory commonly referred to :

Conventional Memory

- refers to that area of memory that lies between 0 and 640KB.

Extended Memory(XMS)

- refers to any memory above the 1MB line which is addressed by the processor in protected mode.

High Memory Area(HMA)

- refers to that area of memory between the 1MB and (1MB + 64KB) line.

Expanded Memory Blocks (EMBs)

- that area of memory above the HMA not accessible from real mode.

UMA,UMB

- refers to that area of memory between 640KB and 1MB.

Expanded Memory(EMS)

- a page mapping technique that provides additional memory support by allowing DOS applications to allocate and access up to 32MB of additional memory.

Topic Summary

In this subtopic the student learned about the virtual DOS machine architecture :

- the VDM Address Space Management.
- the Virtual DOS Machine Manager (VDMM).
- the VDM Initialization.
- the Virtual Device Drivers.

This concludes the first subtopic of the topic
"The DOS Environment."

Settings

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to tune a virtual DOS machine session.

Enabling objectives:

After attending this subtopic the student should be able to:

- Adjust CONFIG.SYS for maximum VDM memory
- Tune DOS auto execute file for maximum VDM memory
- Know how to adjust the DOS Settings for an individual VDM

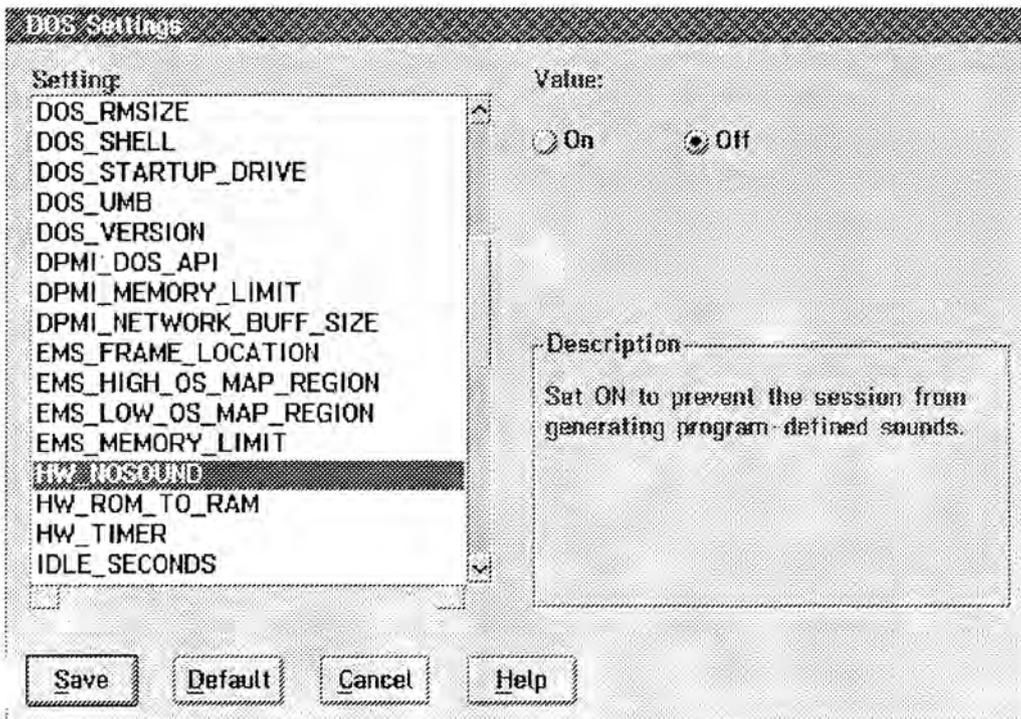


Figure 6-5. DOS Settings

The **DOS settings** are the means by which MVDM provides the ability to customize and control special properties that affect the behaviour of DOS applications running in a VDM. DOS settings are managed on a per-VDM basis. DOS settings are used during creation and initialization of a VDM, and certain settings may also be altered dynamically during VDM execution.

The DOS settings improves the DOS compatibility of a VDM because it allows a user to configure the VDM for DOS applications which might otherwise not work well with the default settings.

on by default. - turn off

video retrace emulation - full screen apps - turn off

© Copyright IBM Corporation 1994

Tuning CONFIG.SYS for maximum VDM Memory

- Device = DOS Statements**
- use DOS settings instead
- Buffers = command**
- Minimize
- LASTDRIVE = command**
- Use lowest letters possible
- FCBS = command**
- Set to 1

P1071602

Figure 6-6. Tuning CONFIG.SYS for maximum VDM memory.

To insure the maximum amount of memory is available to DOS applications in every VDM, it is recommended that the four areas mentioned above be tuned as follows:

- Application specific DOS device drivers should be loaded via the **DOS Device Drivers** option of DOS Settings. **DEVICE =** statements for DOS device drivers should be eliminated from CONFIG.SYS unless the device driver is required for every VDM.
- The number of buffers specified in the **Buffers** command in CONFIG.SYS should be minimized. Each buffer consumes about 500 bytes. Do not reduce this number too much because some programs might not run properly if there are too few buffers. The default number of buffers is 30. The number should not be reduced to fewer than 10 or 15 buffers.
- If CONFIG.SYS includes the **LASTDRIVE** command, this should be set to a letter such as J or K, rather than Z. Each additional drive uses about 100 bytes.
- If the CONFIG.SYS file contains an **FCBS** command, set FCBS to 1.

AUTOEXEC.BAT

```
PATH D:\OS2;D:\OS2\MDOS;D:\;  
LOADHIGH APPEND D:\OS2;D:\OS2\SYSTEM  
PROMPT $!$P$G
```

P1071604

Figure 6-7. AUTOEXEC.BAT

The AUTOEXEC.BAT file is specific to the VDM environment and has no effect on the OS/2 operating system. The AUTOEXEC.BAT file starts memory-resident programs, like network programs, and sets up environment variables. In addition, the AUTOEXEC.BAT file may also define the command prompt.

The default AUTOEXEC.BAT file for all VDMs in OS/2 V2.1 is shown above.

In the above AUTOEXEC.BAT file, the **LOADHIGH** command loads the **APPEND** terminate and stay resident (TSR) program into the High Memory Area. This makes available more memory to all DOS applications running in every VDM. The function performed by the **SET COMSPEC** command that you might expect to find in the AUTOEXEC.BAT has been moved to the CONFIG.SYS file.

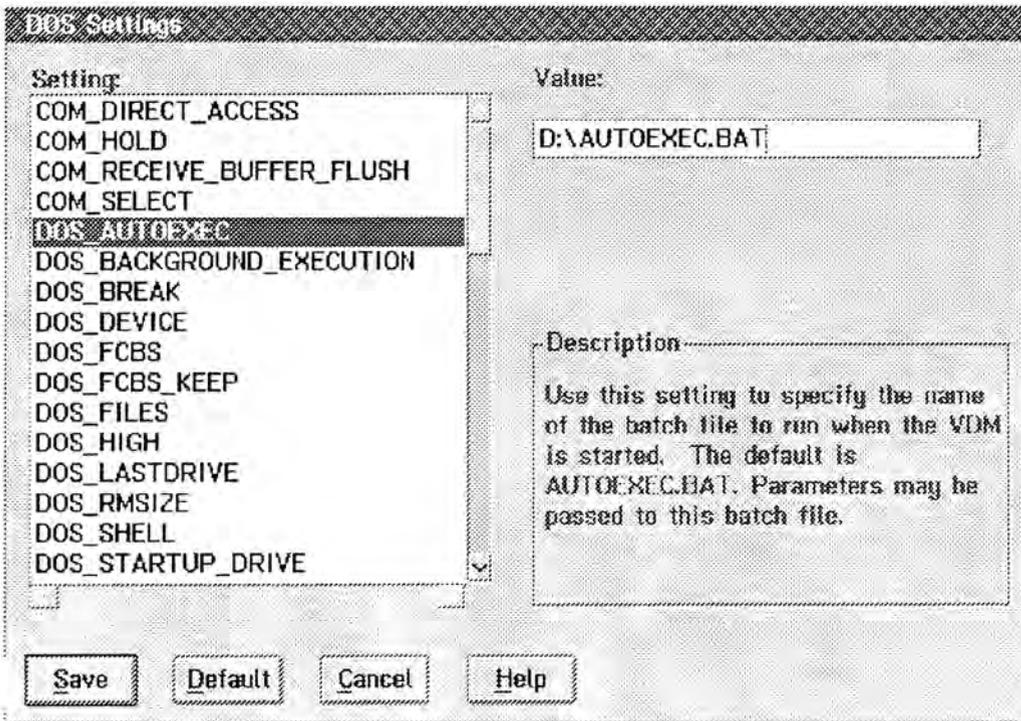
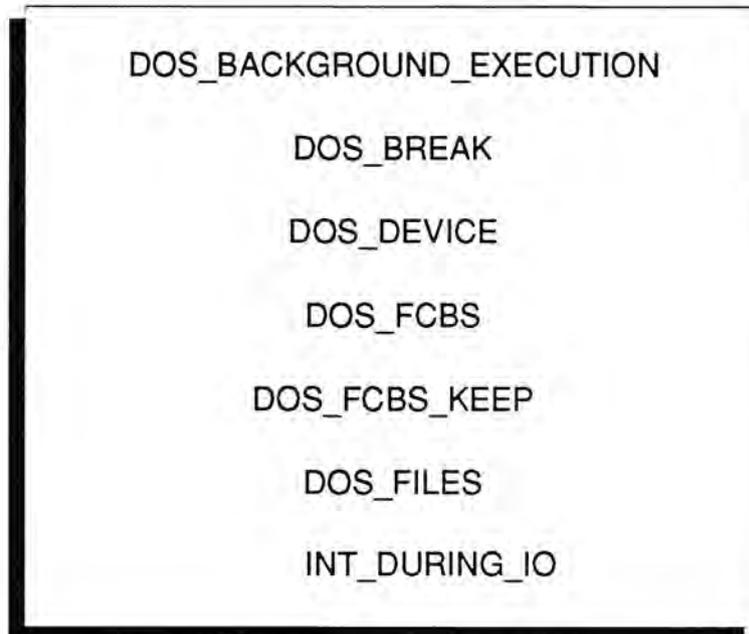


Figure 6-8. DOS_AUTOEXEC Setting

A new DOS setting has been added in OS/2 2.1. The DOS_AUTOEXEC setting enables a specific DOS command (.BAT) file to be executed when a VDM is created.

The BAT file can then be used to tailor the DOS environment by executing programs other than device drivers that need to be run in the VDM environment.

DOS Environment Settings I

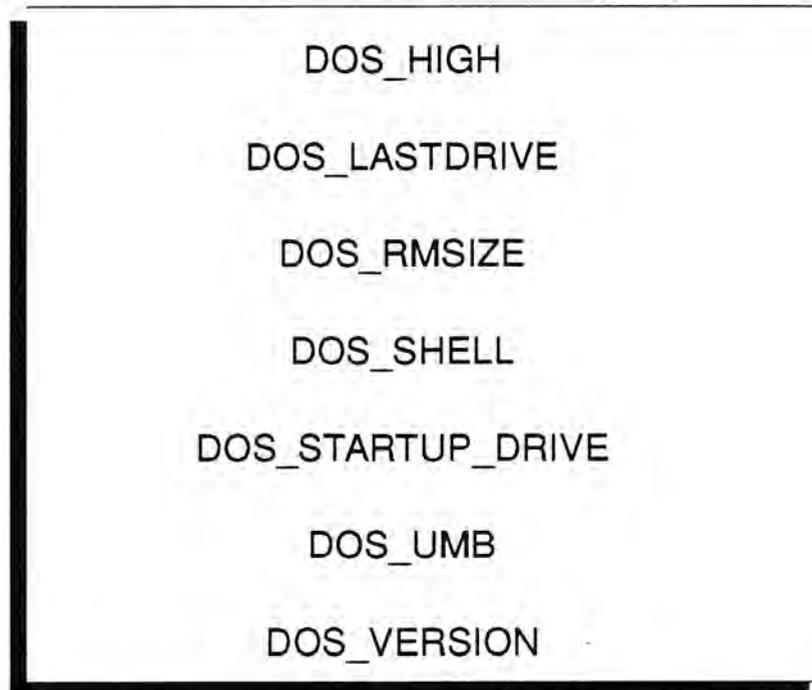


ps515108

Figure 6-9. DOS Environment Settings I

- **DOS_BACKGROUND_EXECUTION** allows to suspend the execution of the program when it is in the background.
- **DOS_BREAK** enables or disables CTRL-Break for the specified VDM.
- **DOS_DEVICE** is used to add or modify information about DOS device drivers for the specified VDM.
- **DOS_FCBS** specifies the maximum number of file control blocks (FCBs) which may be open at the same time in the VDM.
- **DOS_FCBS_KEEP** specifies the number of FCBs that will be protected against automatic closure.
- **DOS_FILES** specifies the maximum number of file handles which may be opened in a VDM.

DOS Environment Settings II

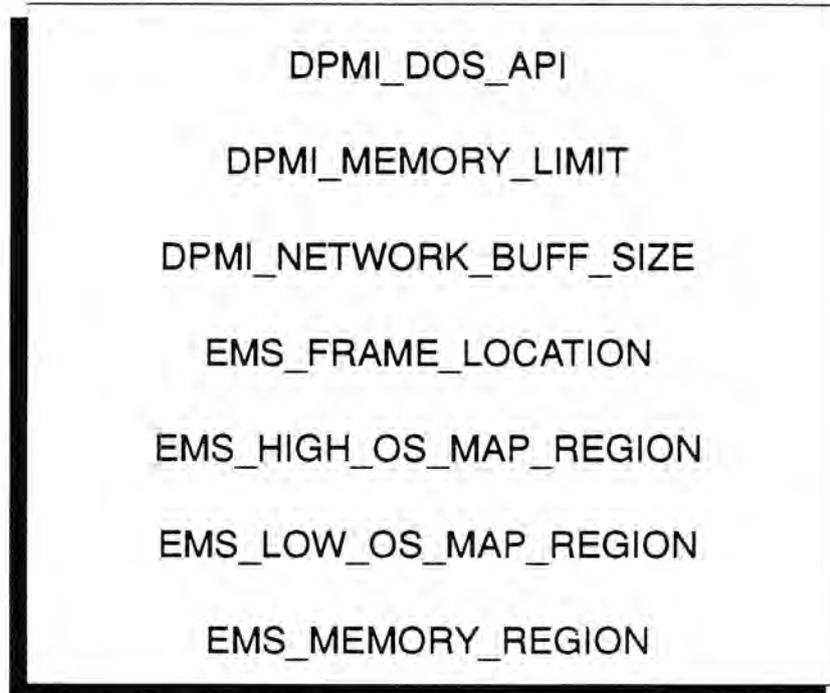


ps515110

Figure 6-10. DOS Environment Settings 2

- **DOS_HIGH** determines whether DOS is loaded outside the 640KB low memory address space.
- **DOS_LASTDRIVE** specifies the highest available logical drive letter for the specified VDM.
- **DOS_RMSIZE** specifies the amount of memory available to DOS applications.
- **DOS_SHELL** used to specify the DOS command processor or to add parameters for the given processor.
- **DOS_STARTUP_DRIVE** specifies the location of the DOS kernel to be loaded into the VDM.
- **DOS_UMB** specifies whether DOS owns Upper Memory Blocks (UMBs) and manages the loading of device drivers and TSR programs.
- **DOS_VERSION** allows the operating system to report a "fake" DOS version number in response to a request from a program in the VDM.

DPMI and EMS Settings

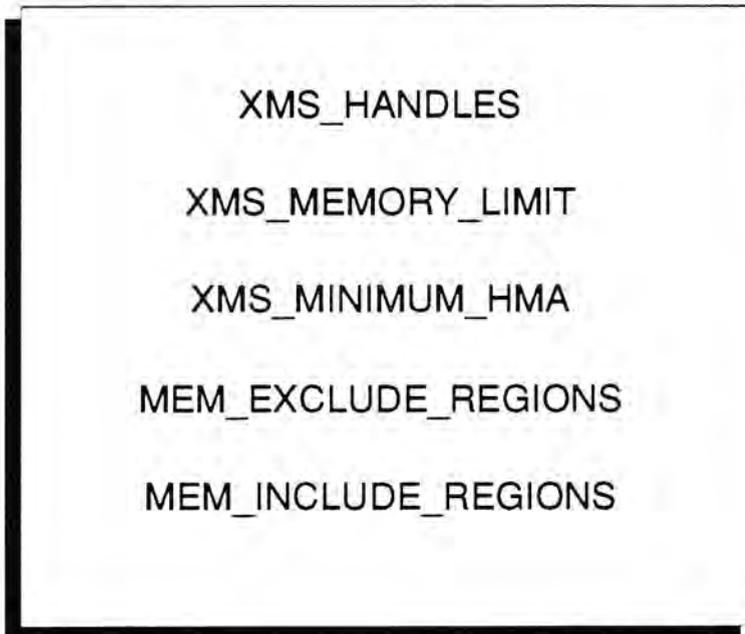


ps515112

Figure 6-11. DPMI and EMS Settings

- **DPMI_DOS_API** determines whether DOS API translation is enabled for the specified VDM.
- **DPMI_MEMORY_LIMIT** specifies the maximum amount of protected mode memory available to DPMI applications running in the VDM.
- **DPMI_NETWORK_BUFF_SIZE** specifies the size of the network translation buffer for DPMI programs in this session.
- **EMS_FRAME_LOCATION** allows to change the location of the LIM EMS region.
- **EMS_HIGH_OS_MAP_REGION** is used to adjust the size of the additional EMS region for programs using additional addresses to access expanded memory.
- **EMS_LOW_OS_MAP_REGION** allows to set the size of the remappable conventional memory available in a VDM.
- **EMS_MEMORY_LIMIT** is used to control the amount of EMS memory available to a VDM.

XMS and Memory Extender Settings

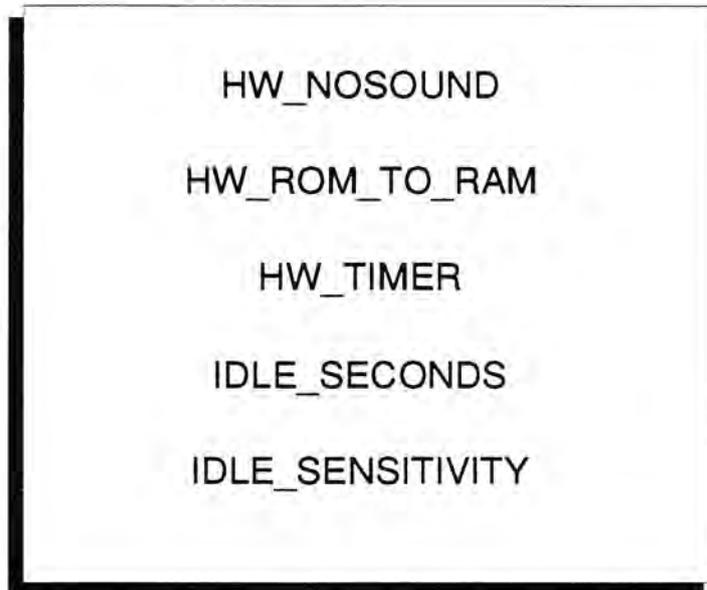


ps515114

Figure 6-12. XMS and Memory Extender Settings

- **XMS_HANDLES** specifies the number of XMS memory block handles.
- **XMS_MEMORY_LIMIT** specifies the per VDM XMS memory limit.
- **XMS_MINIMUM_HMA** specifies the minimum HMA memory request allowed.
- **MEM_EXCLUDE_REGIONS** is used to specify address ranges which should be protected from use by EMS/XMS and direct access by applications.
- **MEM_INCLUDE_REGIONS** specifies regions which should be made available to EMS/XMS.

Hardware Environment and Idle Detection Settings

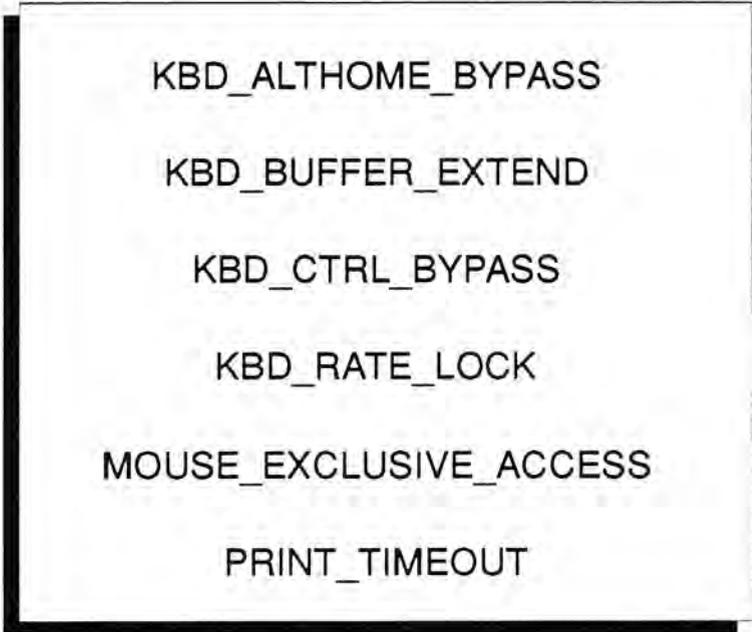


ps515116

Figure 6-13. Hardware Environment and Idle Detection Settings

- **HW_NOSOUND** enables or disables sound started by a DOS program.
- **HW_ROM_TO_RAM** enables the operating system to copy ROM and run the copy in 32-bit RAM.
- **HW_TIMER** allows an application to have direct access to the 8253 timer ports.
- **IDLE_SECONDS** is used to select a period of allowable idle time before the operating system reduces the idle program's portion of processor time.
- **IDLE_SENSITIVITY** sets a threshold for polling time before the operating environment reduces the polling program's portion of processor time.

Keyboard, Mouse and Printer Settings



KBD_ALTHOME_BYPASS
KBD_BUFFER_EXTEND
KBD_CTRL_BYPASS
KBD_RATE_LOCK
MOUSE_EXCLUSIVE_ACCESS
PRINT_TIMEOUT

ps515118

Figure 6-14. Keyboard, Mouse and Printer Settings

- **KBD_ALTHOME_BYPASS** prevents the ALT-Home key sequence from switching the VDM between full screen and windowed mode.
- **KBD_BUFFER_EXTEND** increases a VDM's keyboard type-ahead buffer size.
- **KBD_CTRL_BYPASS** inhibits one or more control key sequences, allowing an application in the VDM to use these sequences for its own purposes.
- **KBD_RATE_LOCK** prevents a DOS application from changing the system keyboard repeat rate.
- **MOUSE_EXCLUSIVE_ACCESS** gives a DOS program exclusive ownership of the mouse, if you have two mouse pointers.
- **PRINT_TIMEOUT** is used to adjust the amount of time, that the operating system waits before forcing a print job to the printer.

Video and Communication Settings

```
VIDEO_FASTPASTE
VIDEO_MODE_RESTRICTION
VIDEO_ONDEMAND_MEMORY
VIDEO_RETRACE_EMULATION
VIDEO_ROM_EMULATION
VIDEO_SWITCH_NOTIFICATION
VIDEO_WINDOW_REFRESH
VIDEO_8514_XGA_IOTRAP
COM_HOLD
```

ps515120

Figure 6-15. Video and Communication Settings

- **VIDEO_FASTPASTE** is used to speed up input from sources other than the keyboard, for example from the clipboard.
- **VIDEO_MODE_RESTRICTION** allows to extend DOS conventions memory by limiting the video mode support to text or CGA graphics.
- **VIDEO_ONDEMAND_MEMORY** delays the allocation of a video-save buffer, which can free memory swap space for use by a full-screen session.
- **VIDEO_RETRACE_EMULATION** disables simulated video retrace.
- **VIDEO_ROM_EMULATION** disables emulation of commonly used video ROM functions.
- **VIDEO_SWITCH_NOTIFICATION** notifies the DOS program when the session switches to or from full-screen.
- **VIDEO_WINDOW_REFRESH** is used to adjust the window update frequency for a specific DOS session.
- **VIDEO_8514_XGA_IOTRAP** allows unrestricted access to 8514/A display adapter hardware.
- **COM_HOLD** is used to keep open a communications resource, e.g. COM1, until the DOS session ends.

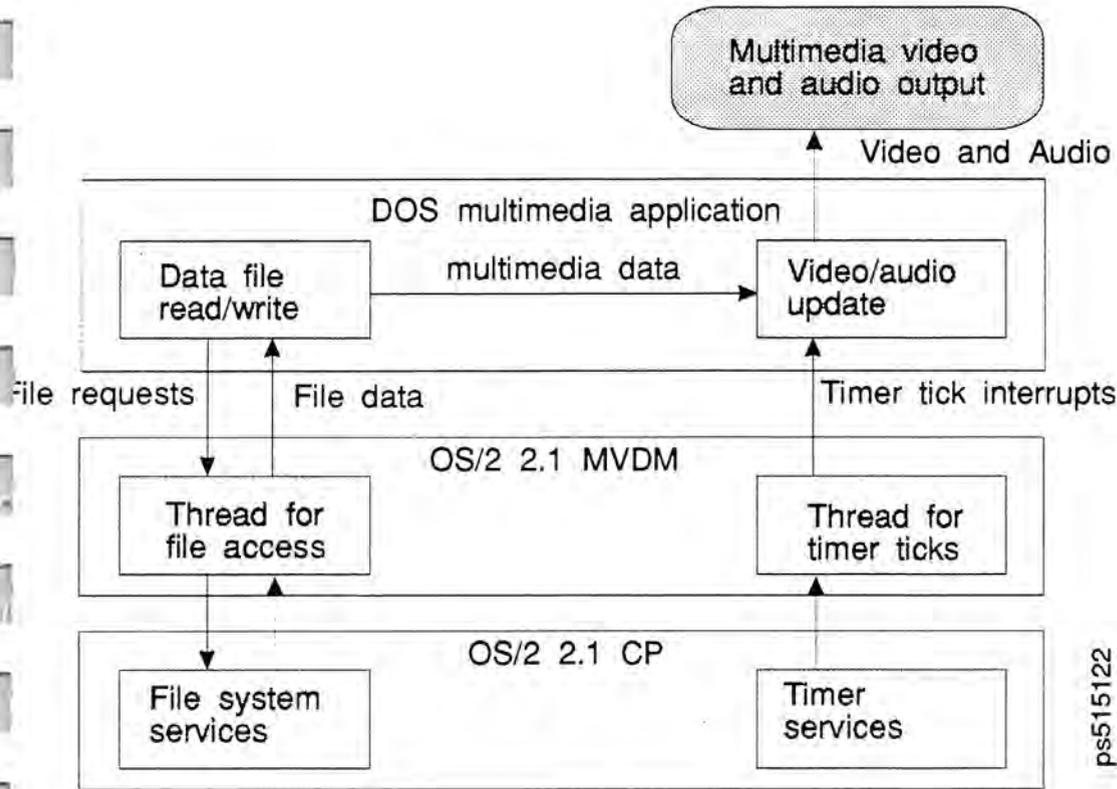


Figure 6-16. Dual Thread Support

Some DOS applications, such as Multimedia, are especially demanding of the DOS environment in that they need to ensure smooth display of pictures and sound reproduction and at the same time read large amounts of data from disk or CD.

DOS Multimedia applications are faced with the challenge of producing video and audio output fast enough, while reading large amounts of data from disk, so that it appears smooth to the user.

In the virtual DOS environment of OS/2 2.0, there was the drawback that the single thread could be blocked waiting for a file system request to complete.

The solution introduced in OS/2 2.1 is the ability to run a second thread in the VDM.

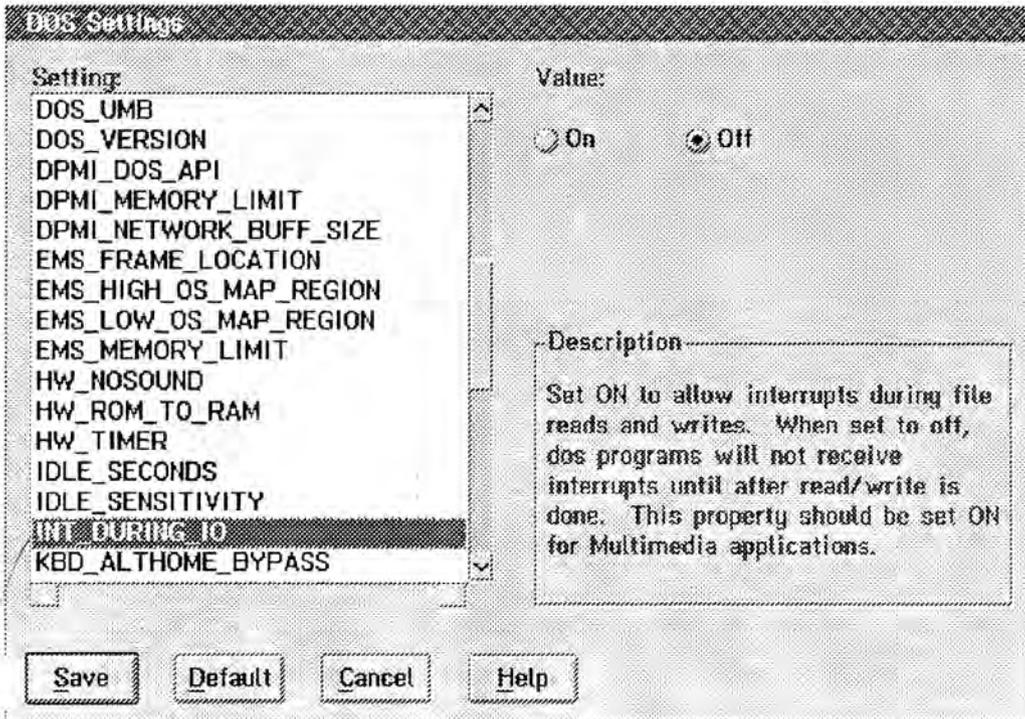


Figure 6-17. Dual Thread VDM

OS/2 2.1 enhances MVDM support by enabling the use of a second thread to be started for file system data access.

multi-media only.

*improves real time video.
Put ON
No cdrom. is ok*

Topic Summary

In this subtopic the student learned to tune a virtual DOS machine session.

The student was taught how to:

- Adjust CONFIG.SYS for maximum VDM memory
- Adjust AUTOEXEC.BAT for maximum VDM memory
- Adjust the DOS Settings for an individual VDM

This concludes the second subtopic of the topic
"The DOS Environment."

Virtual Machine Boot

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to configure a virtual DOS machine boot session.

Enabling objectives:

After attending this subtopic the student should be able to:

- Explain the requirement for VDM Boot.
- List 3 ways that the DOS boot record can be packaged for VDM booting.
- Alter the DOS CONFIG.SYS and AUTOEXEC.BAT files.
- Utilize the VMDISK utility to create an image file.
- Create an object in a folder on the OS/2 desktop which causes DOS to be booted in a VDM when selected.

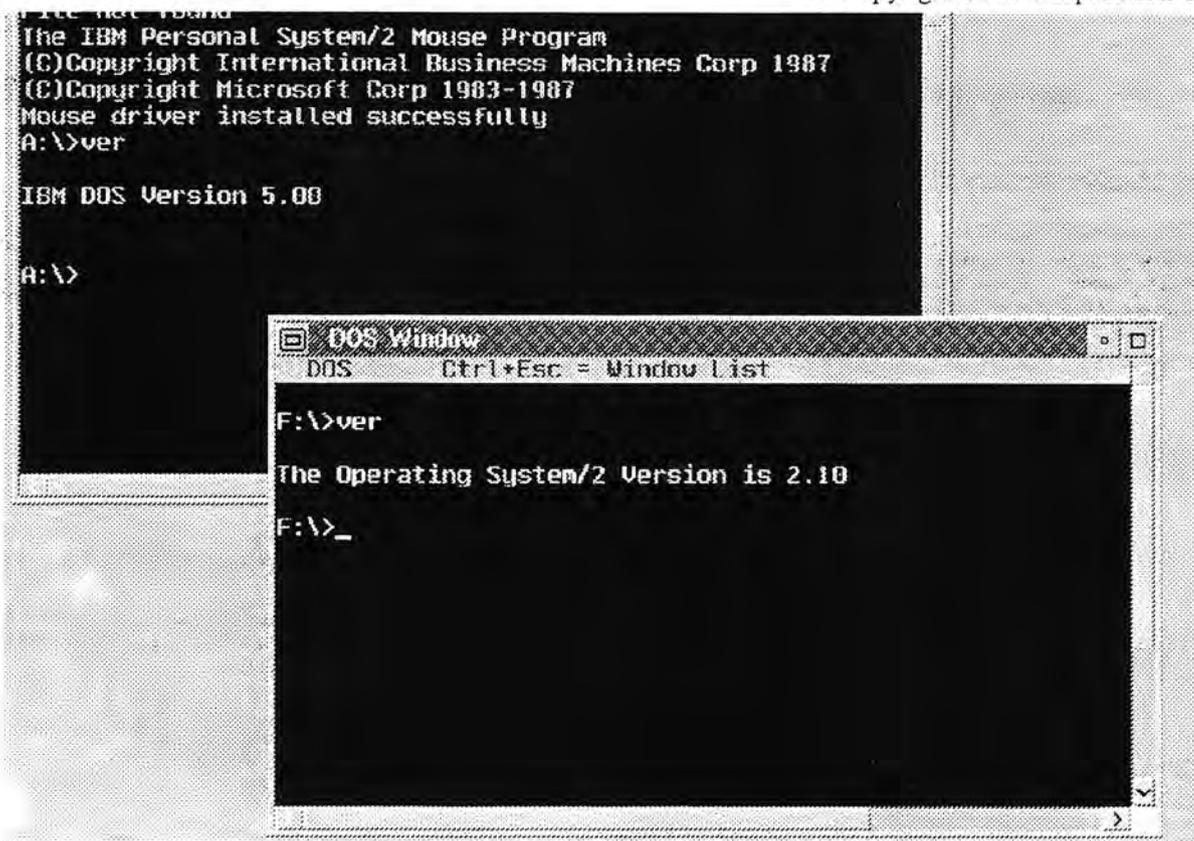


Figure 6-18. 8086 Boot Requirement

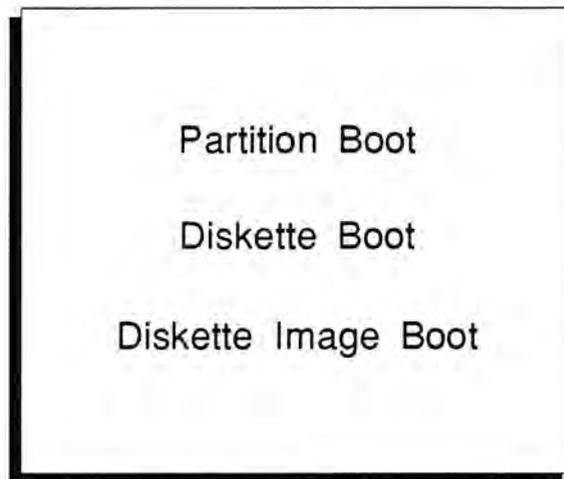
An important goal of OS/2 Version 2.1 is the ability to run past, current, and future DOS programs. Most DOS programs can run unchanged in OS/2's MVDM environment. The MVDM DOS, however, is highly optimized for the underlying OS/2 Version 2.1. Therefore, some internal differences may arise between emulated DOS and real DOS.

The ability to run existing DOS programs is a key end user need. Running the actual DOS 3.x, and above, kernels will provide the maximum achievable compatibility in a multitasking environment.

The **Virtual Machine Boot** feature provides the ability to boot an "off the shelf" 8086 kernel into an OS/2 virtual DOS machine. Each kernel will run in its own V86 session and its access to hardware is controlled by the OS/2 kernel and installed virtual device drivers.

The primary objective is support for DOS applications that have some DOS version sensitivities. Therefore, the VM Boot is formally supported for **IBM** versions of DOS 3.0 and above.

Packaging of the DOS Boot Record

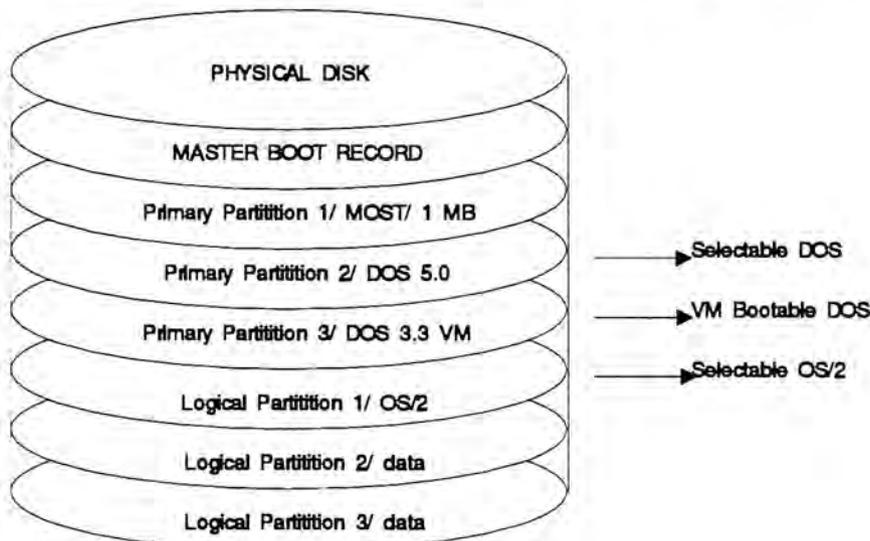


ps515202

Figure 6-19. Packaging of the DOS Boot Record

Partition Boot	The hard drive is partitioned into C: and D: partitions. The DOS version is installed on C: and OS/2 is installed on D:. With OS/2 running, the VM boot feature can then boot the DOS version that is installed on the C: partition.
Diskette Boot	The 8086 operating system is installed on a diskette. The VM boot feature can then boot that operating system when the diskette is installed in a diskette drive.
Diskette Image Boot.	The image of a bootable diskette is saved on the hardfile and the VM boot feature then boots from that image.

Hard Disk VM Boot Partition



ps515204

Figure 6-20. Hard Disk VM Boot Partition

Shown above is an example of a hard disk which has been partitioned with two selectable operating systems and one partition for booting into a virtual machine.

When installing OS/2, the customer will be asked if the installation procedure should install to the C: partition. The user should choose to install to an alternate partition. This action will bring up the **FDISK** utility. At this point, a logical partition should be created for installing OS/2. A primary partition must be created for installing DOS.

After DOS has been installed in the partition meant for VM booting, the CONFIG.SYS and AUTOEXEC.BAT files will need to be altered so as to address some necessary virtual device drivers.

If the user does not have the freedom to partition his hardfile but wants to boot DOS, the customer can set up a VM Boot from diskette.

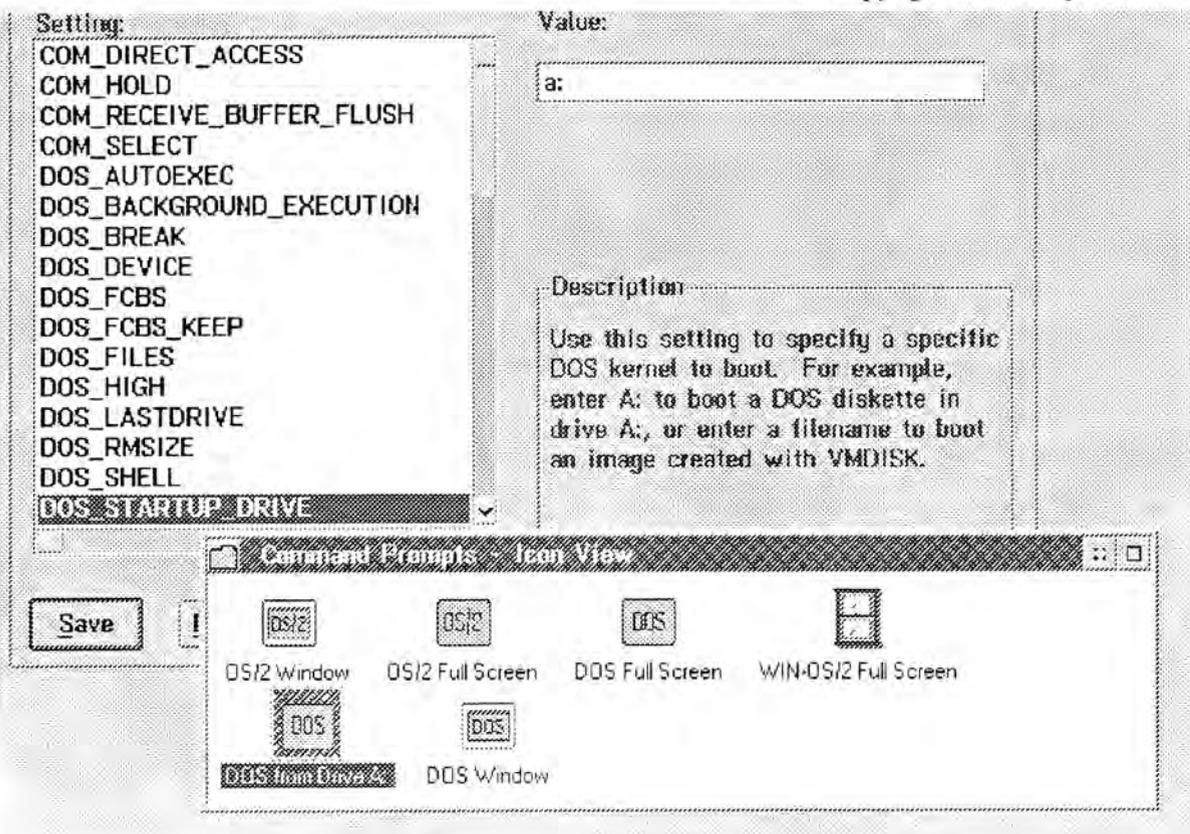


Figure 6-21. VM Boot from Diskette

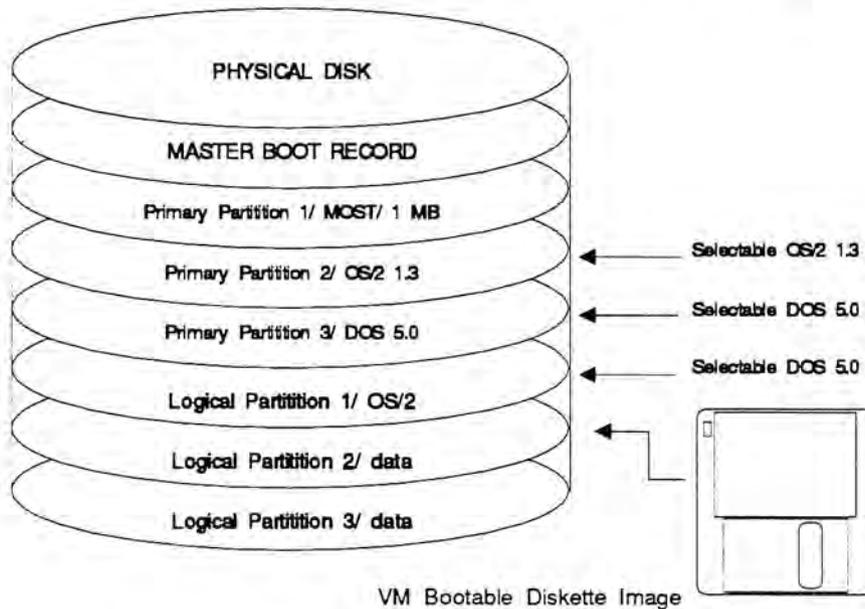
The following steps should be carried out to create a virtual machine boot diskette.

1. Reboot your system via Dual Boot or find a system which has been booted from the DOS version you desire on your diskette.
2. Format a diskette with the /S option.
3. Copy the following files to the diskette:
 - CONFIG.SYS
 - AUTOEXEC.BAT
 - \OS2\MDOS\FSFILTER.SYS
4. Edit the CONFIG.SYS and AUTOEXEC.BAT files as described later in this subtopic.

The booted DOS will always boot from a virtual A: drive and DOS will not search the C: drive unless the full path is specified. Therefore, be sure that all programs and device drivers that are referenced in CONFIG.SYS or AUTOEXEC.BAT have a full drive and path extension.

5. To verify that the diskette has been created properly, reboot the workstation with OS/2, insert the diskette in drive A:, and select *DOS from Drive A:* from the Command Prompts folder.

Partitioning for VM Boot



ps515208

Figure 6-22. VM Boot from Diskette Image

Once the user has verified that the boot diskette is operating properly, he/she may create a file that contains an image of that diskette using the VMDISK.EXE utility supplied with OS/2 V2.1.

A file image of a bootable diskette will boot more quickly than the diskette and eliminates the need to maintain the diskette.

When the VM boot is performed from a diskette image, the DOS that is booted into the VDM will see the image file as its A: drive. The use of the physical A: drive is, therefore, lost to that VDM boot session. The recommended method of correcting this problem is to use the FSACCESS.EXE utility.

FSACCESS.EXE remaps the A: to the OS/2 file system and restores the VDM DOS session's A: drive letter mapping to the physical A: drive. The effect of this remap, however, is that access to the image file is removed. DOS will be unable to reload COMMAND.COM. To avoid this dilemma, copy all the DOS files to a subdirectory on the hard disk and ensure the PATH and COMSPEC point there.

Modifications

```
CONFIG.SYS    DEVICE = FSFILTER.SYS  
                Before any other "DEVICE =" statement  
  
                HIMEM.SYS and EMM386.SYS  
                Use "OS2MDOS" provided drivers  
  
                MOUSE  
                Delete any mouse support statement  
  
                Use Full Drive & Extension  
  
AUTOEXEC.BAT C:\OS2MDOS\MOUSE  
  
                FSACCESS A:
```

ps515210

Figure 6-23. Required CONFIG.SYS Modifications

To enable VM Boot, some changes must be made to the CONFIG.SYS file that is on the disk volume that will be booted. The changes are required for two reasons:

1. A DOS which has been booted in a virtual machine must use the OS/2 file system to write to system DASD.

The **FSFILTER.SYS** device driver (supplied with OS/2 2.1) provides the access between the OS/2 and DOS file systems. **FSFILTER.SYS** must be loaded by a "DEVICE =" statement in the DOS CONFIG.SYS file.

2. A different set of device support programs are required for the **MOUSE**, **Extended Memory Services (XMS)** and **Expanded Memory Services (EMS)** in the virtual machine environment.

The drivers which work under native DOS will not work in the VM boot environment and conversely the VM boot drivers will not work under native DOS.

The CONFIG.SYS file used by OS/2 is **not** the configuration file that needs to be changed. The user must make these modifications on the CONFIG.SYS file on the diskette, or hard drive partition upon which the VM boot will be performed.

CONFIG.SYS MODIFICATIONS

FSFILTER.SYS is a device driver which manages DOS VDM access to OS/2 disks. FSFILTER.SYS should be copied from the \OS2\MDOS directory to the DOS diskette. Secondly, the following statement should be added to the CONFIG.SYS file of the bootable DOS diskette or image.

```
device = fsfilter.sys
```

1. Insert the line "**DEVICE = FSFILTER.SYS**" before any other "DEVICE =" statement in CONFIG.SYS. Be sure to copy the file FSFILTER.SYS from \OS2\MDOS\FSFILTER.SYS to the boot volume or provide a full path to it in the statement such as:

```
DEVICE = D:\OS2\MDOS\FSFILTER.SYS
```

2. If there is any "DEVICE =" statement which references the **DOS** driver **HIMEM.SYS** or the DOS driver **EMM386.SYS** in the CONFIG.SYS file, the following change must be made:

Edit any reference to the **DOS** XMS driver (HIMEM.SYS) or the **DOS** EMS driver (EMM386.SYS) so that it refers to the device drivers in \OS2\MDOS directory of the OS/2 boot volume.

The OS/2 versions of these device drivers will work only in a VM boot session and will not provide XMS or EMS services in a native DOS environment.

AUTOEXEC.BAT

```
C:\OS2\MDOS\MOUSE
```

```
FSACCESS A:
```

*dos 3 windows
end
vol # 4*

ps515212

Figure 6-24. AUTOEXEC.BAT Modifications.

These changes should be made to the virtual machine (VM) boot volume **AUTOEXEC.BAT** file:

1. If mouse support is needed, add this line to the AUTOEXEC.BAT file:

```
D:\OS2\MDOS\MOUSE
```

This assumes OS/2 is installed on C:. If OS/2 is installed on a different drive, change the drive letter accordingly. The OS/2 version of the mouse support will work only in a VM boot session. It will not provide mouse services in a native DOS environment.

2. Assuming the DOS files are in the C:\DOS directory, redirection would be accomplished by adding the following statements to the AUTOEXEC.BAT file:

```
SET COMSPEC = C:\DOS\COMMAND.COM  
SET PATH = C:\DOS; . . . other path directories  
:  
FSACCESS A:
```

3. It is recommended that the "FSACCESS A:" statement be included at the end of the AUTOEXEC.BAT. This statement releases the boot diskette and makes it available to another VM Boot session. Executing "FSACCESS A:" is equivalent to removing the boot diskette from the A: drive.

VM Diskette Image Utility

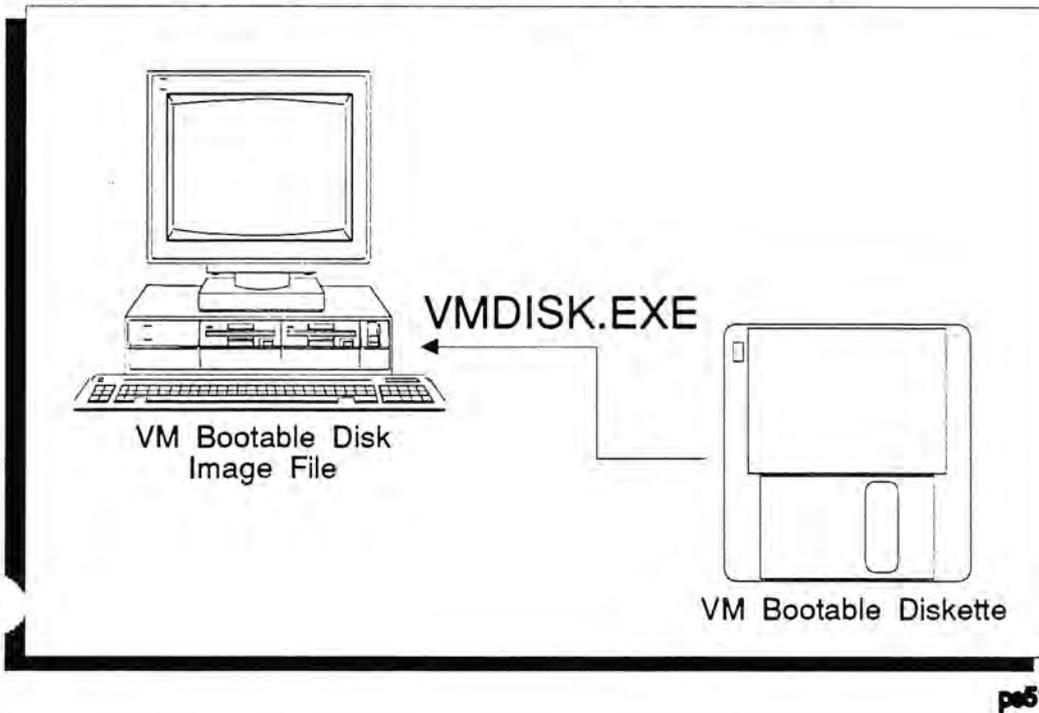


Figure 6-25. VMDisk Utility

The VMDisk utility provides the user with a method for transforming a bootable diskette into a file that contains the image of that diskette.

VMDISK is a command line (not PM) based application which runs under OS/2 1.x, OS/2 2.x, and DOS so as to facilitate transport of the image file to the OS/2 2.x environment. VMDISK supports 3.5 and 5.25 inch diskette formats.

The format of the command is as follows:

```
VMDISK source_drive: target_drive:\path\filename.ext
```

where:

- source_drive: is the diskette drive containing the bootable DOS diskette.
- target_drive: is the location and file name for the image file.

The file name must be specified. If the target drive and path are not specified, the current path will be used. Ensure that the target Drive has sufficient space for the image file.

The size of the diskette image is exactly the size of the diskette used. For Example, a 720 KB diskette is preferable to a 1.44 MB so as to save space on the hard disk which will store it.

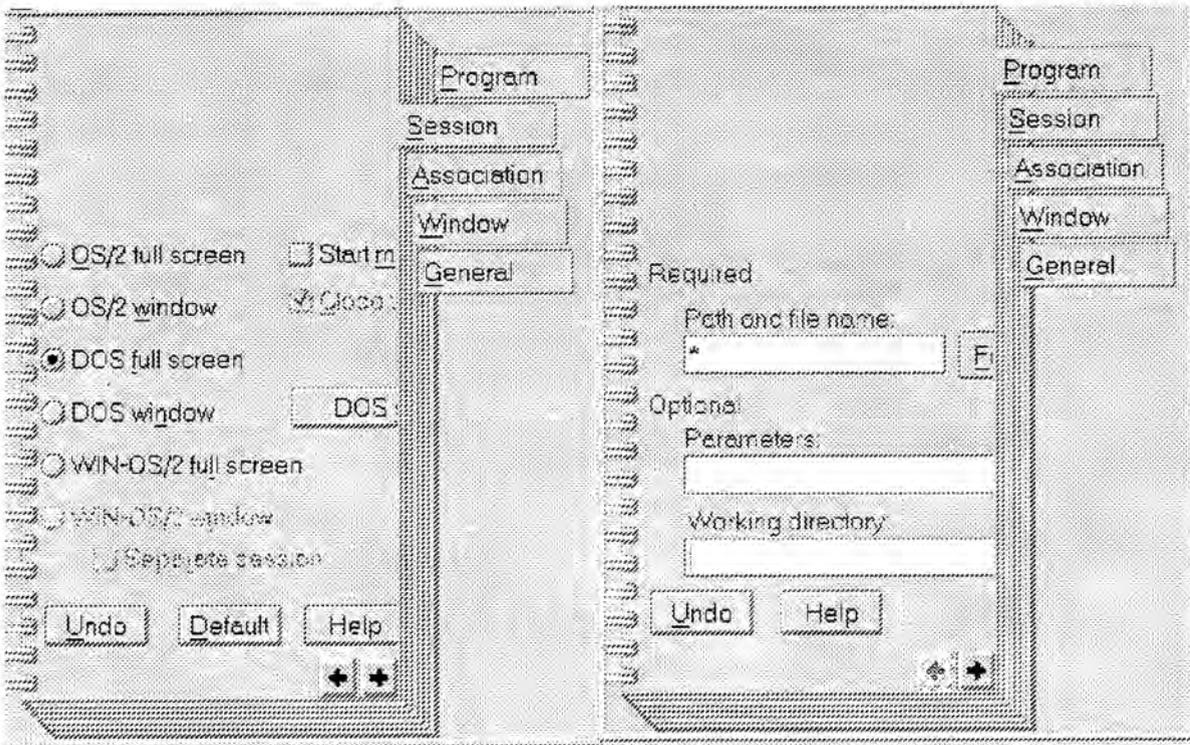


Figure 6-26. Setting Up a VM Boot

The user creates the VDM Boot object by creating a new DOS session entry in one of his program groups. In the figure above, notice the dialogue on the left. The fields of the *Program* dialogue should be filled as follows:

- Path and file name *
- Parameters (left blank)
- Working directory (left blank)

The *Session* dialogue (above right) is where the selection for "DOS Full Screen" or "DOS Window" is made. Next, the *DOS Settings* button is selected to configure the DOS environment.

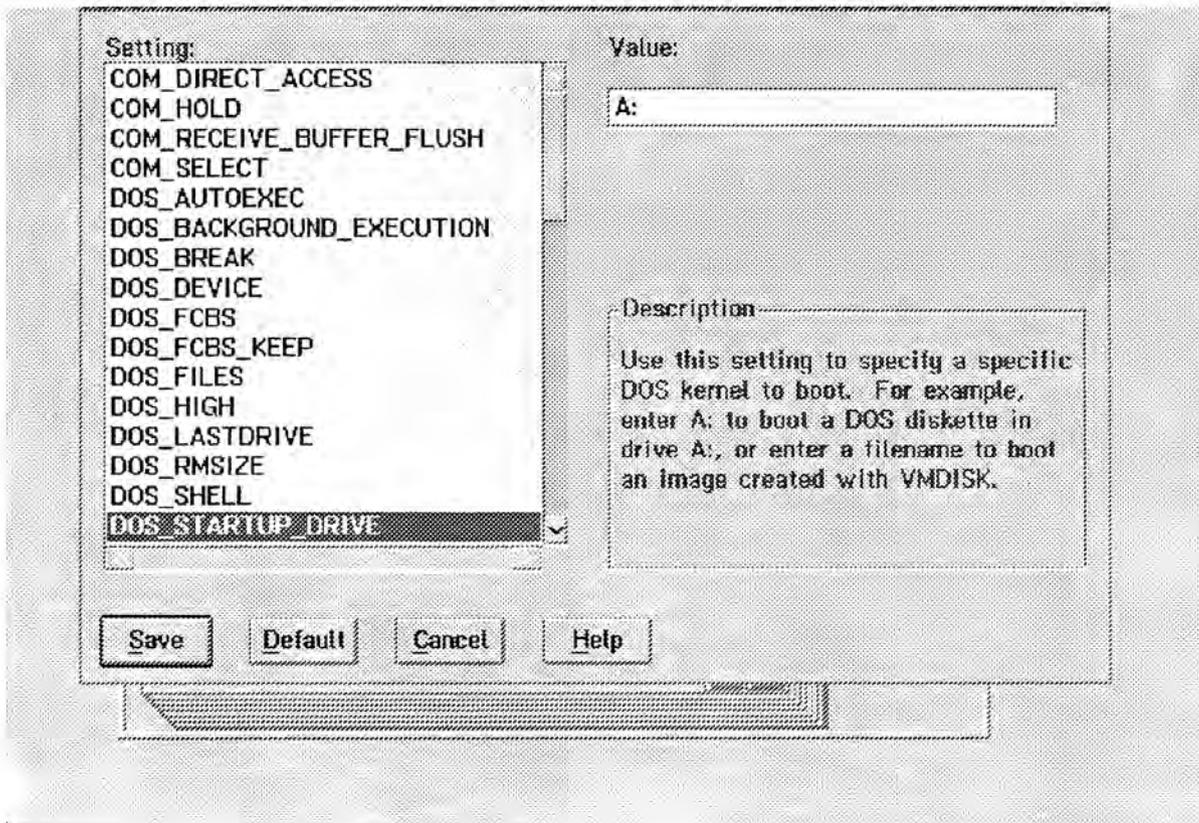


Figure 6-27. DOS Startup Drive

The VM boot feature introduces a new DOS property titled the "DOS Startup Drive." A physical diskette drive or the file name of a diskette image on a hard disk can be specified. The mapping "C:", with no path, will boot the partition boot record of the active "C:" partition. The syntax is as follows:

```
physical_drive:path;
```

Where:

- "physical_drive:path" is either the "C:" partition, a diskette, or the drive and path of a file that contains the image of the bootable diskette.

Examples:

- A:

Specifies that a virtual 86 machine should be booted from the A: drive.

- D:\DOS33.IMG

Specifies that a virtual 86 machine should be booted from the diskette image contained in file DOS33.IMG on the "D:" drive. Once booted, the contents of this file will appear as the "A:" volume from within that session.

- C:

Specifies that the partition boot record on the **active** "C:" partition should be booted.

Subtopic Summary

In this subtopic the student learned how to configure a virtual DOS machine boot session.

This subtopic illustrated and explained:

- the requirement for VDM Boot.
- 3 ways that the DOS boot record can be packaged for VDM booting.
- how to alter the DOS CONFIG.SYS and AUTOEXEC.BAT files.
- how to utilize the VMDISK utility to create an image file.
- how to create an object in a folder on the OS/2 desktop which causes DOS to be booted in a VDM when selected.

This concludes the third subtopic of the topic
"The DOS Environment."

Topic Summary

In this topic the student learned to tune the OS/2 2.1 DOS Environment and enable a Virtual DOS Machine Boot.

The student learned how to:

- Edit CONFIG.SYS and AUTOEXEC.BAT for VDM efficiency
- Use DOS Settings to optimize a DOS application running under OS/2 2.1's DOS environment
- Enable a DOS Virtual Machine Boot

This concludes the topic
"The DOS Environment."

TOPIC 7: The Win-OS/2 Environment

Topic objective:

Terminal objective:

After attending this topic the student should be able to provide a brief, general description of OS/2's Win-OS/2 as well as execute Windows applications on the OS/2 2.x platform.

Enabling objectives:

Upon completion of this topic the student should be able to:

- Describe IBM's WIN-OS2 environment.
- Install and execute Windows applications under OS/2.

's Win-OS/2 Environment

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to provide a general description of the IBM Win-OS/2 operating environment.

Enabling objectives:

After attending this subtopic the student should be able to describe:

- MAVDM
- SAVDM

in the context of the Win-OS/2 operating environment.

- Windows Applications Direct on the PM Desktop
- Multiple Application Virtual DOS Machines
- Single Application Virtual DOS Machines

ps516100

Figure 7-1. Win-OS/2 Implementation

Win-OS/2 supports Windows™ applications in one of three ways:

- In a Multiple Application Virtual DOS Machine (Fullscreen (MAVDM))
- In a Single Application Virtual DOS Machine (Fullscreen SAVDM)
- Seamlessly (Windowed)

- **Seamless Windowed Environment**
 - OS/2 Presentation Manager Screen Group
 - Seamless Migration to Windows Application Compatibility

- **Full Screen Environment in OS/2 2.1**
 - Maximum Compatibility for Windows 3.1
 - Standard or Enhanced Mode

ps516102

Figure 7-2. Windowed and Fullscreen Solutions

The **"Windowed Environment"** will accommodate Windows™ applications executing in the OS/2 Presentation Manager screen group. It is this environment that provides the "seamless" solution to Windows™ application compatibility.

The **"Full Screen Environment"** is designed to provide maximum compatibility for Windows™ applications that execute under Windows 3.1™. This solution provides the user with a more traditional Windows™ environment than that provided by the "seamless" solution.

- **WINOS2.COM**
 - Starts Win-OS/2 3.1
 - Loads VDM environment for Win-OS/2 3.1

- **OS2K386.EXE**
 - Replaces Win-OS/2 3.0 Kernel (OS2K286.EXE)
 - Full screen
 - Seamless

ps516104

Figure 7-3. The IBM Win-OS/2 Kernel

The ability to run Windows™ applications required that some changes be made to OS/2. OS/2 1.x could only run Windows™ applications in real mode. This level of support is inadequate to run all Windows™ applications unmodified.

Some applications and device drivers required a standard mode Windows™ environment. The features added subsequently to OS/2 enable the level of support necessary to run those applications.

The Windows™ kernel (co-owned by IBM) has been modified to use the DOS Protect Mode Interface (DPMI) for memory management and interrupt management. This modified kernel executes in a full screen VDM using VDM architecture with DOS emulation.

Both standard and enhanced mode of the Windows 3.1™ kernel are supported in this VDM.

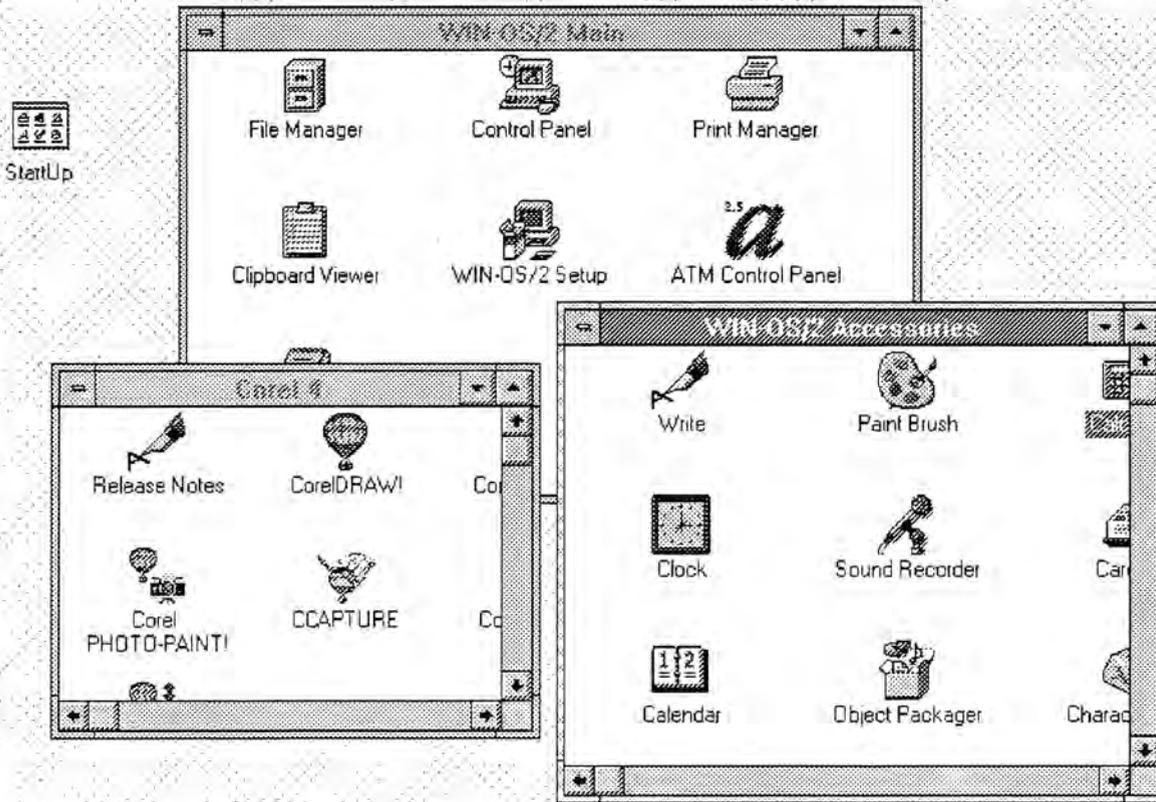


Figure 7-4. Full Screen MAVDM

The Multiple Application Virtual DOS Machine (MAVDM) full screen implementation of Win-OS/2 3.1 is almost identical to running DOS/Windows 3.1™. The Win-OS/2 3.1 Program Manager is used to start multiple Windows applications within the same VDM. The MAVDM uses the built in kernel and is launched from a VDM. This implementation provides the maximum "look and feel" compatibility for the DOS/Windows™ user migrating to OS/2 Version 2.x.

The kernel can be started in **standard or enhanced** mode. DPMI is automatically activated. As is shown in the figure above, the Win-OS/2 Program Manager is running. Multiple Windows applications can, therefore, run under this environment. Those applications can share data.

Many of the Windows 3.1™ applets are now included with Win-OS/2 3.1. Users can also enjoy the additional benefits from the improved OLE support, multimedia support for audio and TrueType font support.

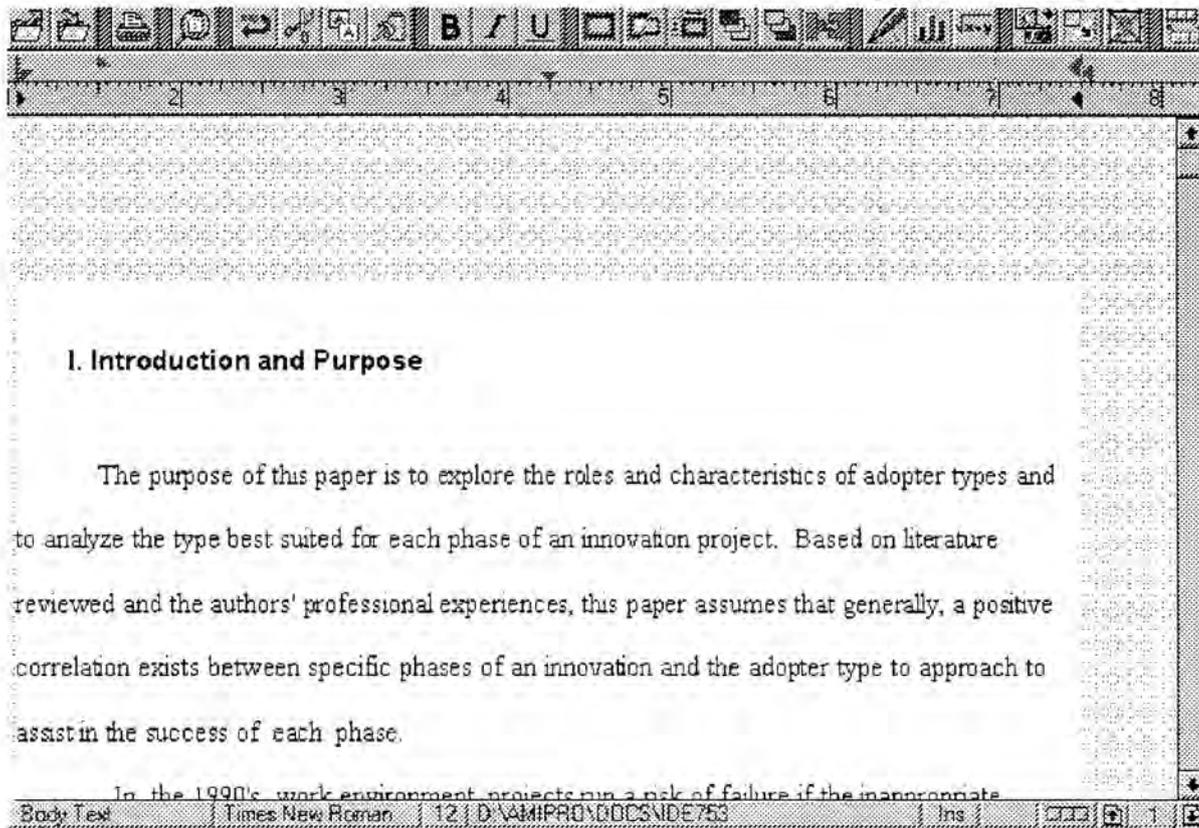


Figure 7-5. Full Screen SAVDM

The Single Application Virtual DOS Machine (SAVDM) full screen implementation of OS/2 Version 2.1's Win-OS/2 is the recommended way of running Windows™ applications under OS/2 and is started directly when the Windows™ application is executed from an OS/2 command line or from an OS/2 Work Place Shell icon.

The kernel can be started in **standard or enhanced** mode. As is shown in the figure above, the Program Manager will **not** be started because only one application will be running under this environment.

By running in SAVDMs, Windows™ applications are timesliced more effectively because one of them is under the control of OS/2's **pre-emptive** multitasking scheduler. In a MAVDM environment, all Windows™ applications in there are still subject to the **cooperative** multitasking design of Windows™.

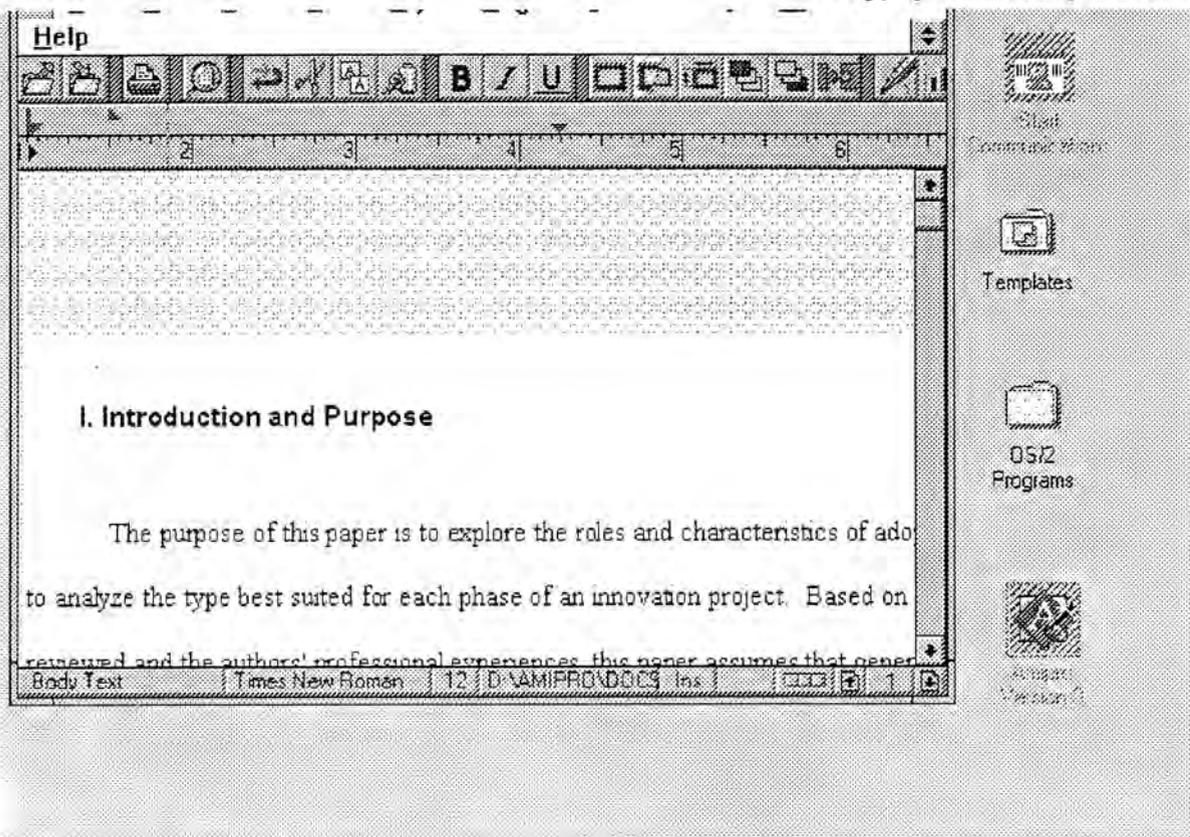
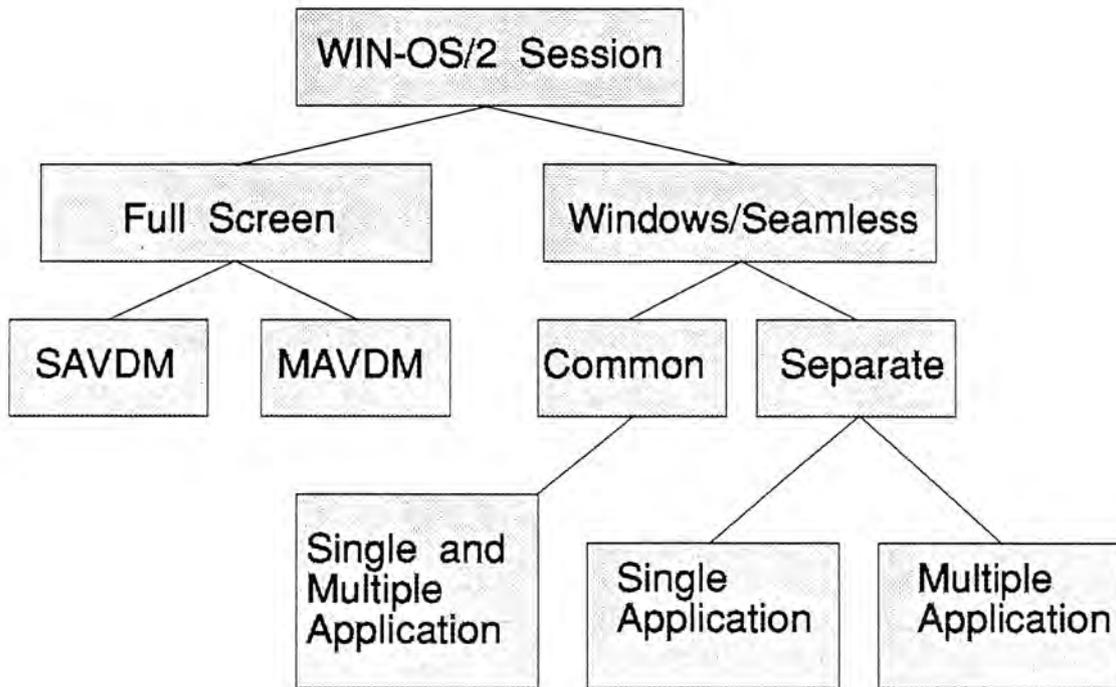


Figure 7-6. Seamless Solution

The SAVDM provides a seamless approach to Presentation Manager integration. The application is loaded from the Workplace Shell in a very similar way to a DOS application and the user may easily switch back to Presentation Manager, as well as share data via the clipboard or DDE with other Windows™ or Presentation Manager applications.

OS/2 Version 2.1 provides the capability for Windows™ applications to run **seamlessly**. That is, the Windows™ application will execute on the **OS/2** desktop with no apparent difference between it and PM applications.

Like the fullscreen SAVDM and MAVDM solutions, the seamless implementation is a protected mode process subject to the same application protection facilities that OS/2 provides protected mode (i.e. OS/2) applications. Windows applications are protected from DOS and OS/2 applications executing in the system.



ps516112

Figure 7-7. WIN-OS/2 Sessions

A Win-OS/2 3.1 session can be either full screen or seamless (windowed).

It is recommended that the OS/2 2.1 Migrate Applications utility be used once Windows applications have been installed. Migrate Applications sets up the Win-OS/2 3.1 settings correctly for the application and creates an icon in the *Windows Applications* folder.

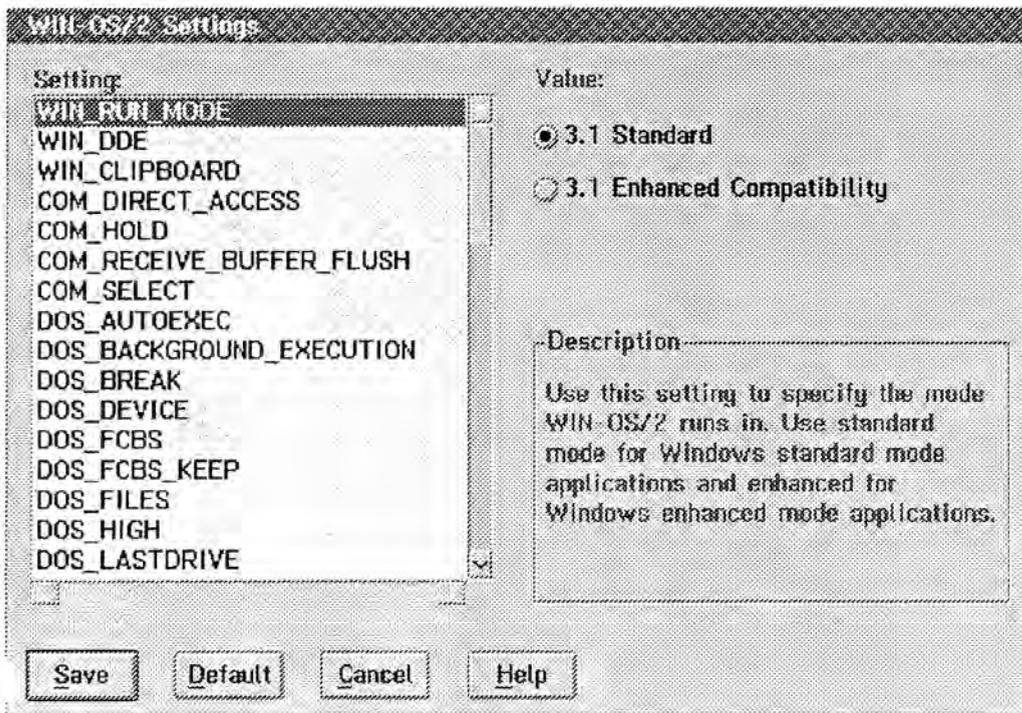


Figure 7-8. WIN-OS/2 Mode

Windows applications default to standard mode. Display the WIN-OS/2 Settings option from within the WIN-OS/2 Settings notebook. The **WIN_RUN_MODE** parameter can be changed to specify enhanced mode.

Another way to start a Win-OS/2 3.1 application is to type the application name from an OS/2 or DOS command prompt as follows:

```
[C:\>]WINOS2 /E PROGRAM.EXE
```

for enhanced mode launching or,

```
[C:\>]WINOS2 /3 PROGRAM.EXE
```

for standard mode launching.

Subtopic Summary

In this subtopic the student learned about the IBM WINOS2.COM operating environment that supports the execution of Windows™ applications.

The following terms were described and discussed in the context of the Win-OS2 operating environment.

- Standard mode
- Enhanced mode
- MAVDM
- SAVDM
- Seamless

This wraps up the subtopic of the topic
"The Win-OS/2 Environment".

Running Applications Under Win-OS2

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to install and execute Windows applications under OS/2.

Enabling objectives:

After attending this subtopic the student should be able to:

- Install OS/2 so it supports Windows applications.
- Open the appropriate Workplace Shell object to customize a Windows application.
- Add a subsequent Windows application to an existing OS/2 Version 2.1 system.
- Launch a Windows application.

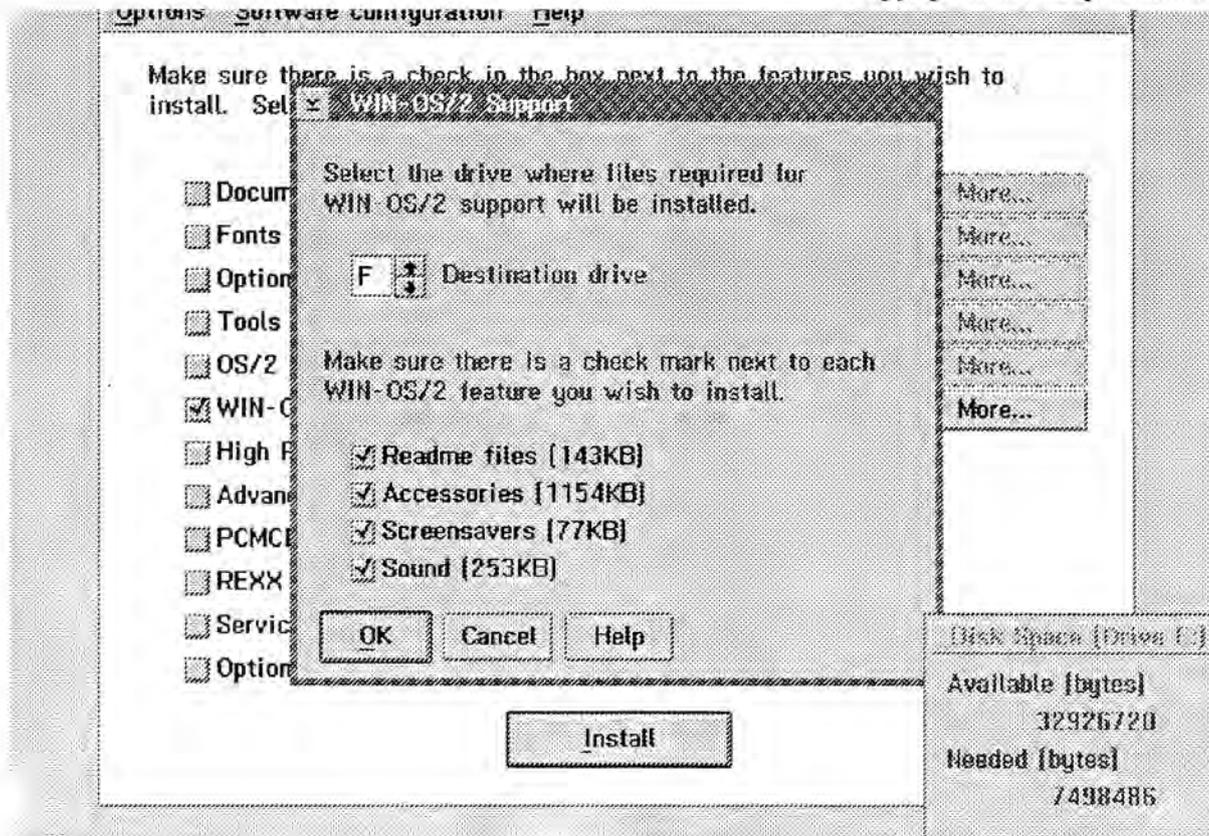


Figure 7-9. Win-OS/2 Support

Win-OS/2 is provided by default during the installation of OS/2 Version 2.1. If the user marks the *WIN-OS/2 Support* checkbox, all the files necessary to provide Win-OS/2 support will be installed in the following subdirectories:

- \OS2\MDOS\WINOS2
- \OS2\MDOS\WINOS2\SYSTEM

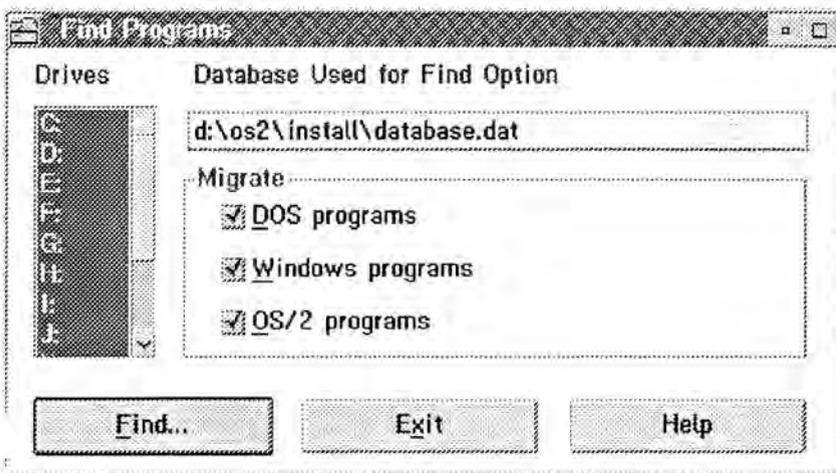


Figure 7-10. The Migration Utility

During initial installation of OS/2 Version 2.1, the migration utility will search the path statement in the user's current AUTOEXEC.BAT file. If a Windows™ directory is referenced in there, the current *.INI and *.GRP files will be read and the necessary changes will be applied to them. The updated versions will be stored in the

`\OS2\MDOS\WINOS2`

subdirectory. These changes will effectively migrate the user's Windows Desktop, including all their Windows applications into an IBM Win-OS2 MAVDM environment.

Upon completion of the installation process, the user is given the opportunity to migrate installed Windows applications (defined to the Windows Program Manager) to the OS/2 Version 2.1 Workplace Shell. All Windows™ applications that are to be migrated must have the appropriate DOS and Windows™ settings defined in the **Certified Application Database (CAD)**, that is shipped as a standard component of OS/2 Version 2.1.

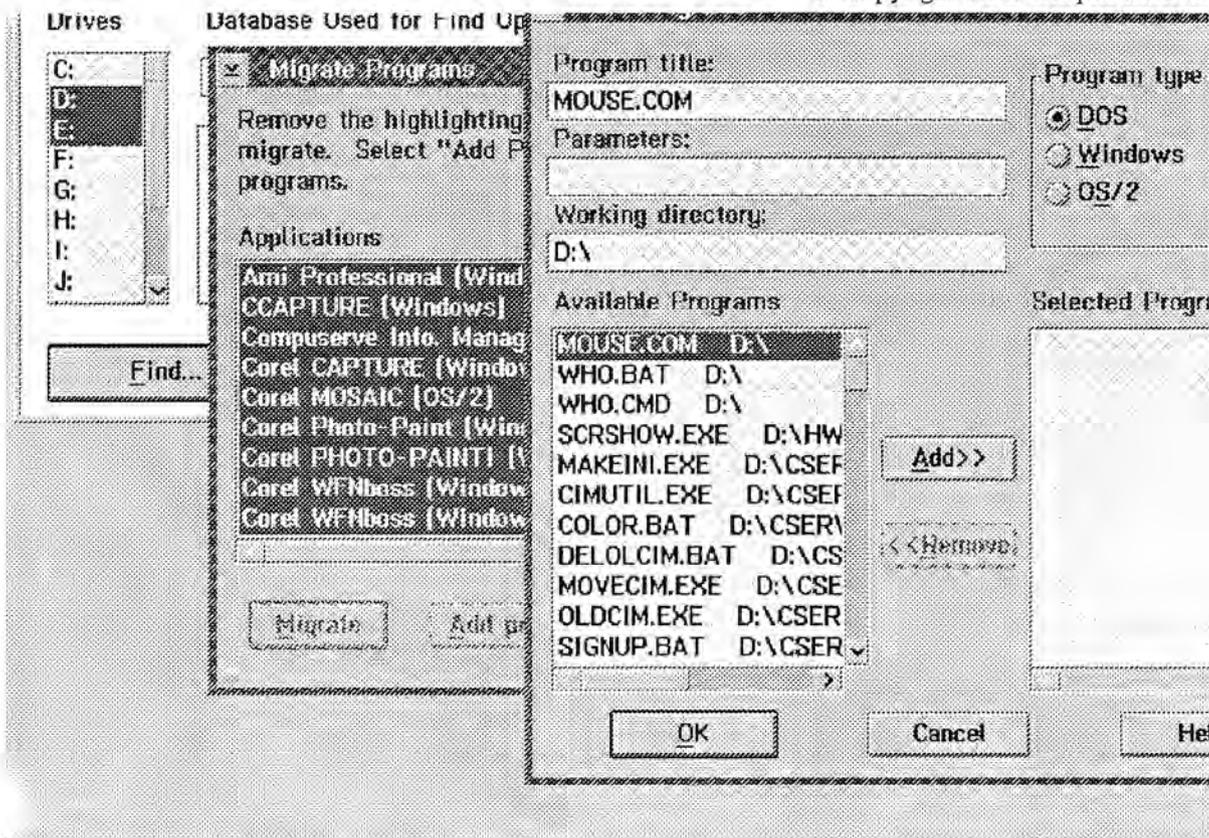


Figure 7-11. Adding Application to CAD

In the event that a Windows application is installed and that application is not listed in the CAD, the Migration Utility can be used to add the application to the CAD enabling the utility to then migrate the application.

If the system is already installed and the application was subsequently added:

1. Select the Migration Utility from the System Setup folder.
2. Indicate the search drives and options in the dialog and select *Find*.
3. When the Migrate Programs dialog appears, select *Add Programs*.
4. The Add Programs dialog will list available programs.
 - a. Highlight the program you wish to add to the CAD.
 - b. Select *Add*
 - c. Select *Ok*
5. The Migrate Programs dialog will return with the new application listed. Select *Migrate*

The new application will now have a program reference icon on the Desktop and will run in a SAVDM. If the system is so enabled, it will run seamlessly.

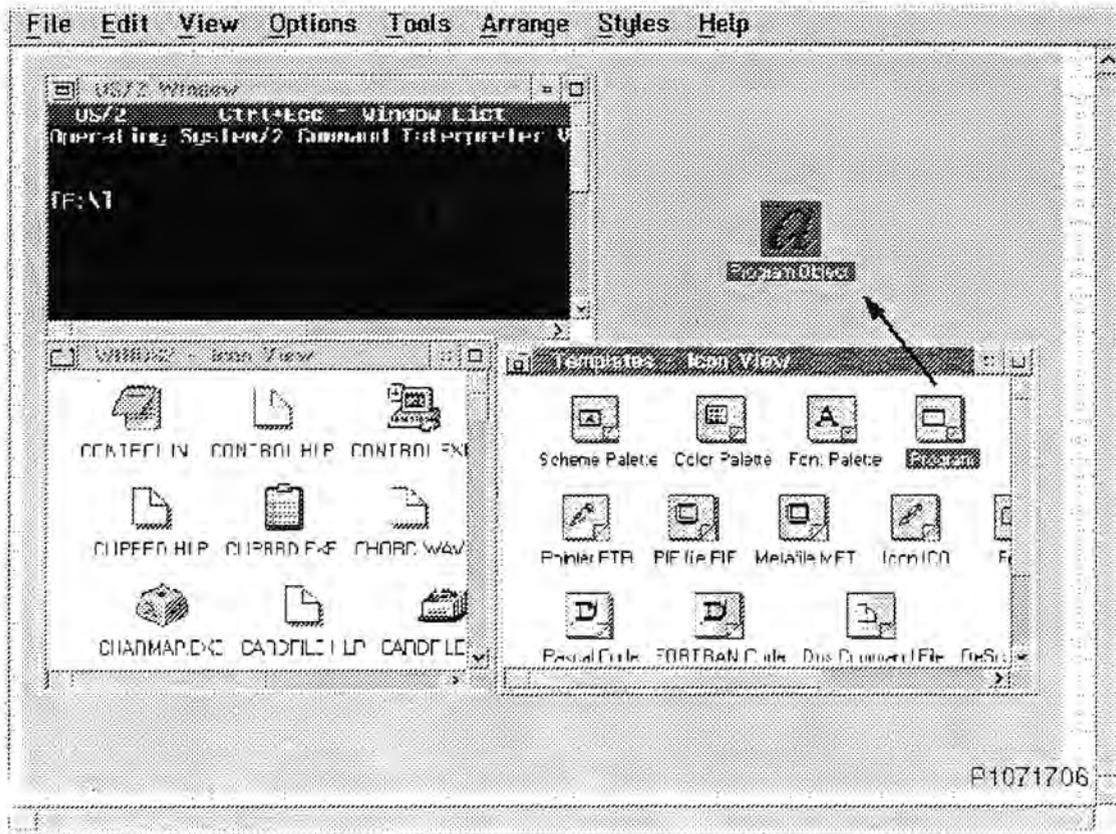


Figure 7-12. Launching Applications

The following methods may be used to start Windows applications:

- Enter the application name at an OS/2 command prompt.
- Select the application's program object icon from a Drives folder.
- Install the application in a folder on the Workplace Shell desktop.

If the application is started from either the OS/2 Drives folder or an OS/2 command prompt, a SAVDM will be created. If the application is started from an icon, either a SAVDM or a MAVDM will be created.

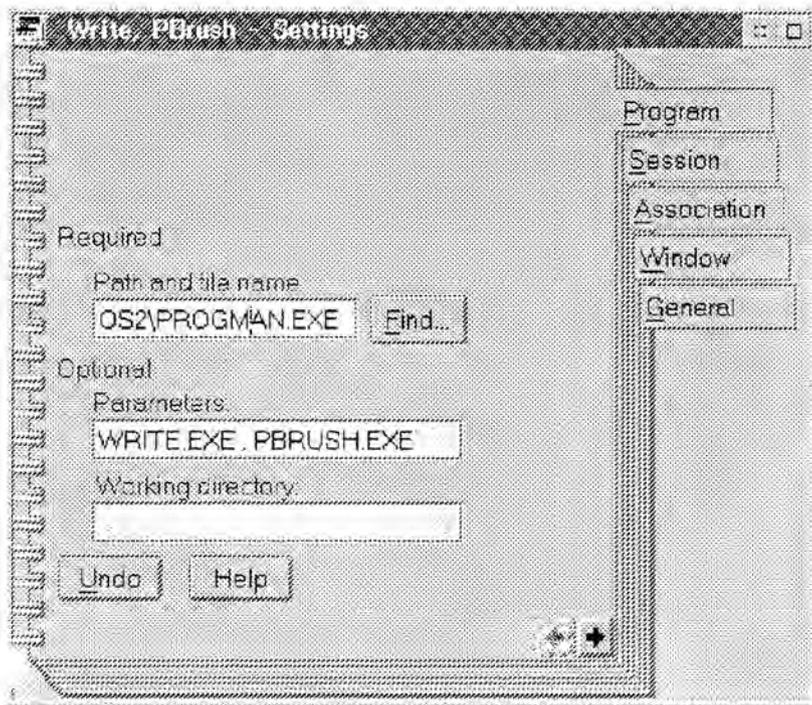


Figure 7-13. Defining Program Objects

There are two types of program object definitions. One for the single application environment and the other for the multiple applications definition.

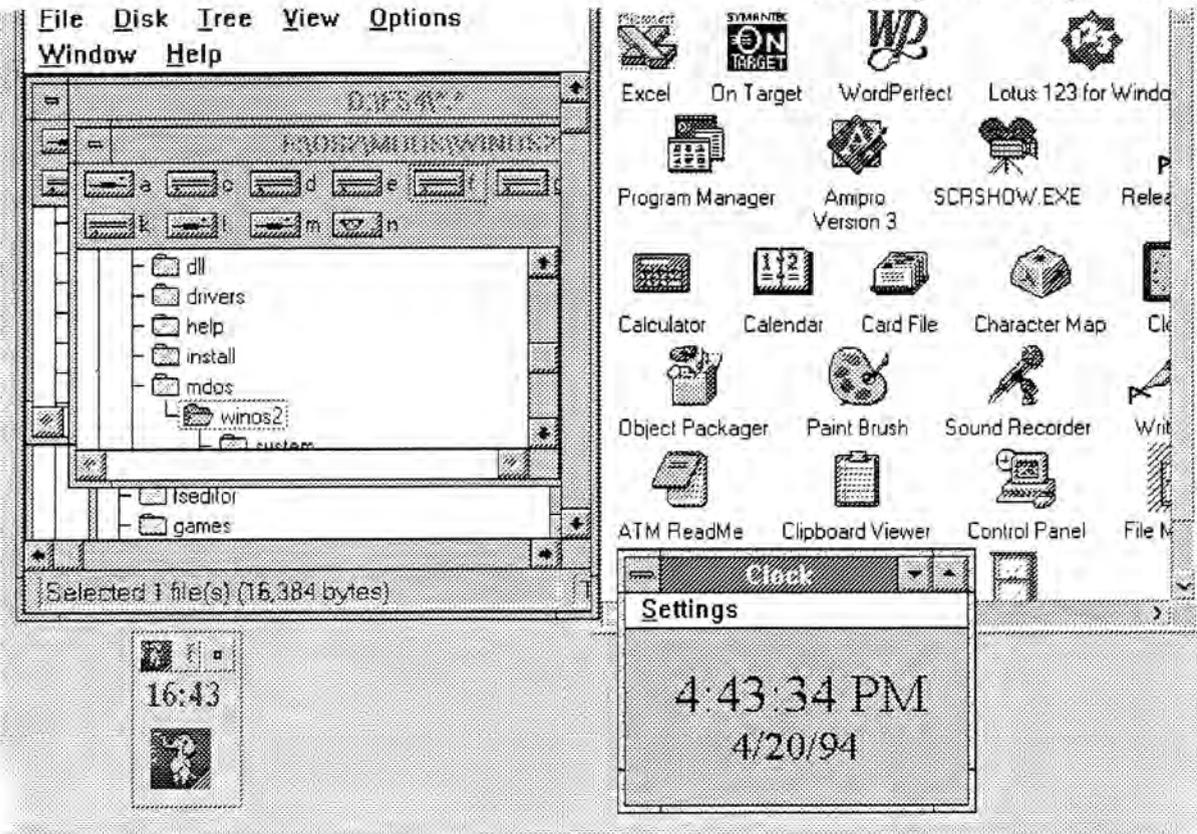


Figure 7-14. Starting DOS and OS/2 Apps

Some Windows 3.1™ applications rely on their ability to call DOS utility programs for basic and utility functions. WIN-OS/2 3.0 did not provide that level of support.

Subtopic Summary

In this subtopic the student learned to install and execute Windows applications under OS/2 Version 2.1's Win-OS2. The following points were covered:

- Installation of OS/2 so it supports Windows applications.
- Migration of existing Windows applications to the Win-OS2 environment.
- The Workplace Shell object used to customize a Windows application.
- Adding a subsequent Windows application to an existing OS/2 Version 2.1 system.
- Launching a Windows application.

This concludes another subtopic of the topic
"The Win-OS2 Environment".

Topic Summary

In this topic the student learned about OS/2's Win-OS/2 environment as well as how to execute Windows applications in OS/2.

This topic discussed:

- The Win-OS2 environment.
- Installation and execution of Windows applications.

This concludes the topic
"The Win-OS/2 Environment."

TOPIC 8: Migrating applications

Topic objective:

Terminal objective:

After attending this topic the student should be able to provide a brief, general description of how to migrate applications to the Desktop and how to create their own application migration database.

Enabling objectives:

Upon completion of this topic the student should be able to:

- Describe the steps to migrate applications to the Desktop
- Create a simple migration database

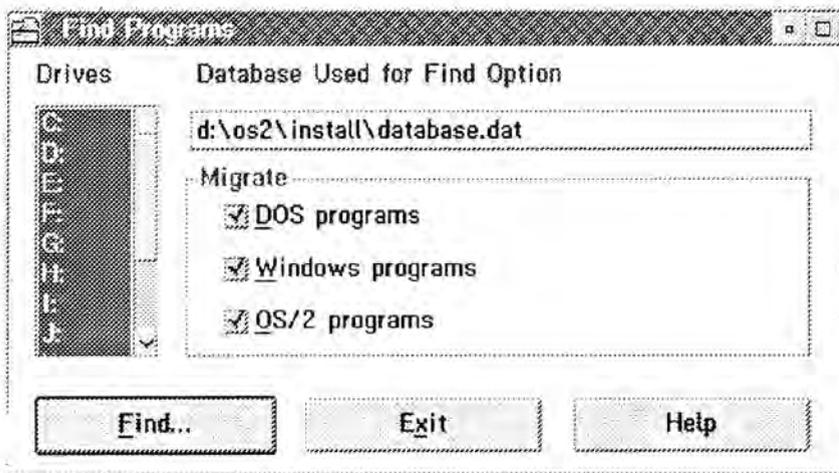


Figure 8-1. The Migration Utility

If you have already previously installed programs on your hard drive, and did not choose to format your hard drive during installation, program references are created and saved in folders on the Desktop. This function is performed by the Migrate Applications program. During installation, the migration program will also initialize settings for programs using the application database located in the \OS2\INSTALL\DATABASE.DAT.

Program folders are created for DOS, Windows, and OS/2 applications found on the hard drive during migration, including WIN-OS2 groups if WIN-OS2 is installed.

Additional folders are created for applications not found in the database, but which are manually added to the system using the Add Programs function of the Migration Application program. Settings for these programs will be the system defaults.

Users can also add or modify this database for their own applications. The PARSEDB command is used to create the database. PARSEDB uses a processing file (DBTAGS.DAT) and an ASCII input file to create the Database. Procedures are located in the Master Help Index and in the Command Reference.

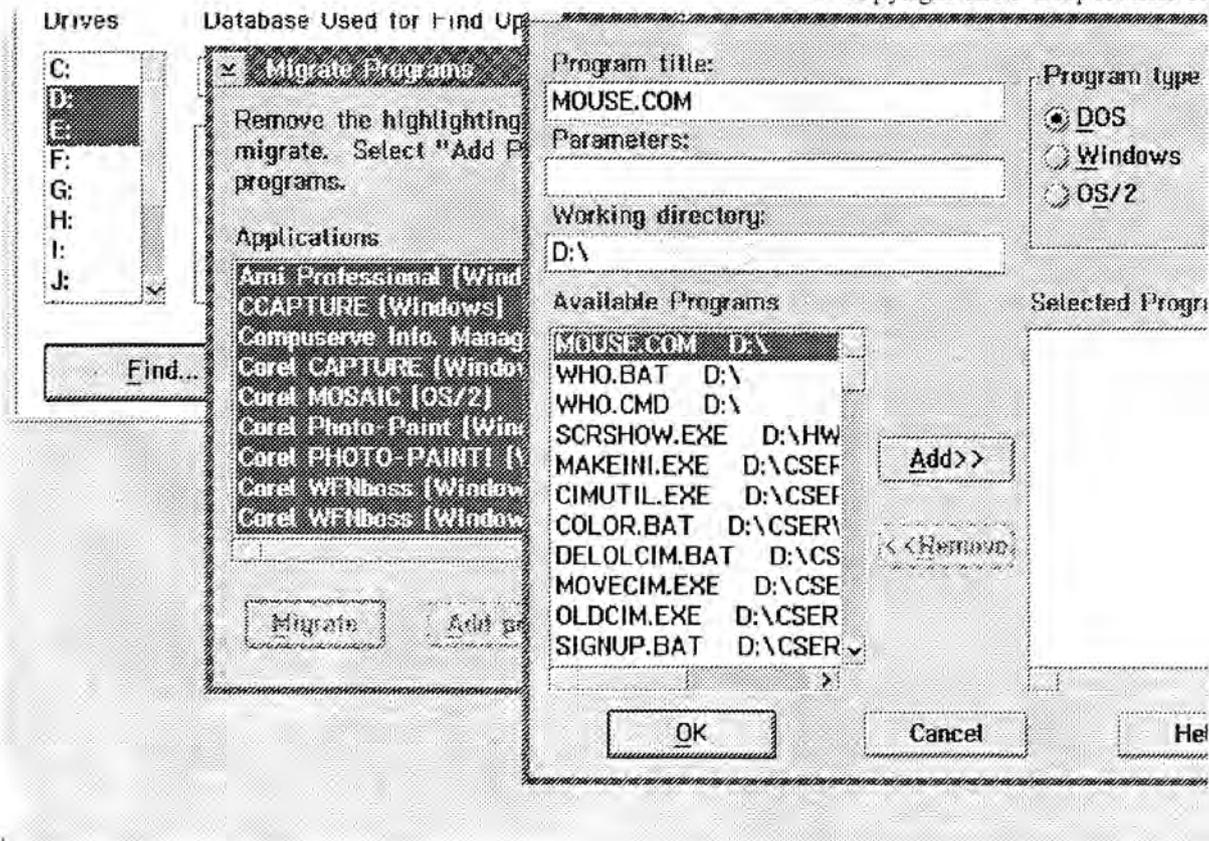


Figure 8-2. Adding Application to CAD

In the event that a Windows application is installed and that application is not listed in the CAD, the Migration Utility can be used to add the application to the CAD enabling the utility to then migrate the application.

If the system is already installed and the application was subsequently added:

1. Select the Migration Utility from the System Setup folder.
2. Indicate the search drives and options in the dialog and select *Find*.
3. When the Migrate Programs dialog appears, select *Add Programs*
4. The Add Programs dialog will list available programs.
 - a. Highlight the program you wish to add to the CAD.
 - b. Select *Add*
 - c. Select *Ok*
5. The Migrate Programs dialog will return with the new application listed. Select *Migrate*

The new application will now have a program reference icon on the Desktop and will run in a SAVDM. If the system is so enabled, it will run seamlessly

Topic Summary

In this topic the student learned about migrating their applications to the Desktop.

This topic discussed:

- The steps to migrate applications to the Desktop
- How to create a migration database

This concludes the topic
"Migration of Applications."

1 bit

4

2.16 bits to rep 1 pix.

1024 x 1024

16M colors

TOPIC 9: OS/2 Configuration File

Topic objectives

Terminal objective:

After attending this topic the student should have a general understanding of the meanings of the parameters in the OS/2 configuration file discussed.

Enabling objectives:

Upon completion of this topic the student should be able to describe the meaning of each parameter found in the OS/2 configuration file.

Prerequisite knowledge

The student should have attended the previous topics in this course.



OS/2 Initialization Files

subtopic objectives

Terminal objective:

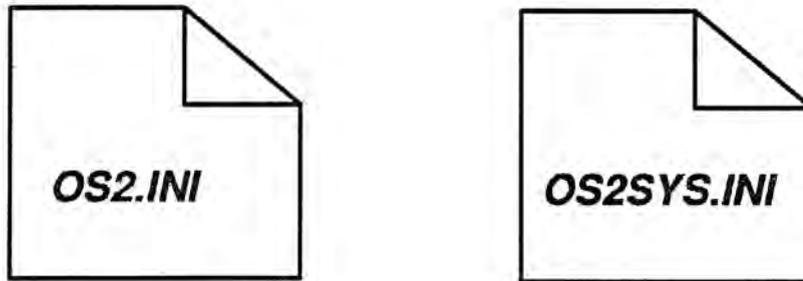
After attending this subtopic the student should be able to explain what can be found in the OS/2 initialization files.

Enabling objectives:

After attending this subtopic the student should be able to explain the

- contents of the OS2.INI file
- contents of the OS2SYS.INI file
- procedure available to recreate these two files

2 Files:



located in the \OS2 subdirectory

ps512b12

Figure 9-1. Inside the initialization Files

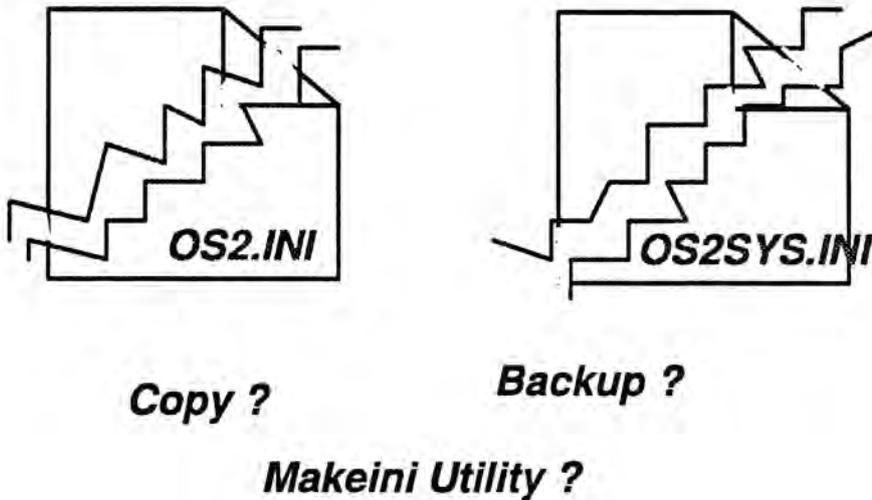
These are two binary files extensively used by the operating system and by applications.

System settings, such as application defaults, display options, and file options, are contained in the OS2.INI startup file located in the OS2 directory of your hard disk.

There is also a system file called OS2SYS.INI, which contains information regarding the installed fonts and printer drivers.

ALT + F1 wait til keyboard beeps

What if...



ps512b13

Figure 9-2. What if...

If you receive a message stating that the OS2.INI file is corrupted, the OS2.INI file installed on your system must be replaced by another valid copy of the OS2.INI file.

Added to your system during the installation process, the MAKEINI.EXE file creates a new OS2.INI file containing default information. You can re-create both the user and the system INI files using MAKEINI (located in the OS2 directory of your hard disk). If you decide to use this utility, the defaults will be those of a newly installed OS/2 2.1. Any folders, color preferences etc. set up previously will not appear.

One can also protect their INI files by having them automatically backed up each time you start the system. For instance, in the CONFIG.SYS file, a backup copy of your current INI files and a backup of the INI files as they existed at the previous system startup can be made.

- `CALL = D:\OS2\XCOPY.EXE D:\OS2*.INX D:\OS2*.INX`
- `CALL = D:\OS2\XCOPY.EXE D:\OS2*.INI D:\OS2*.INX`

*INI → *.INA.*

*4 levels of
copy to be
just sure!!!*

Topic Summary

In this subtopic the student were introduced to the importance of the OS/2 initialization files.

This concludes the first subtopic of the topic
"OS/2 Configuration file".

CALL = c:\OS2\XCOPY.

15

make ini target, source.

OS2INT

OS2.RC

edit ini

BBS.

OS/2 configuration file parameters

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to explain the purpose of parameters found in the OS/2 configuration file that have not yet been discussed in this course.

Enabling objectives:

After attending this subtopic the student should be able to explain the

- parameters not yet discussed in the previous subtopics

CURRENT OS/2 CONFIGURATION FILE

- IFS = D:\OS2\HPFS.IFS /CACHE:512 /CRECL:4 /AUTOCHECK:G
- PROTSHELL = D:\OS2\PMSHELL.EXE
- SET USER_INI = D:\OS2\OS2.INI
- SET SYSTEM_INI = D:\OS2\OS2SYS.INI
- SET OS2_SHELL = D:\OS2\CMD.EXE
- SET AUTOSTART = PROGRAMS,TASKLIST,FOLDERS
- SET RUNWORKPLACE = D:\OS2\PMSHELL.EXE
- SET COMSPEC = D:\OS2\CMD.EXE
- LIBPATH = .;D:\OS2\DLL;D:\OS2\MDOS;D:\OS2\APPS\DLL;
- SET
PATH = D:\OS2;D:\OS2\SYSTEM;D:\OS2\MDOS\WINOS2;D:\OS2\INSTALL;D:\OS2\MDOS;D:\OS2\APPS;
- SET
DPATH = D:\OS2;D:\OS2\SYSTEM;D:\OS2\MDOS\WINOS2;D:\OS2\INSTALL;D:\OS2\BITMAP;D:\OS2\MDOS;D:\OS2\APPS;
- SET PROMPT = \$i \$p
- SET HELP = D:\OS2\HELP;D:\OS2\HELP\TUTORIAL;
- SET GLOSSARY = D:\OS2\HELP\GLOSS;
- PRIORITY_DISK_IO = YES
- FILES = 20
- DEVICE = D:\OS2\TESTCFG.SYS
- DEVICE = D:\OS2\DOS.SYS
- DEVICE = D:\OS2\PMDD.SYS
- BUFFERS = 30
- IOPL = YES
- DISKCACHE = 512,LW
- MAXWAIT = 3
- MEMMAN = SWAP,PROTECT
- SWAPPATH = D:\OS2\SYSTEM 2048 3072
- BREAK = OFF
- THREADS = 256
- PRINTMONBUFSIZE = 134,134,134
- COUNTRY = 001,D:\OS2\SYSTEM\COUNTRY.SYS
- SET KEYS = ON
- REM
DELDIR = C:\DELETE,512;D:\DELETE,512;E:\DELETE,512;F:\DELETE,512;

*Installable
File System
needs to be*

FAT cache.

- BASEDEV = PRINT02.SYS
- BASEDEV = IBM2FLPY.ADD
- BASEDEV = IBM2ADSK.ADD
- BASEDEV = OS2DASD.DMD
- BASEDEV = OS2SCSI.DMD
- SET BOOKSHELF = D:\OS2\BOOK
- SETPATH = D:\OS2\APPS
- DEVICE = D:\OS2\APPS\SASYNCD.B.SYS
- PROTECTONLY = NO
- SHELL = D:\OS2\MDOS\COMMAND.COM D:\OS2\MDOS
- FCBS = 16,8
- RMSIZE = 640
- DEVICE = D:\OS2\MDOS\VEMM.SYS
- DEVICE = D:\OS2\MDOS\VMOUSE.SYS
- DOS = LOW,NOUMB
- DEVICE = D:\OS2\MDOS\VDPX.SYS
- DEVICE = D:\OS2\MDOS\VXMS.SYS /UMB
- DEVICE = D:\OS2\MDOS\VDPMI.SYS
- DEVICE = D:\OS2\MDOS\VWIN.SYS
- DEVINFO = SCR,VGA,D:\OS2\VIOTBL.DCP
- SET VIDEO_DEVICES = VIO_VGA
- SET VIO_VGA = DEVICE(BVHVGA)
- DEVICE = D:\OS2\MDOS\VVGA.SYS
- DEVICE = D:\OS2\POINTDD.SYS
- DEVICE = D:\OS2\MOUSE.SYS
- DEVICE = D:\OS2\COM.SYS
- DEVICE = D:\OS2\MDOS\VCOM.SYS
- CODEPAGE = 437,850
- DEVINFO = KBD,US,D:\OS2\KEYBOARD.DCP

device drivers.
These load 1st

for modem.

works with Country.SYS.
key board

set video-device v10-svga
set v10-vga=dev(bhuga,bkgvsg)
video support.



CONFIG.SYS

```
SET RUNWORKPLACE=D:\OS2\PMSHELL.EXE  
SET USER_INI=D:\OS2\OS2.INI  
SET SYSTEM_INI=D:\OS2\OS2SYS.INI  
SET OS2_SHELL=D:\OS2\CMD.EXE  
SET AUTOSTART=PROGRAMS, TASKLIST,  
FOLDERS
```

ps512a21

Figure 9-3. OS/2 Parameters

*Take out
Programs
so they
don't annoy
start up*

CONFIG.SYS

```
SET PATH=D:\OS2;D:\OS2\SYSTEM;  
D:\OS2\MDOS\WINOS2;D:\OS2\INSTALL;  
D:\;D:\OS2\MDOS;D:\OS2\APPS;  
  
SET DPATH=D:\OS2;D:\OS2\SYSTEM;  
D:\OS2\MDOS\WINOS2;D:\OS2\INSTALL;  
D:\;D:\OS2\BITMAP;D:\OS2\MDOS;  
D:\OS2\APPS;
```

ps512a22

Figure 9-4. OS/2 Parameters

- **SET PATH =**

This sets a search path for commands and programs. Setting this Path in the config.sys and autoexec.bat files lets you avoid having to set PATH from the command prompt each time you turn on your system.

- **SET DPATH =**

Gives application programs the search path to data files that are outside the current directory. This environment variable can only be set using the SET command in OS/2 sessions.

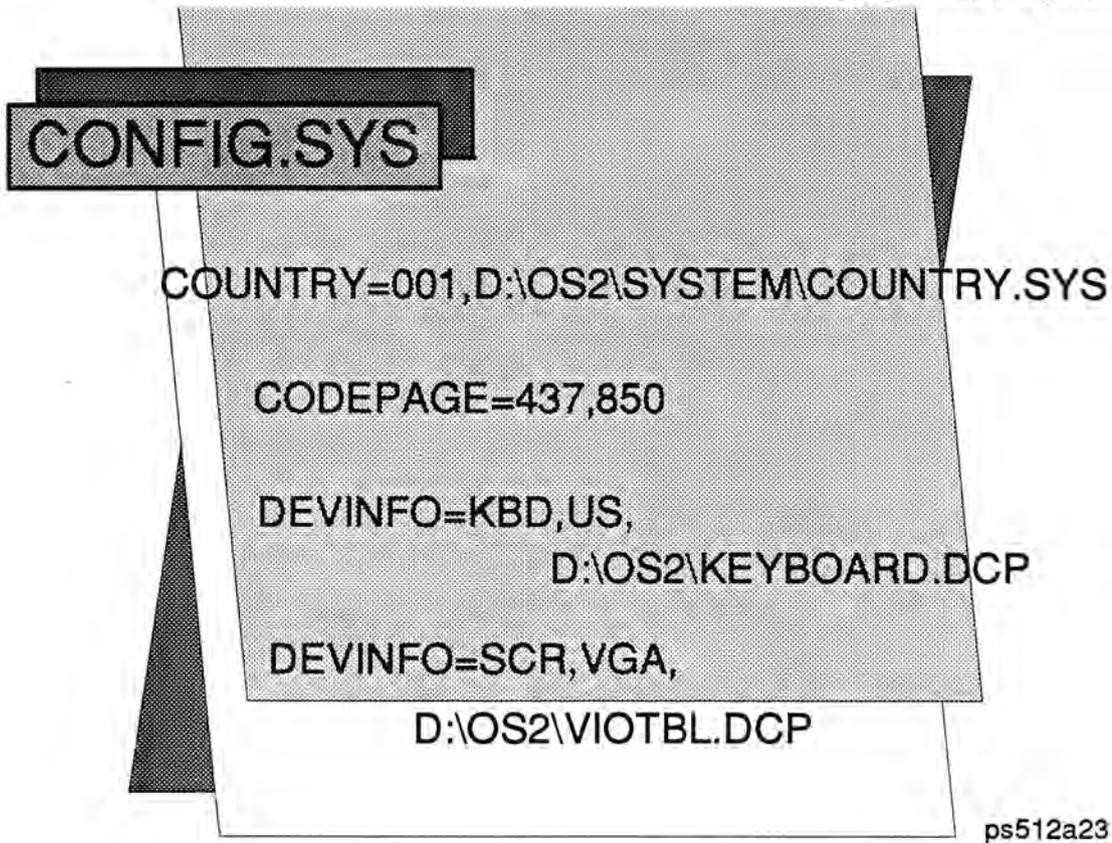


Figure 9-5. OS/2 Parameters

- **COUNTRY =**
Identifies the information such as Date/Time format and Decimal separator. 001, specifies the actual country being supported.
- **CODEPAGE =**
A code page contains letters, numbers, symbols and other characters common to a particular country. Each character has a specific number (1 to 255) assigned to it. Character number 212 would display a character in the US code page (437) that would be different then the same number but in a different code page say 860.
Code pages 437 and 850 are the default code pages for country 001.
- **DEVINFO**
This statement prepares the device for system code page switching.
The display statement specifies your display name and a file names **VIOTBL.DCP** that contains a video font table for displaying characters in each of the code pages supported by the system.

When using a file that was created in another code page, you can switch to that code page or to the multilingual code page. We recommend you use the multilingual

code page (850) whenever possible because it supports many languages. For example, suppose you create a file using code page 850 and send it to someone in another country. When that file is viewed or printed using code page 850, it is identical to your copy. If, however, the file you send was not created using the multilingual code page, the receiver will need to switch to the code page that it was created with. Once code pages are defined on your system, you can switch back and forth between the prepared code pages.

In the OS/2 operating system, a program or user can change the active code page. Two pages can be active simultaneously. Code pages for the keyboard, display, and printer can be set independently; however, code-page switching can take place only in printers and displays that support code-page switching, including the following products:

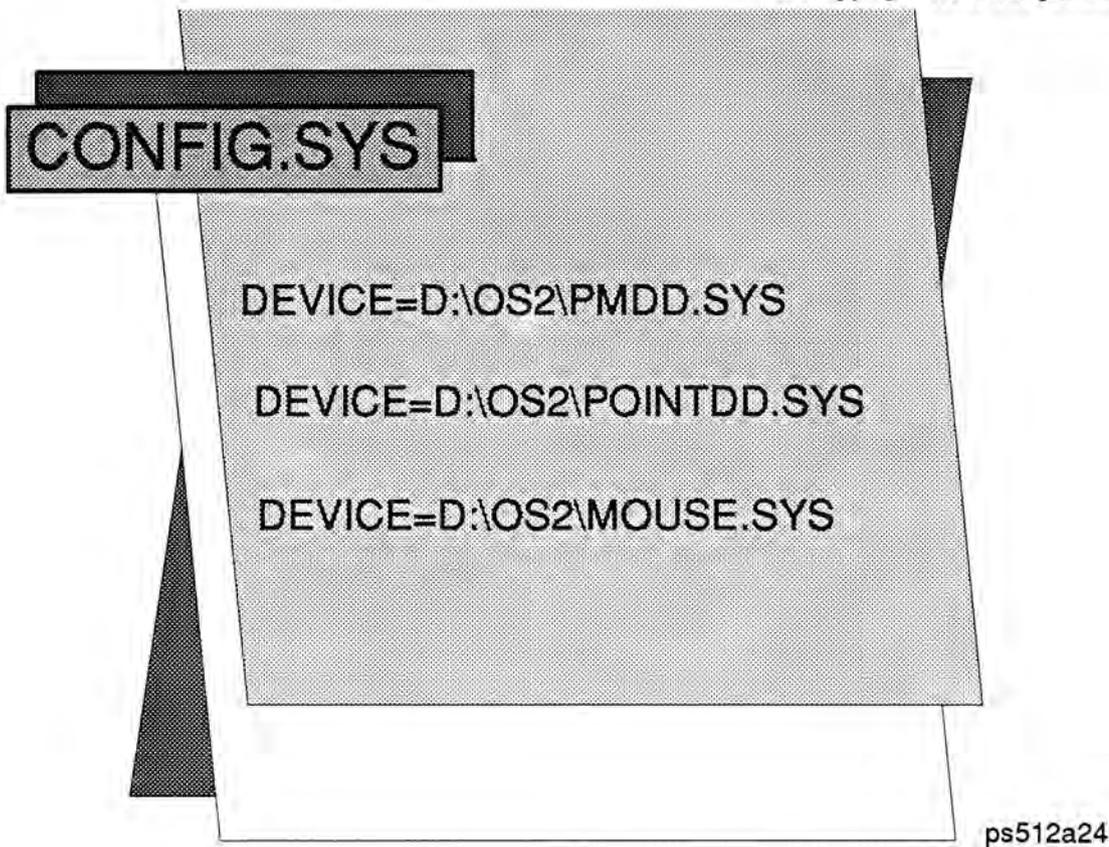


Figure 9-6. OS/2 Parameters

- **DEVICE = D:\OS2\PMDD.SYS**
This device, provides pointer draw support for the OS/2 sessions. If this statement is not in the config.sys file, the system will not restart.
- **DEVICE = D:\OS2\POINTDD.SYS**
Provides mouse pointer draw support. To effectively use a mouse, you must load this device driver in addition to specifying the appropriate mouse device driver statements in the config.sys file.
- **DEVICE = D:\OS2\MOUSE.SYS**
This is a statement that identifies the mouse to **mouse.sys**.

CONFIG.SYS

```
SET PROMPT=$I[$P]  
  
SET VIDEO_DEVICES=VIO_VGA  
  
SET VIO_VGA=DEVICE(BVHVGA)  
  
DEVICE=D:\OS2\COM.SYS
```

ps512a25

Figure 9-7. OS/2 Parameters

- **SET PROMPT =**

This indicates how the command prompt will appear. The default prompt for OS/2 is the current directory of the default drive enclosed within bracket symbols. The default prompt for a DOS session is the default drive letter followed by the 'greater than' symbol.

- **SET VIDEO_DEVICES =**

Is required for graphics support.

- **DEVICE =**

Allows OS/2 applications or system programs, such as SPOOL, to use serial devices. **COM.SYS** system file supports ports COM1 and COM2.

This device driver supports applications with the following RS-232C (serial device) interface functions:

- Duplex communication
- Automatic flow control (XON/XOFF) for both transmit and receive
- Various modem line handshaking modes
- Standard and nonstandard baud rates

CONFIG.SYS

```
SET HELP=D:\OS2\HELP;  
          D:\OS2\HELP\TUTORIAL;  
SET GLOSSARY=D:\OS2\HELP\GLOSS;  
SET BOOKSHELF=D:\OS2\BOOK;  
SET KEYS=ON
```

ps512a26

Figure 9-8. OS/2 Parameters

- **SET HELP =**

Provides a help line as part of the command prompt, displays help information for warning and error messages, or displays a specific topic within a book.

Enter this command without a parameter to display the HELP options available for the current mode of operation. These options allow you to:

- Return to the Desktop.
- Switch to the next session
- Exit the current OS/2 session
- Get additional help on error and warning messages.

If you are writing an application program or working with unfamiliar software, it is advisable to specify HELP OFF, or messages from your program might not appear on the display screen.

If you specify HELP ON, the value of the PROMPT environment is overwritten until you specify HELP OFF. Specifying HELP OFF resets the prompt back to the default system prompts for DOS and OS/2 sessions.

- **SET GLOSSARY =**

Indicates the path to find the glossary information.

- **SET BOOKSHELF =**

Indicates the path to find the on-line command references information.

- **SET KEYS =**

Permits previously issued commands to be retrieved and edited.



CONFIG.SYS

```
PROTSHELL=D:\OS2\PMSHELL.EXE

REM SET DELDIR=C:\DELETE,512;D:\DELETE,512;
      E:\DELETE,512;F:\DELETE,512;

PRIORITY_DISK_IO=YES

PRINTMONBUFSIZE=134,134,134
```

ps512a27

Figure 9-9. OS/2 Parameters

- **PROTSHELL**
This loads the PMSHELL.exe as the user interface program and OS/2 command processor. This, replaces the default OS/2 command processor (cmd.exe) with another command processor.
- **PRINTMONBUFSIZE =**
Sets parallel port device driver buffer size. This statement enables you to increase the size of the parallel port device driver and thereby increase the performance of data to devices connected to the parallel port. There is support for 3 parallel ports. The number positioning, corresponds to LPT1, LPT2, and LPT3 character monitor buffer size.
- **PRIORITY_DISK_IO**
Specifies disk input/output priority for applications running in the foreground. If set to YES, an application running in the foreground will receive disk I/O priority over applications running in the background.
- **SET DELDIR =**
If used, this is the location where the deleted files will actually be held after deletion.

When UNDELETE is specified, if the file is still recoverable, it is reclaimed and restored to its specific path. If a duplicate file name exists, you are prompted to rename it or it is ignored by the system.

UNDELETE can be used in both DOS and OS/2 sessions. Files that are available for recovery are reported as used bytes on the disk.

These are the system installation defaults for the PROTSHELL statement. Should you wish to run another Presentation Manager program say ABCD.EXE then, one would alter the PROTSHELL statement to indicate that ABCD.EXE program.

CONFIG.SYS

```
SET COMPSEC=D:\OS2\CMD.EXE  
DEVICE=D:\OS2\TESTCFG.SYS  
DEVICE= D:\OS2\DOS.SYS  
PROTECTONLY=NO  
SET EPMPATH=D:\OS2\APPS;
```

ps512a28

Figure 9-10. OS/2 Parameters

- **SET COMSPEC =**

If you specify a SHELL statement that contains a file specification other than the default, set the COMSPEC environment variable to the same file specification when a DOS session is started. For example, if you have a command processor named DOSPROC.COM in a DOSPGMS sub-directory, you can place the following SET command in your AUTOEXEC.BAT file: SET COMSPEC = C:\DOSPGMS\DOSPROC.COM

- **PROTECTONLY =**

The OS/2 operating system requires this statement in the CONFIG.SYS file. The PROTECTONLY = YES statement allows memory under 640KB, which is normally used for DOS programs, to be available for OS/2 programs. When PROTECTONLY = YES, you cannot run programs in DOS sessions. If you later decide that you want to run DOS programs in the lower 640KB of memory, specify PROTECTONLY = NO. This allows you to use both DOS and OS/2 programs.

- **SET EPMPATH =**

Is another environment variable, in this case used by the Enhanced Editor.

- **DEVICE = D:\OS2\TESTCFG.SYS**

This device driver is used during the install process to test the system configuration. It should not be removed from this file as it is also used by the selective install process and during device driver installation.

CONFIG.SYS

```
BASEDEV = PRINT02.SYS  
BASEDEV = IBM2FLPY.ADD  
BASEDEV=IBM2SCSI.ADD  
BASEDEV=OS2SCSI.DMD  
BASEDEV=IBMINT13.I13  
BASEDEV=OS2DASD.DMD
```

ps512a29

Figure 9-11. OS/2 Parameters

Installs a base device driver by specifying the complete file name of the device driver in your CONFIG.SYS file. The above is an indication of some of the possible base drivers that get installed.

The BASEDEV statement is used to load base device drivers. Device support for disks, diskettes, printers connected to the workstation, and other devices, is loaded with the BASEDEV statement.

The following base device drivers are included with your OS/2 diskettes:

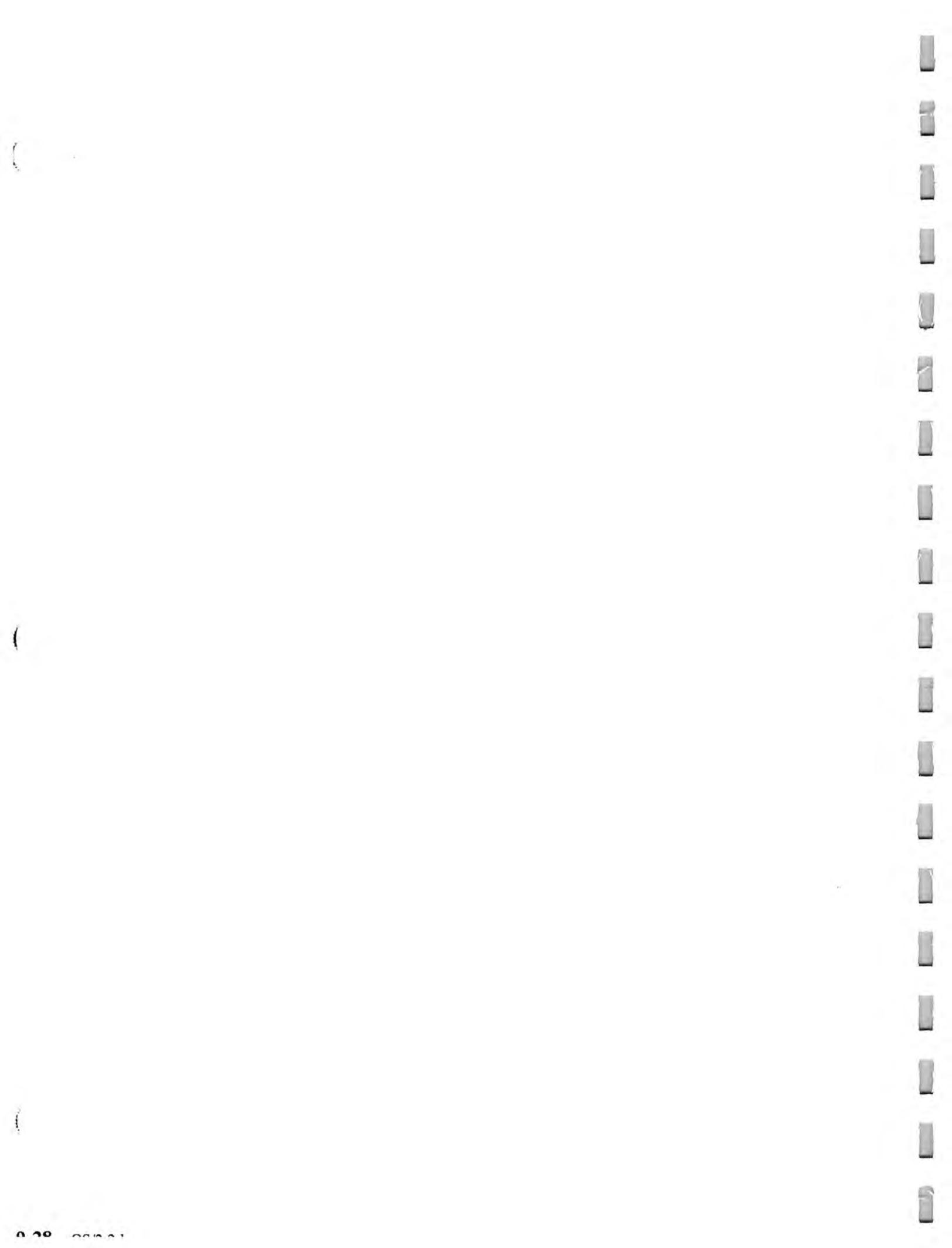
- PRINT01.SYS Device support for locally attached printers on non-Micro Channel workstations.
- PRINT02.SYS Device support for locally attached printers on Micro Channel* workstations.
- IBM1FLPY.ADD Device support for diskette drives on non-Micro Channel workstations.
- IBM2FLPY.ADD Device support for diskette drives on Micro Channel workstations.
- IBM1S506.ADD Device support for non-SCSI disk drives on non-Micro Channel workstations.

- IBM2ADSK.ADD Device support for non-SCSI disk drives on Micro Channel workstations.
- IBM2SCSI.ADD Device support for Micro Channel SCSI adapters.
- IBMINT13.I13 General-purpose device support for non-Micro Channel SCSI adapters.
- OS2DASD.DMD General-purpose device support for disk drives.
- OS2SCSI.DMD General-purpose device support for non-disk SCSI devices.

Subtopic Summary

In this subtopic the student were introduced to the remainder OS/2 configuration parameters that had not been previously discussed.

This concludes this subtopic of the topic
"OS/2 Configuration file".



Previously discussed OS/2 configuration file parameters

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to explain the purposes of the parameters discussed in previous topics, that pertain to the OS/2 configuration file.

Enabling objectives:

After attending this subtopic the student should be able to explain the

- DOS parameters
- Windows parameters
- File System parameters
- Memory Management parameters

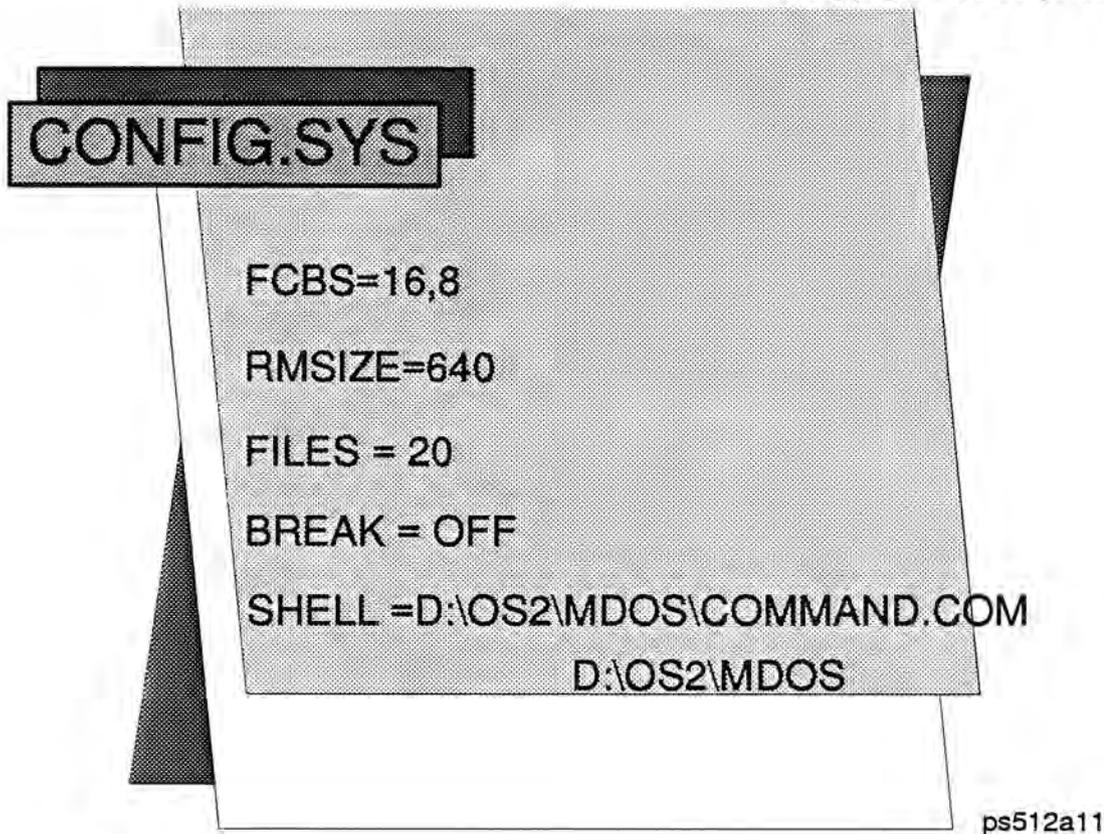


Figure 9-12. DOS Parameters

- **FCBS**

Determines file-control-block information for DOS sessions.

A File Control Block is a record that contains all of the information regarding a file (for example, its structure, length, and name).

Some programs use file control blocks to create, open, delete, read, and write to files. New programs written for the operating system use internal file IDs (handles) for file input/output.

- **RMSIZE**

This statement specifies the highest memory allowed for a DOS session. If you specify a number that exceeds the amount of memory allowable for our system configuration, the system will give you an error message. In such a case, the system will indicate that the value is not acceptable and it will ignore your statement.

- **BREAK**

Instructs DOS to check whether the CTRL and BREAK keys have been pressed before carrying out a program request.

- **FILES**

Determines the maximum number of files available in DOS sessions. A file is in use when a program is processing some kind of operation in it. When a program is using a file for its operation, that file is unavailable to another program. The file is returned to availability when the program has finished its operation and the file is closed.

Placing a FILES = statement in the CONFIG.SYS file increases the default value for all DOS sessions. Each session can also be customized by changing the appropriate DOS setting.

- **SHELL = D:\OS2\MDOS\COMMAND.COM D:\OS2\MDOS**

This loads and starts the DOS command processor **command.com**. It also permits you to replace this processor with another. If this statement is omitted from the config.sys file, the default DOS command processor is loaded and started with a /p parameter to retain COMMAND.COM in memory.

CONFIG.SYS

```
DEVICE=D:\OS2\MDOS\VMOUSE.SYS
DEVICE=D:\OS2\MDOS\VVGA.SYS
DEVICE=D:\OS2\MDOS\VCOM.SYS
DEVICE=D:\OS2\MDOS\VDPMI.SYS
DEVICE=D:\OS2\MDOS\VWIN.SYS
```

ps512a12

Figure 9-13. DOS Parameters

- **DEVICE = D:\OS2\MDOS\VMOUSE.SYS**
Virtual mouse support
- **DEVICE = D:\OS2\MDOS\VVGA.SYS**
Virtual video support
- **DEVICE = D:\OS2\MDOS\VCOM.SYS**
Virtual asynchronous communications port support
Recall, that virtual device drivers maintain the hardware state for each VDM and prevent an application in one VDM from corrupting another VDM.
- **DEVICE = D:\OS2\MDOS\VDPMI.SYS**
Virtual DOS Protect Mode Interface
- **DEVICE = D:\OS2\MDOS\VWIN.SYS**
Device driver required for the seamless Windows applications capability

CONFIG.SYS

```
DEVICE=D:\OS2\MDOS\VEMM.SYS
DOS=LOW,NOUMB

DEVICE=D:\OS2\MDOS\VDPX.SYS
DEVICE=D:\OS2\MDOS\VXMS.SYS /UMB

BUFFERS= 30
```

ps512a13

Figure 9-14. DOS Parameters

- **DEVICE = D:\OS2\MDOS\VEMM.SYS**
Virtual device driver which provides the LIM V4.0 Expanded memory to DOS sessions.
- **DOS = LOW,NOUMB**
Specifies that the DOS kernel should reside in conventional (low) memory below 640KB. This means that DOS applications can use the high memory area. The high memory area (HMA) refers to space between 1MB and 1MB + 64KB.
"NOUMB" specifies that the operating system does not control the upper memory blocks (UMBs). This means that DOS applications can allocate UMBs but cannot be loaded there.
- **DEVICE = D:\OS2\MDOS\VDPX.SYS**
Virtual DOS Protect Mode Extender driver support
- **DEVICE = D:\OS2\MDOS\VXMS.SYS /UMB**
VXMS.SYS is a virtual device driver that provides Extended Memory Specification (XMS) emulation to DOS sessions. XMS allows DOS applications to access more than 1MB of memory, under XMS control.

- **BUFFERS = 30**

This sets the number of disk buffers that the system uses. Buffering is one memory management method used by the system. by using the disk buffer, the operating system can read and write blocks of information. Once the information is read into the buffer, the system is ready to process the information. While the information is in the buffer or being processed, the input device can begin reading in new information so that the processor need not wait unnecessarily to process the information.

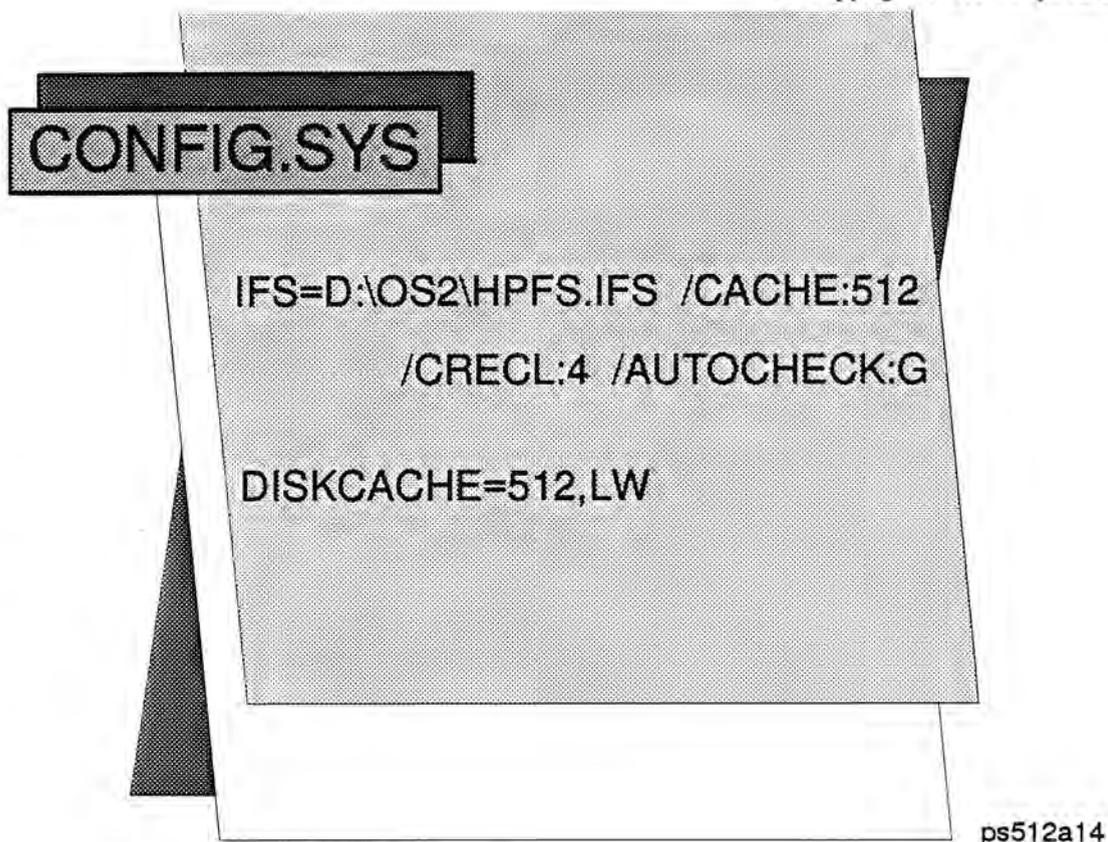


Figure 9-15. Parameters

- **IFS =**
Notice the appearance of **autocheck**. The *format* command places this parameter on that statement. Upon powering on your machine, the system will perform an autocheck to ensure that any HPFS partitions were properly 'shut down' prior to the last powering down of your system.
- **DISKCACHE =**
Specifies the number of blocks of memory to allocate to the FAT file system cache.

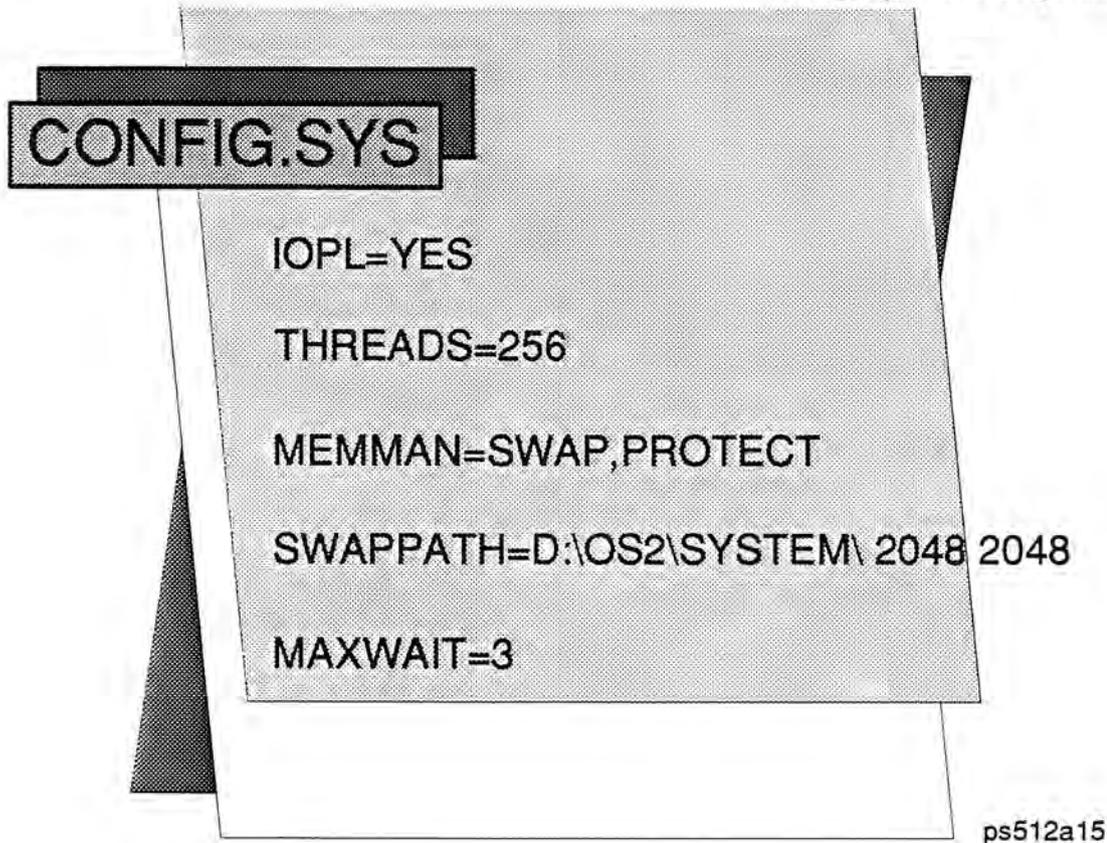


Figure 9-16. Parameters

- **IOPL =**
Allows I/O privilege to be granted to requesting processes in OS/2 sessions.
- **THREADS =**
This specifies the maximum number of threads available to OS/2 and its applications.
- **MEMMAN =**
This will enable **swapping** and it will also enable APIs to allocate and use protected memory by DLLs.
- **SWAPPATH =**
This points to the location of the swapper file. The initial value is the **minfree** value. This value denotes the amount of disk space at which the system will warn you that there is less than this amount of space left on the partition containing the swapper file.
- **MAXWAIT =**
This sets the length of time, in seconds, a thread waits before the system assigns it a higher priority.

CONFIG.SYS

```
LIBPATH=.;D:\OS2\DLL;D:\OS2\MDOS;  
D:\;D:\OS2\APPS\DLL;
```

ps512a16

Figure 9-17. Parameters

This parameter is used to identify a set of directories to be searched when OS/2 loads dynamic link libraries.

9 Topic Summary

In this subtopic the student were reminded of the config.sys parameters that have been previously discussed.

This concludes another subtopic of the topic "OS/2 Configuration file".

Device driver
LOAD order
irregardless of order in config.sys

- 1 SYS
- 2 B.ID
- 3 VSD
- 4 TSD
- 5 ADD
- 6 TIB
- 7 FLD
- 8 DmD

Recovery

subtopic objectives

Terminal objective:

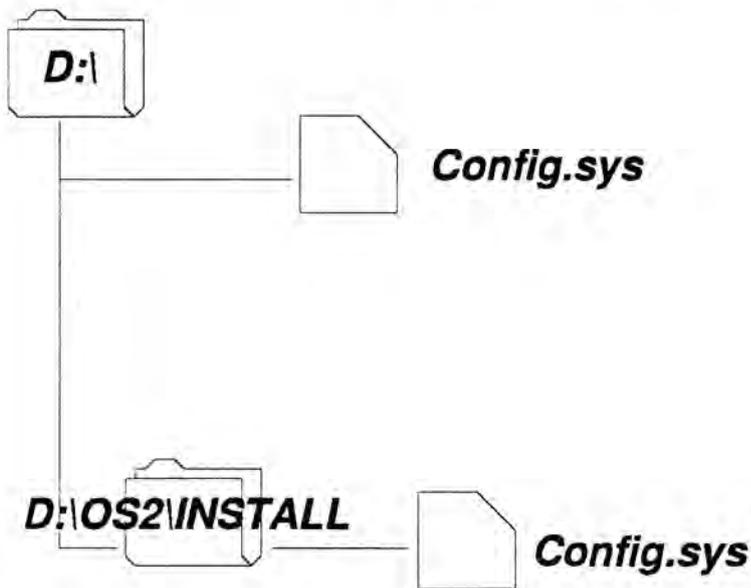
After attending this subtopic the student should be able to explain how to backup the Desktop and recover a config.sys file.

Enabling objectives:

After attending this subtopic the student should be able to explain

- how to recover a config.sys file
- how to recover from hard drive errors
- how to backup the Desktop

Recovery of - Config.sys



ps512b21

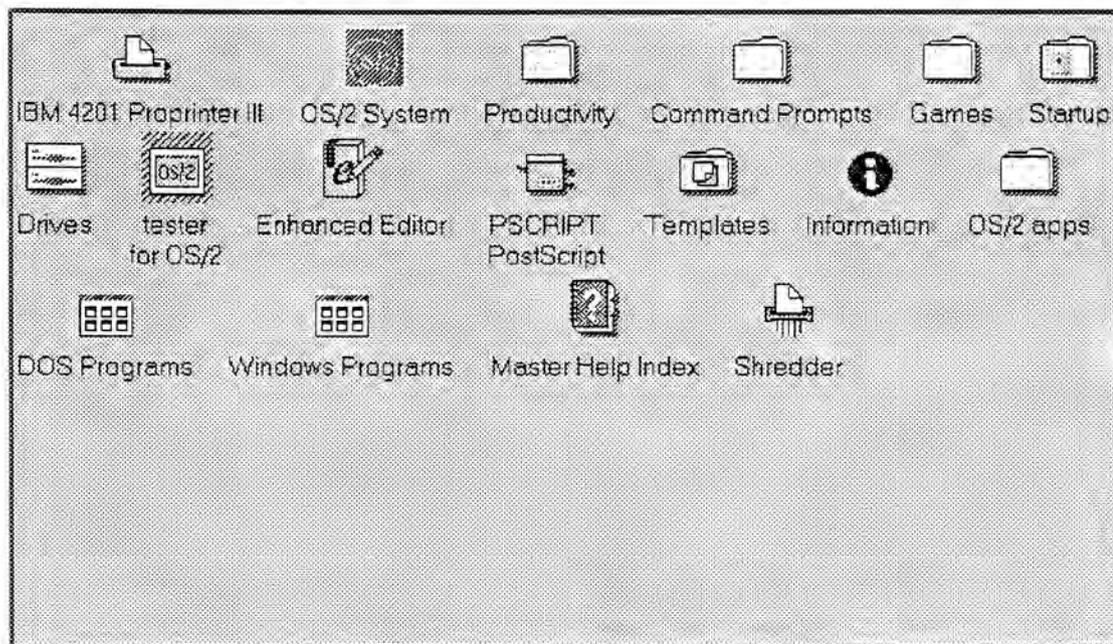
Figure 9-18. Recovery of Config.sys

The CONFIG.SYS file contains command statements that are used to configure your system upon startup. If the file is changed incorrectly, you might not be able to restart the system or edit the file. For example, some programs write information to this file when they are installed. In some cases, this information can cause the CONFIG.SYS file to be unusable.

If your CONFIG.SYS file becomes unusable you can either

- Make use of your backup CONFIG.BAK. This is a file that sometime earlier, you would have created as a backup of your good working copy of CONFIG.SYS.
- Use the CONFIG.SYS file that the system created for you when you initially installed the operating system. Keep in mind that you will probably have to edit that file to include any changes you had made through time on your system (for instance installing a device driver).

If your CONFIG.SYS file has been modified such that you can not even start the system, then you will have to boot with the *OS/2 Installation Diskette* and then use *Diskette 1* to get to a *A* command prompt.



ps512b23

Figure 9-20. Backup of the Desktop

Once your Desktop is setup the way you want it, you can copy the setup to a diskette and use it as a backup. You can also then copy your setup to another workstation.

Once again, one must boot from the **Installation Diskette** and get to a command prompt through *Diskette 1*. The XCOPY command can then be used.

- `XCOPY D:\Desktop A:\DESKTOP /S /E`

An extended attribute is a special area for data that describes the the file or directory to the base operating system or to an application. XCOPY will copy extended attributes of a source file to the target file.

Topic Summary

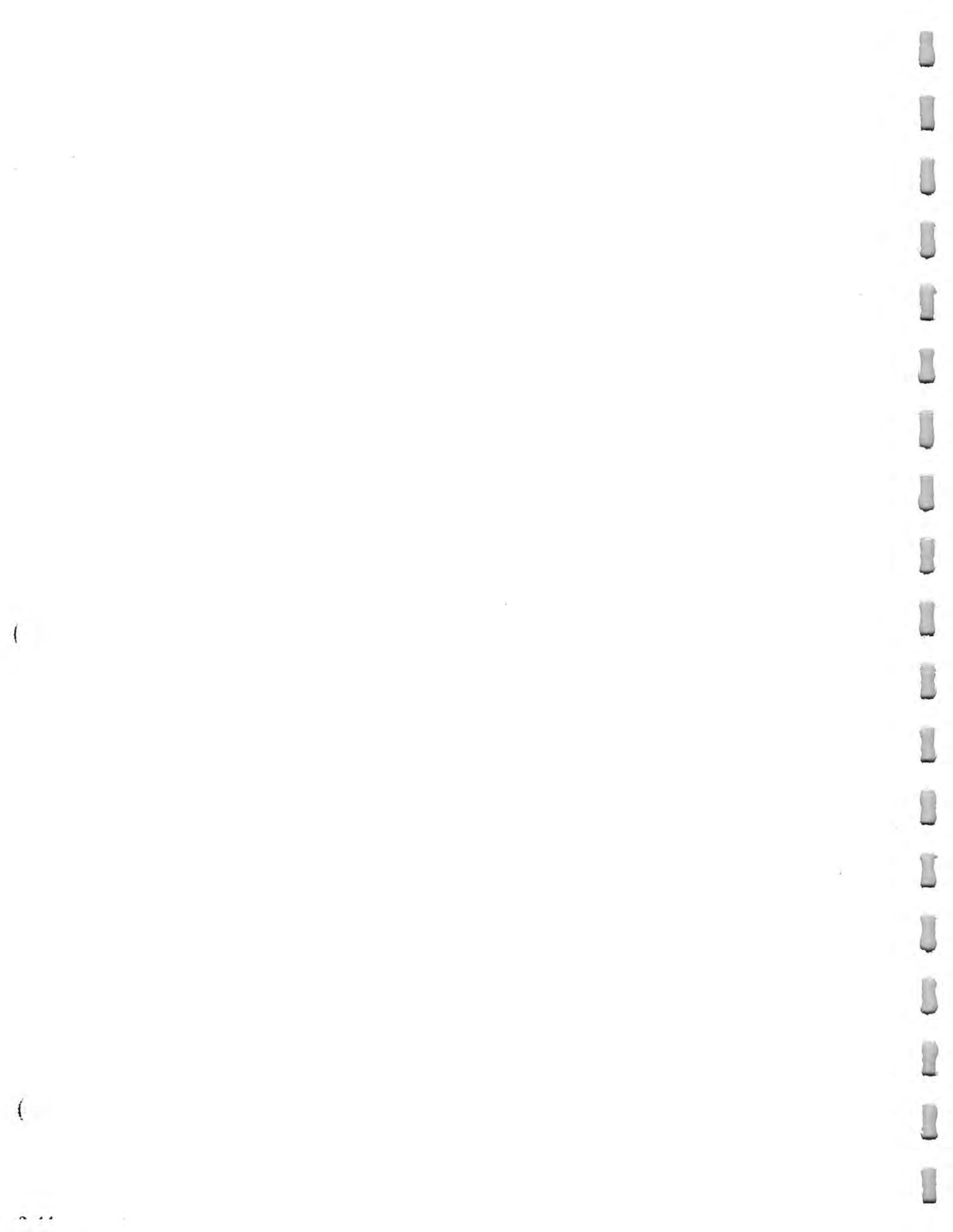
In this subtopic the student were introduced to the available methods for recovering their Desktop and their config.sys file.

This concludes this subtopic of the topic
"OS/2 Configuration file".

Topic Summary

In this topic the student learned what the various parameters in the OS/2 configuration file represent, and which ones can be altered.

This concludes the topic
"OS/2 Configuration file".



TOPIC 10: OS/2 Version 2.1 Printing

Topic objectives

Terminal objective:

After attending this topic the student should understand and be able to utilize OS/2 Version 2.x's object oriented approach to printing.

Enabling objectives:

Upon completion of this topic the student should be able to customize print output by manipulating the following components of the OS/2 Version 2.x print subsystem:

- Printer objects
- Printer driver objects
- Port, job, spooler and queue
- Win-OS/2 Printing



Printer Object

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to customize print output by manipulating the printer object component of the OS/2 Version 2.1 print subsystem.

Enabling objectives:

After attending this subtopic the student should be able to:

- Create a printer object
- Customize the printer object's Settings notebook.

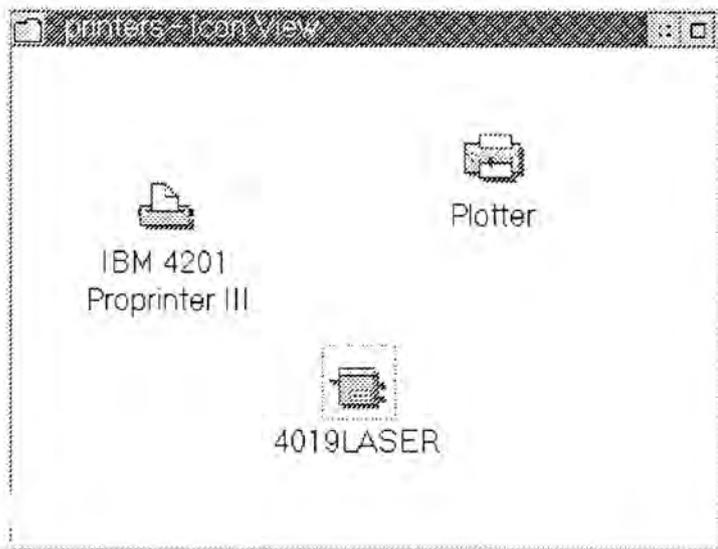


Figure 10-1. Printer Object Icons

The OS/2 user interface consists of a total of six types of objects accessible from the desktop. Each of these objects represents a real functioning component of the print subsystem which collectively interact in various ways to produce the desired output.

Use of this interface is simplified by having just one type of object visible on the desktop through which printing configuration and management can be directed.

This object is called the **Printer Object**. It allows you access to the other objects associated with printing such as:

- Printer driver object
- Port object
- Job object
- Spooler object
- Queue driver object.

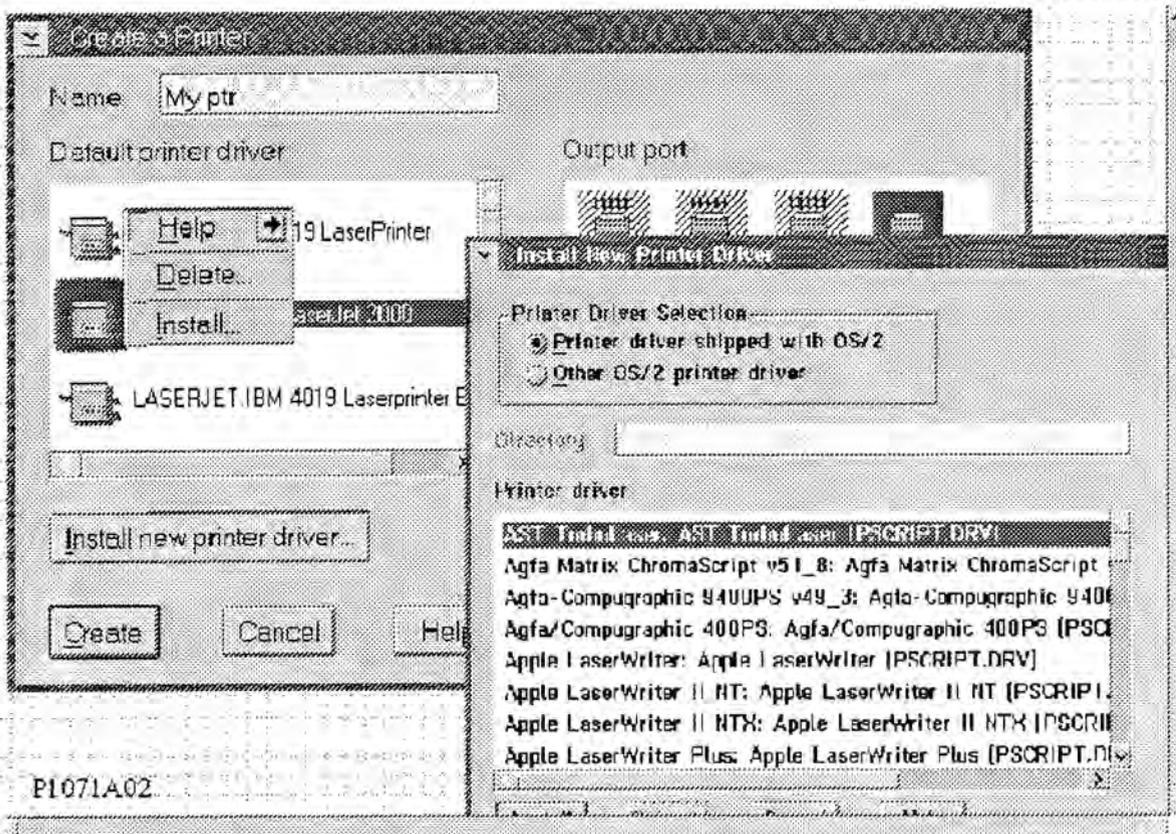


Figure 10-2. From the Templates Folder

To create a printer object from the templates folder:

1. Double click on the *Templates* folder.
2. Drag the *Printer* template to a folder or to some other available place on your desktop.

This action brings up the *Create a Printer* dialog.

3. Type a unique name into the *Name* field.
4. Double click on a printer driver from the **Default printer driver** container.

If the printer driver you want is not yet installed, you can install it at this point by opening the Settings notebook of any installed printer driver object shown in the Default printer driver container.

5. Double click on an output port in the *Output port* container.
6. Double click on *Create*.

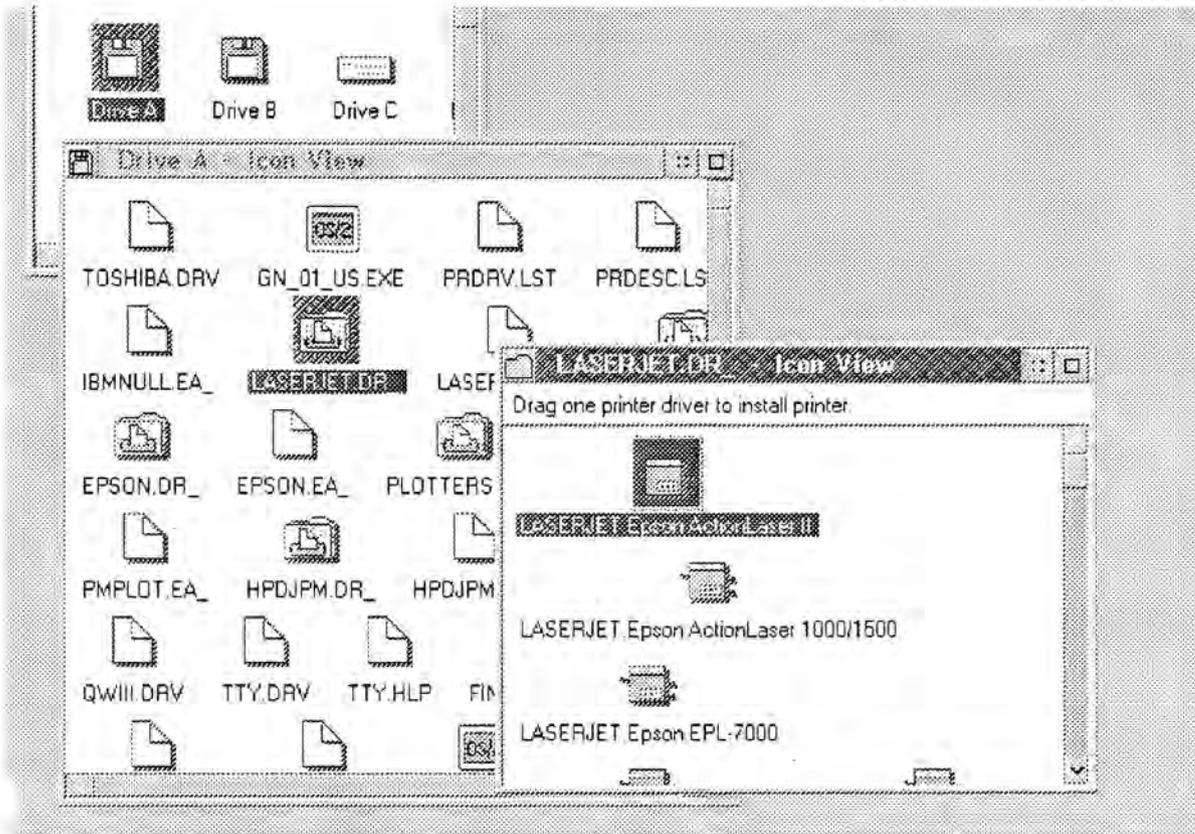


Figure 10-3. From a Device Driver Diskette

This method has the advantage of not only creating a new printer object on the desktop, it will also automatically install the printer device driver object.

However, in order to accomplish this you need to know which of the OS/2 device driver diskettes contains the driver for the printer you wish to install. When you know which diskette contains the file:

1. Insert the desired OS/2 device driver diskette into Drive A.
2. Double click on Drive A.
3. Double click on the appropriate printer driver.
4. Drag the appropriate printer driver icon from the Drive A window to a folder or to some other available place on your desktop.

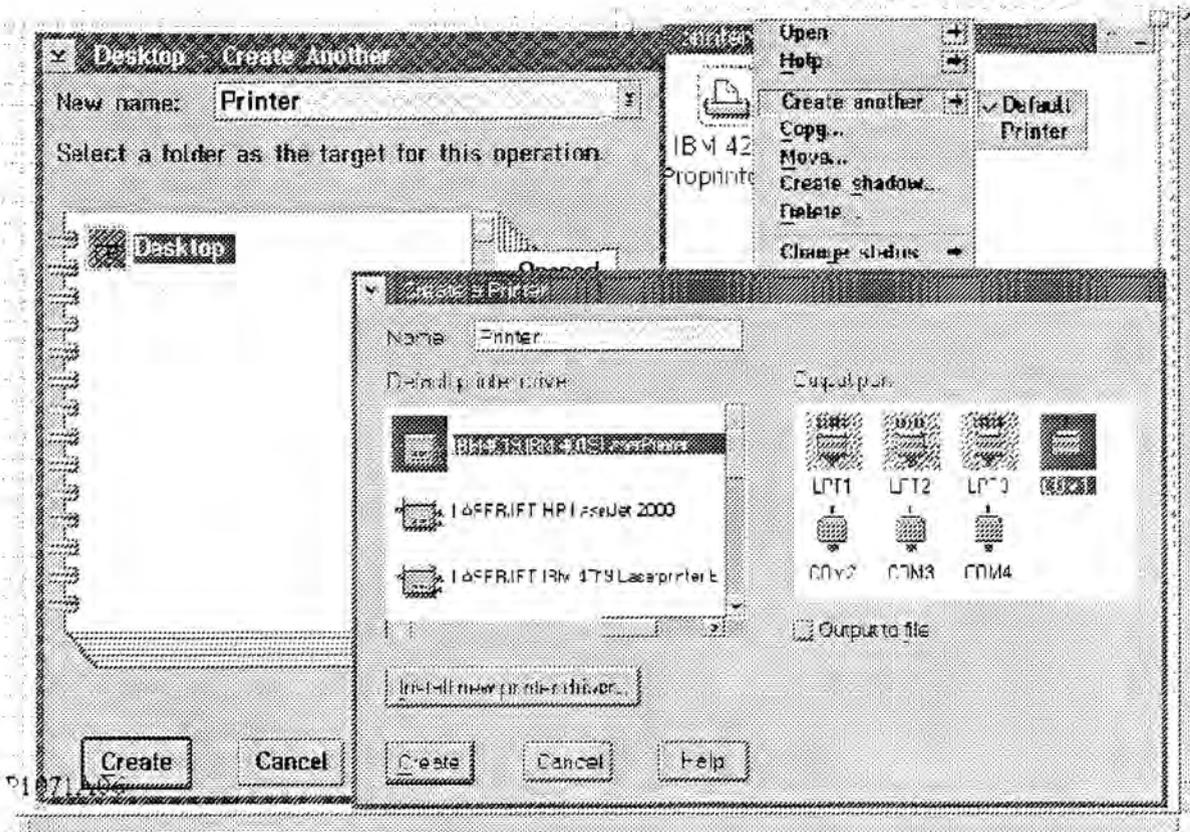


Figure 10-4. From an Existing Printer Object

To use this method, you must have previously installed a printer object.

1. Display the context menu for this printer object.
2. Click on *Create another*.

This brings up the *Create Another* dialog.

3. Optionally, change the name in the *New name* field.
4. Select a folder as the target in which to create this print object.
5. Click on *Create*.

This will bring up the *Create a Printer* dialog.

6. Change the name in the *Name* field as this is the name that will appear together with the icon in your system.
7. Select an *Output port* or click the *Output to file* box.
8. Select a *Default printer driver*.

If the printer driver you want is not yet installed, you can install it from here.

9. Click on *Create*.

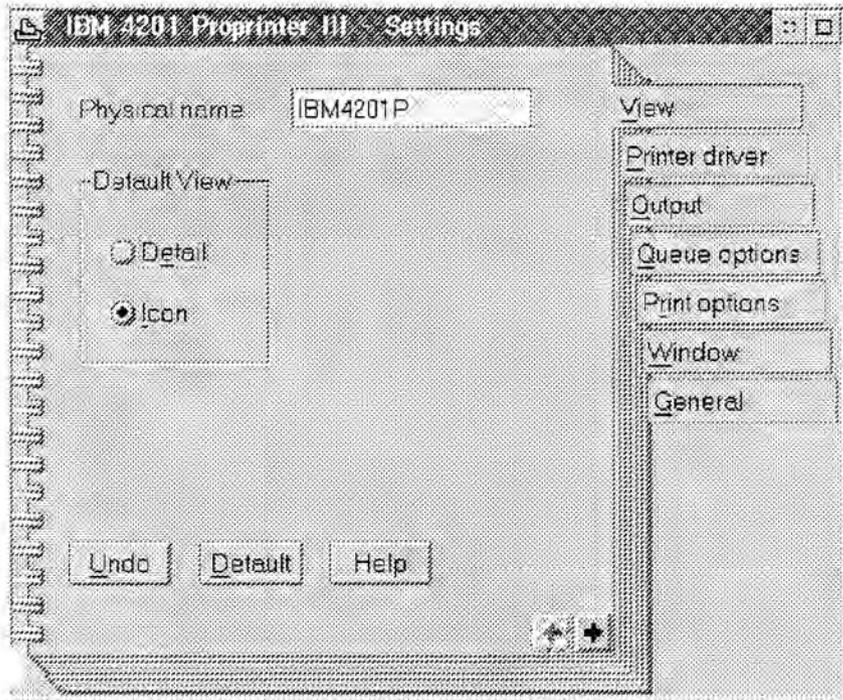


Figure 10-5. View

The View page shows you the system's **physical** name, i.e. the internal name of the object's queue and the name of the queue folder (directory) in the spooler folder (directory).

This name is entered at creation time and cannot be changed in these settings. It is limited to eight characters to accommodate applications that do not use the longer names available in OS/2. While some applications use this name to identify the printer, others use the *Title* name in the *General* tab.

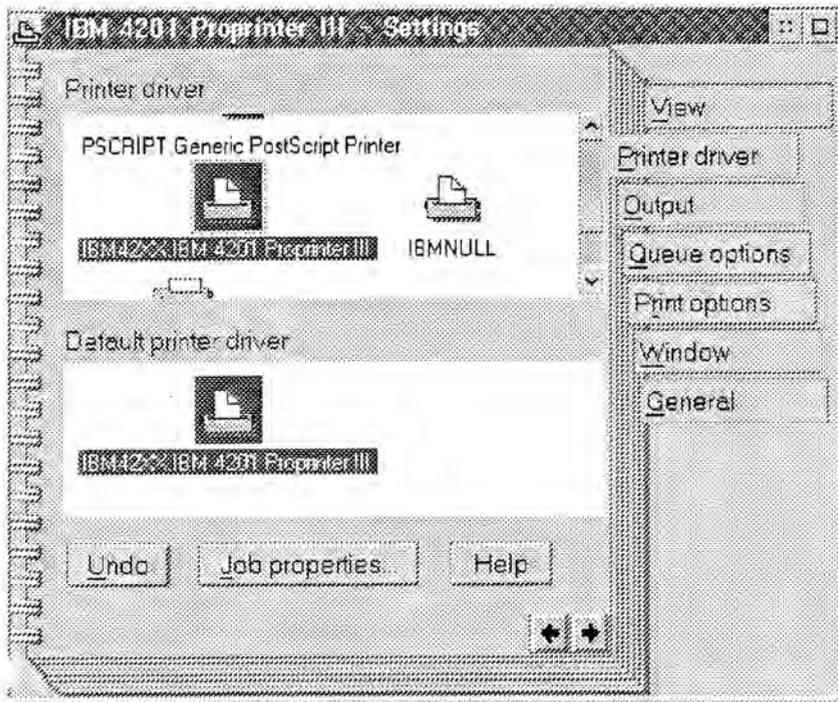


Figure 10-6. Printer driver

The printer driver page shows the installed *Printer drivers* and the *Default printer driver* used by this print object. You can change your selections or make a different driver the default driver. Click on the *Job properties* pushbutton to set up default job requirements unique to this object, such as paper size or resolution. If you change the physical setup of your printer, such as the size of the paper in the printer, you can install additional printer objects for the same physical printer but with different job properties.

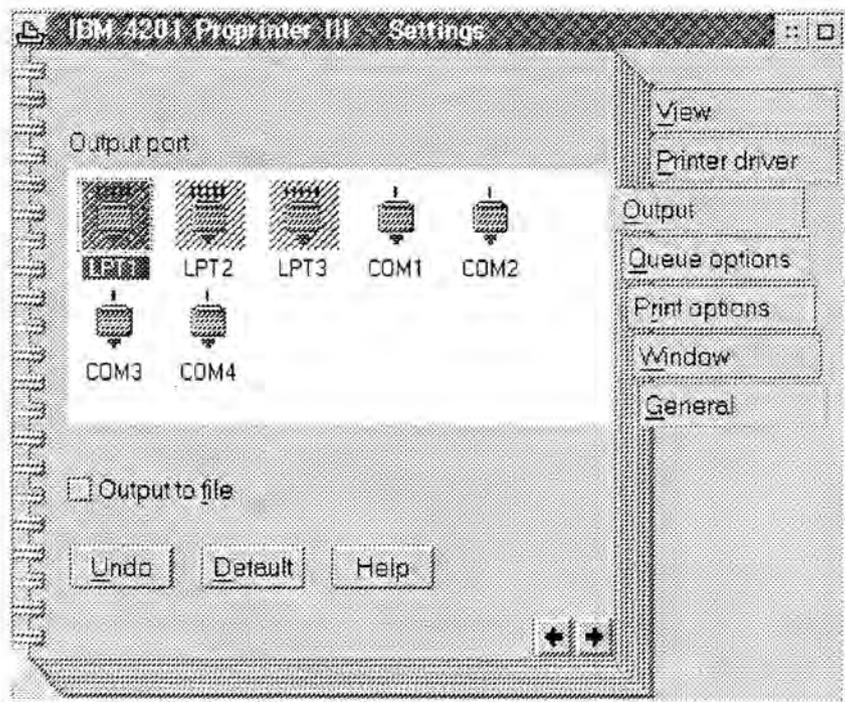


Figure 10-7. Output

The output page shows the port(s) selected and the port(s) available for this printer object. You can select a different port or more than one port to route print-jobs. Instead of a port, you can set up a printer object that saves your print-job as a printable file.

If you desire to have no port or "NONE" associated with this printer object, then simply click on any space inside the *Output port* container but outside of the port objects. You can also redirect LPT output to a COM port, however, you cannot redirect output from a COM port to an LPT port.

The redirection option is only available when two or more printer objects are defined.

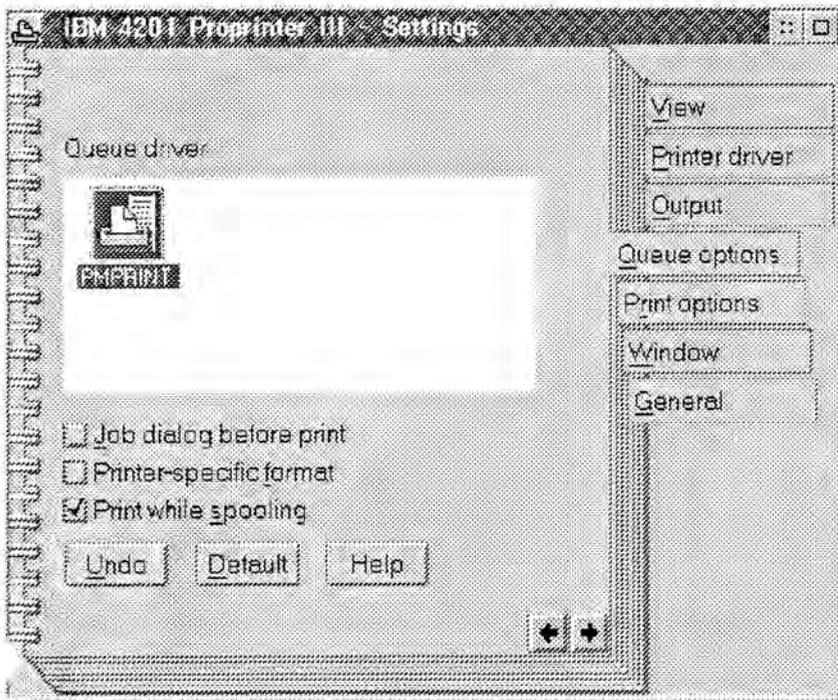


Figure 10-8. Queue options

Use this window to choose a *Queue driver* for this particular printer object.

The *Job dialog before print* option will display the printer object's job properties dialog *only* if you initiate printing by dragging and dropping a file on the printer object.

When you print from applications, most of them allow you access to these properties via options like "Printer setup."

You also use this window to indicate that you want to print jobs in *Printer-specific format* (PM_Q_RAW).

The *Print while spooling* option allows the printer to start processing the print job before the application has finished sending the entire job to the spool queue. This "threading" will increase through-put but could cause timeout problems while printing large files with images.

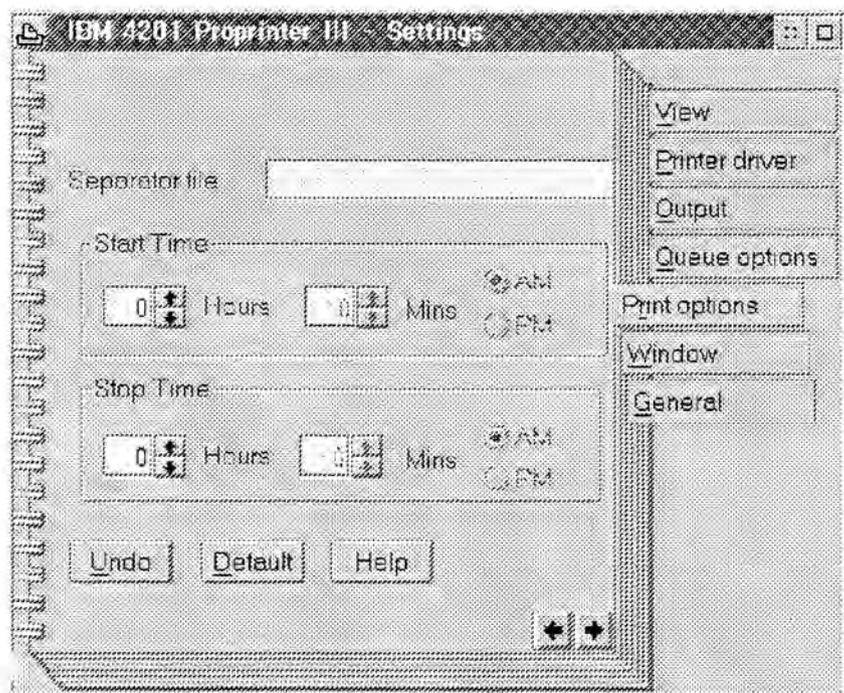


Figure 10-9. Print options

You can use the *Separator file* on this page to specify a separator page file. Separator pages are most common in a network environment. Some printers, such as the IBM 4029, have pre-defined separator pages on utilities diskettes.

Start Time and *Stop Time* can be entered for each print object. This feature allows the workstation to run as fast as possible while the user is there and still be productive when the user is not there.

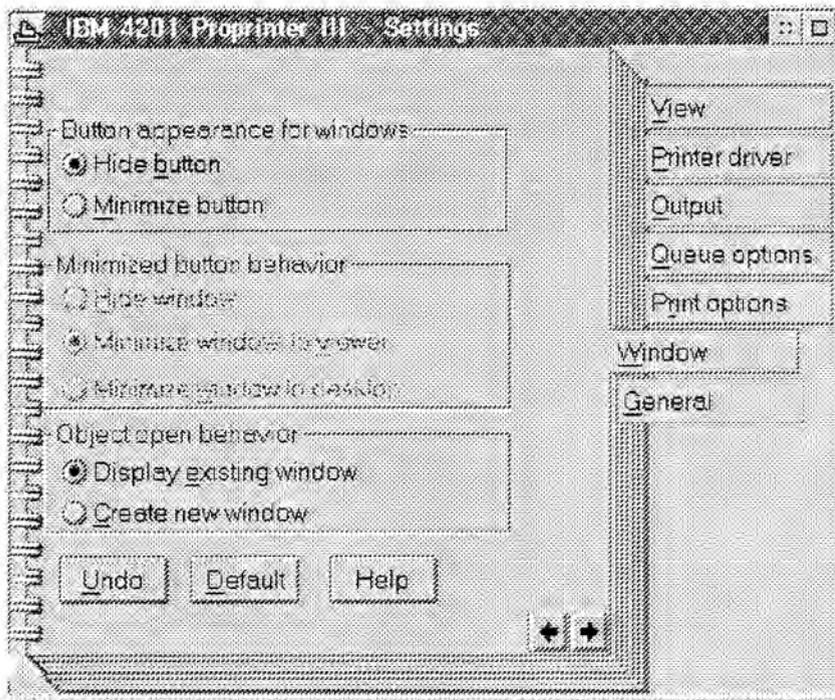


Figure 10-10. Window

The window page lets you tell the system how you want the printer object to act when you minimize it. If you click on the *Minimize button*, then you can select an entry in the *Minimize button behavior* group. This selection will allow you to specify that the minimized printer object window be placed on the desktop, in the *Minimized Window Viewer* or you could select *Hide window*.

Although you can change the *Object open behavior*, it will have no effect on printer objects as they will always use *Display existing window*.

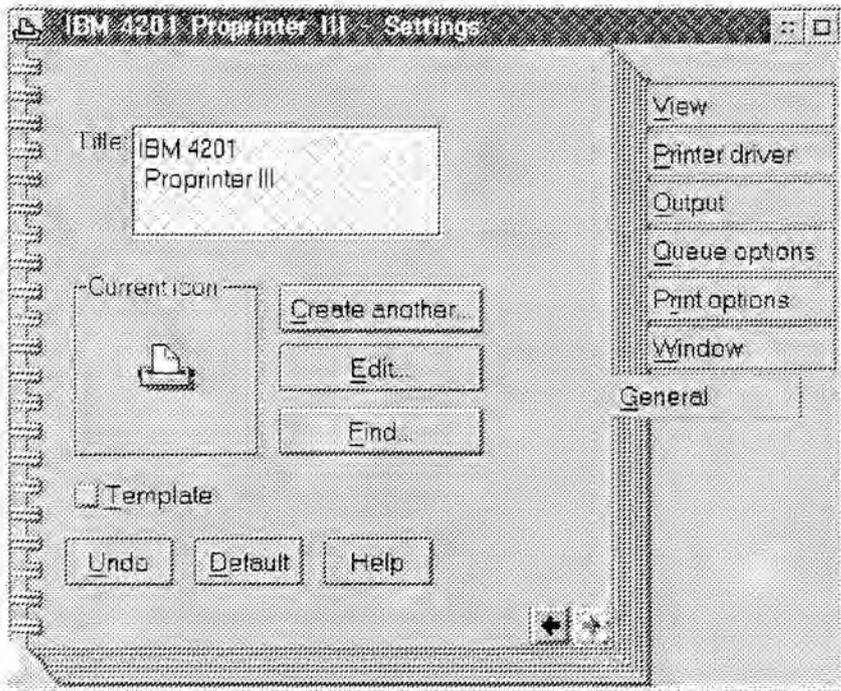


Figure 10-11. General

You can change the name of the selected object in *Title:*. You can change the icon that represents this object by using the *Create another...*, *Edit...*, or the *Find...* options.

You can click on the *Template* box to use the selected object as a template to create new printer objects with the same settings.

Subtopic Summary

In this subtopic the student learned how to customize print output by manipulating the printer object component of the OS/2 Version 2.x print subsystem.

The student was taught how to:

- Create a printer object
- Customize the printer object's Settings notebook.

This concludes the subtopic of the topic
"OS/2 Version 2.x Printing".



Printer Driver Object

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to customize print output by manipulating printer drivers within the OS/2 Version 2.x print subsystem.

Enabling objectives:

After attending this subtopic the student should be able to:

- Install a printer driver
- Access the printer properties dialog of a printer driver.

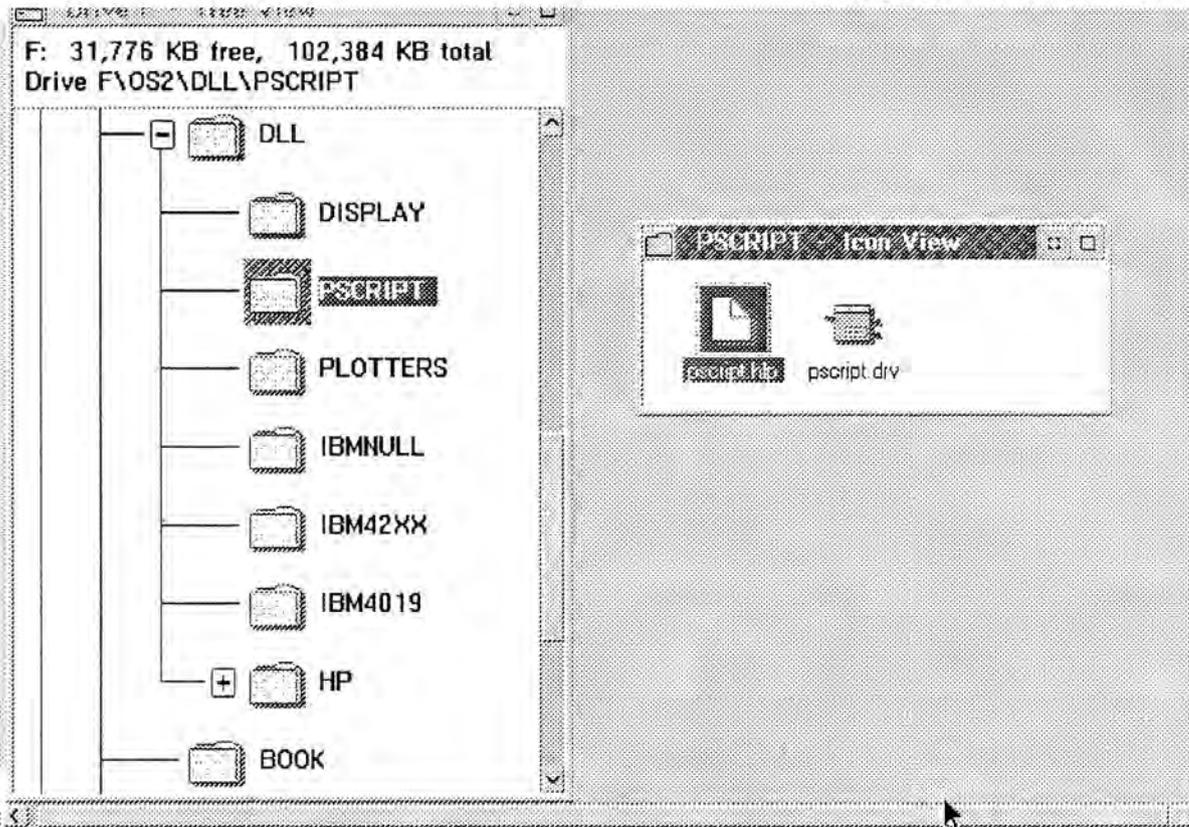


Figure 10-12. Printer Drivers

Each printer connected to your system requires at least one printer driver. Printer drivers provide information that enables OS/2 to create a data stream appropriate for the particular printer model you select for that object.

The printer driver object represents the file(s) that comprise your OS/2 printer driver.

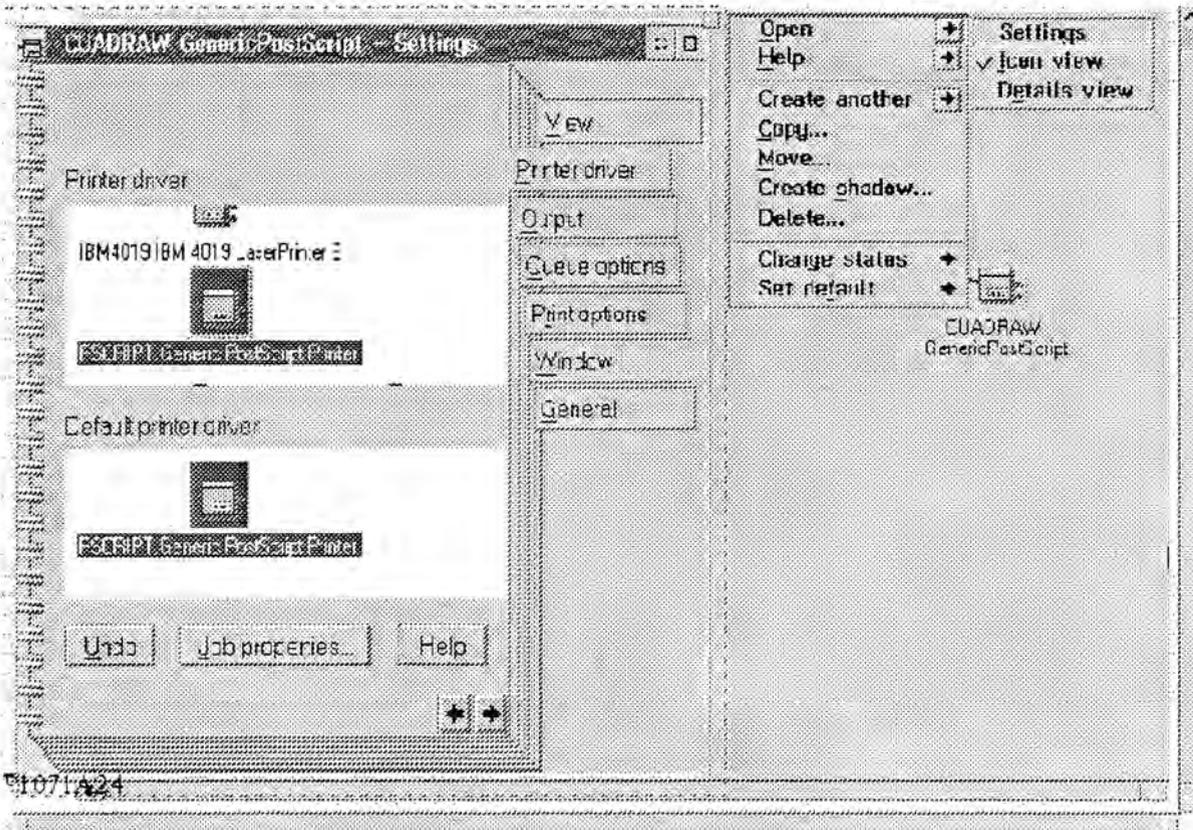


Figure 10-13. During Installation of OS/2

The only print object that exists on the Workplace Shell is the **Printer object**. The printer **device driver object** exists "inside" of the defined Printer object(s). There are three ways to install a printer device driver in OS/2.

1. During the installation of OS/2.
2. While creating a Printer object
3. From an existing Printer object.

When you install OS/2 Version 2.x, you are prompted to install a printer that will be used as the default. If you do so, the system will create a printer object and automatically install the device driver for that printer. If you do not make a selection, no printer object is created on the desktop but the IBMNULL printer driver is added to your system.

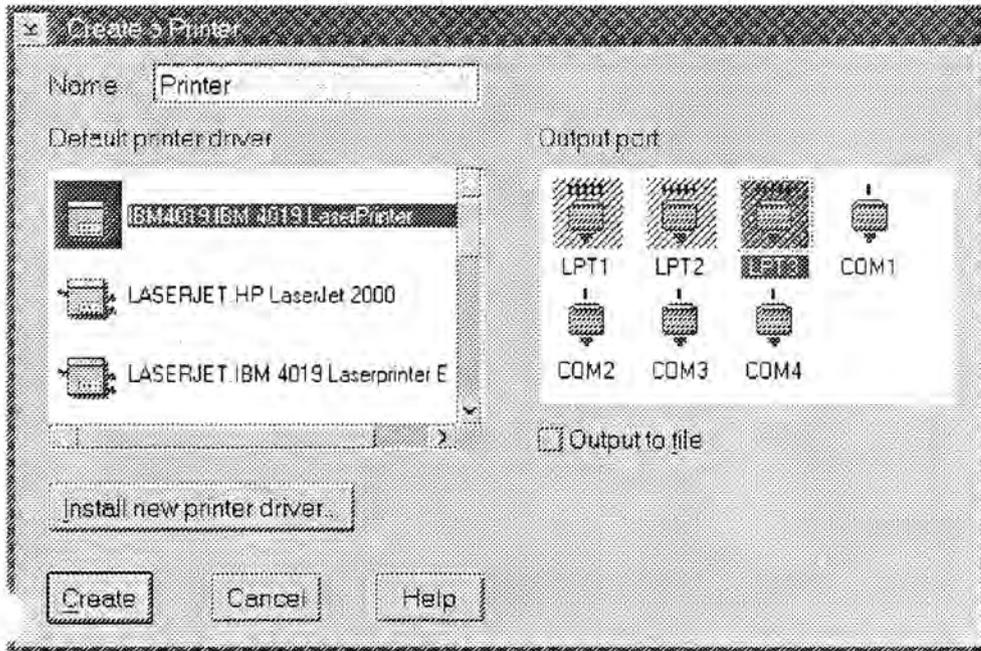


Figure 10-14. While Creating a Printer Object

Previously, we outlined three different ways to create a Printer object.

1. From a template
2. From a device driver file
3. From an existing printer object.

If you create a Printer object from a device driver file then the device driver selected is automatically installed and becomes the default device driver for that Printer object.

If you create a Printer object from a template or an existing Printer object then you will be presented with a Create a Printer dialog box with a Default printer driver container. Remember, the drivers in the *Default printer driver* container have already been added to your system. You are simply assigning them as the default printer device driver for the printer object that you are presently creating.

If the printer device driver you desire is not present on your system, you will need to add it to your system (install it) from diskette. See the next page.

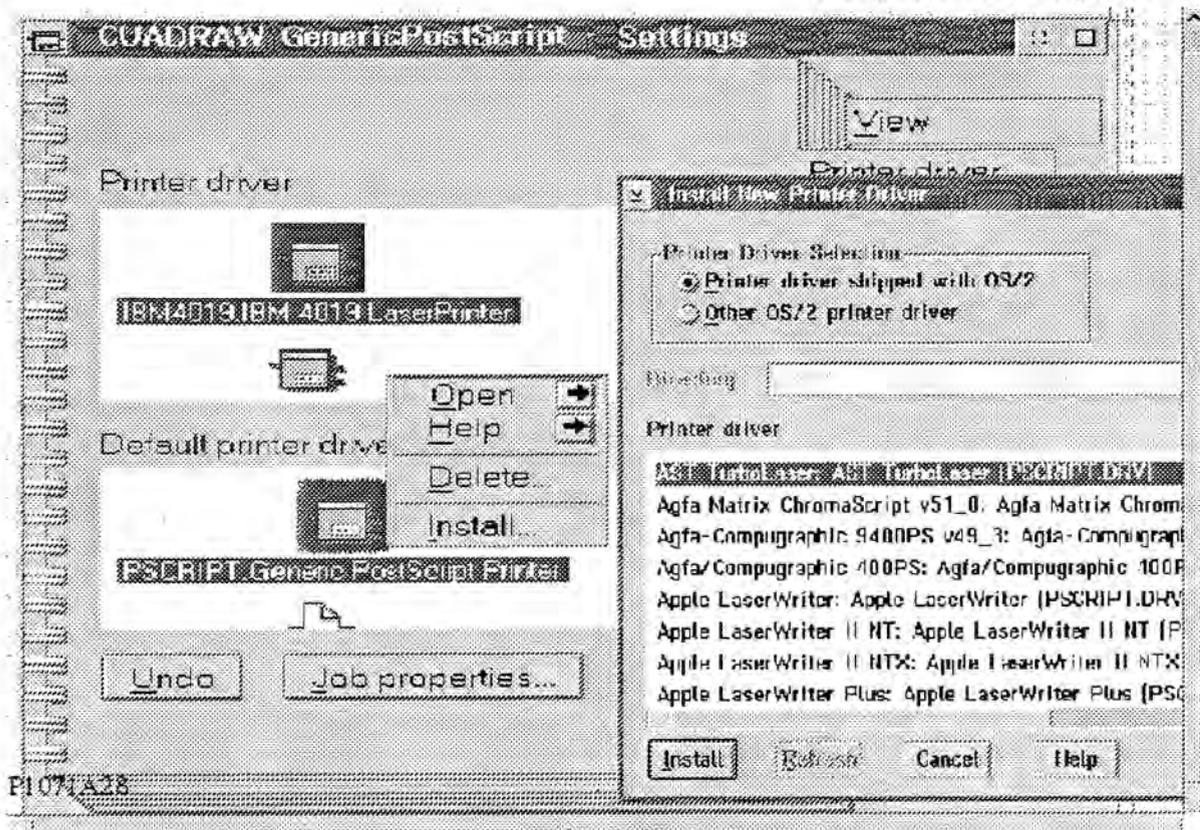


Figure 10-15. From an Existing Printer Object

To install a printer device driver from an existing printer object:

1. Bring up the context menu of an existing Printer object.
2. Click on the arrow of *Open*
3. Click on *Settings*
4. Click on the *Printer driver* tab
5. Display the context menu for any of the printer driver objects displayed in the *Printer driver* container.
6. Click on *Install*.

This will bring up the *Install New Printer Driver* dialog:

7. Insert the device driver diskette in drive A or type in the path.
8. Click on *Refresh*.
9. Click on the printer driver you want to install
10. Click on *Install*.
11. Click on *Cancel*.

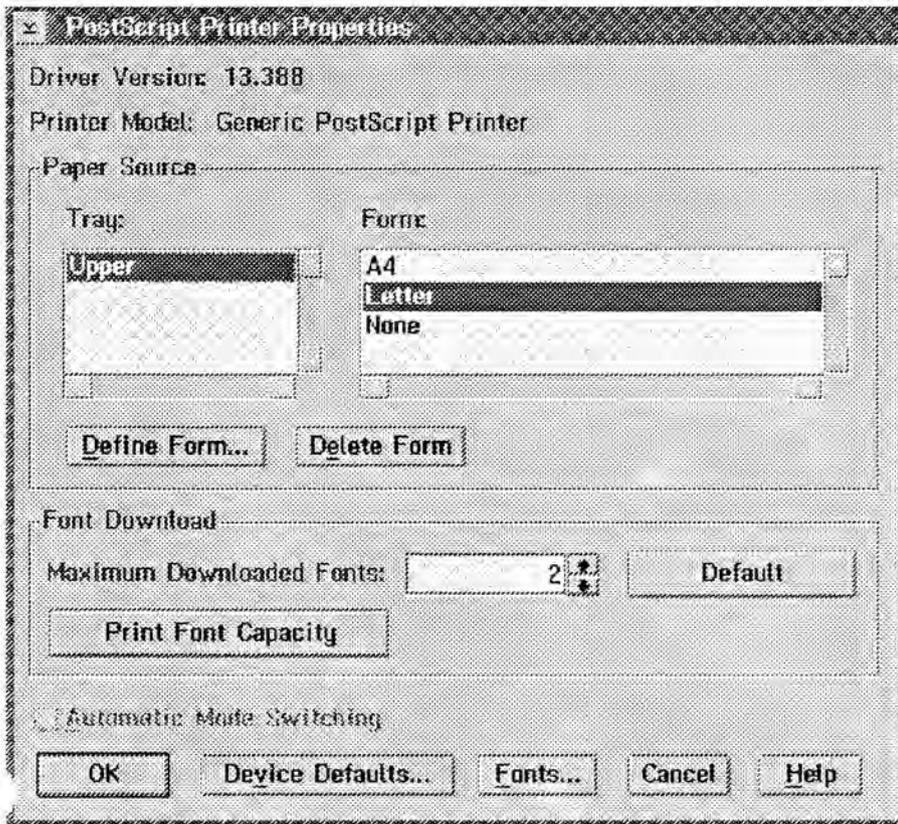


Figure 10-16. Printer Properties

The default physical setup of your printer is described to the system through printer properties within the printer driver settings of the printer object.

To view or change printer properties:

1. Display the context menu for *any* printer object whose default printer driver is the one you wish to view.
2. Click on the arrow to the right of *Open*
3. Click on *Settings*
4. Click on the *Printer driver* tab
5. Double click on the icon for the *Default printer driver* object to display the printer properties dialog.
6. View or change the printer properties as you wish.

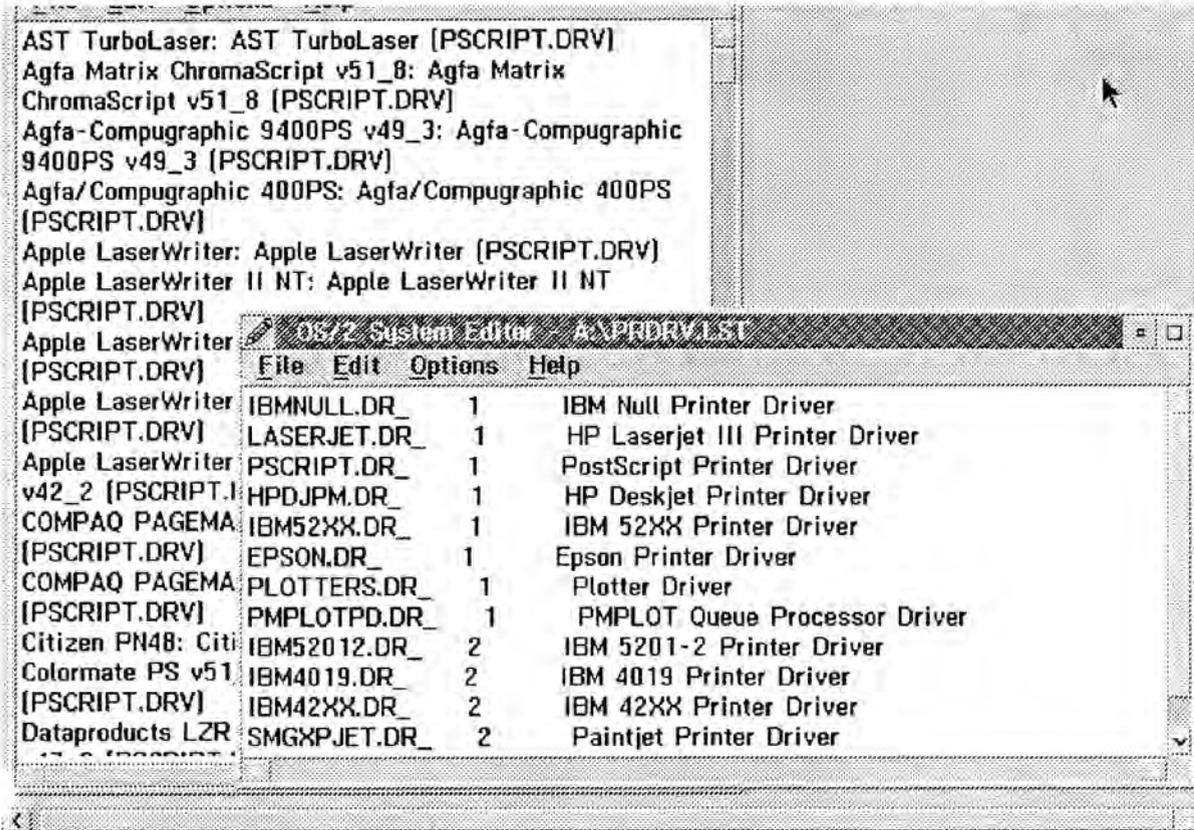


Figure 10-17. Listing supported Printers and Drivers

To find out which printer driver to select for a particular printer model:

1. Insert OS/2 device driver diskette 1 in Drive A
2. Double click on the *Drive A* icon on the Workplace Shell
3. Double click on the **PRDESC.LST** icon

A window appears that lists supported printers and plotters by model name. The printer driver for a particular model appears after the model name.

To find out which OS/2 device driver diskette contains the printer driver you want:

1. Insert OS/2 printer driver diskette 1 in Drive A
2. Double click on the **Drive A** icon on the Workplace Shell.
3. Double click on the **PRDRV.LST** icon.

A window appears that lists the printer drivers by their data file name and shows the device driver diskette number for each.

Subtopic Summary

In this subtopic the student learned to customize print output by manipulating printer drivers within the OS/2 Version 2.x print subsystem.

The student was taught how to:

- Install a printer driver
- Access the printer properties dialog of a printer driver.

This concludes another subtopic of the topic
"OS/2 Version 2.x Printing".

Spooler, Queue, and Job

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to customize print output by manipulating the port, spooler, queue, and job objects.

Enabling objectives:

After attending this subtopic the student should be able to:

- Install and customize a port object.
- Enable/Disable and define the path of the spooler.
- Customize a queue by setting its options.
- Identify the status of a job by observation of its icon.

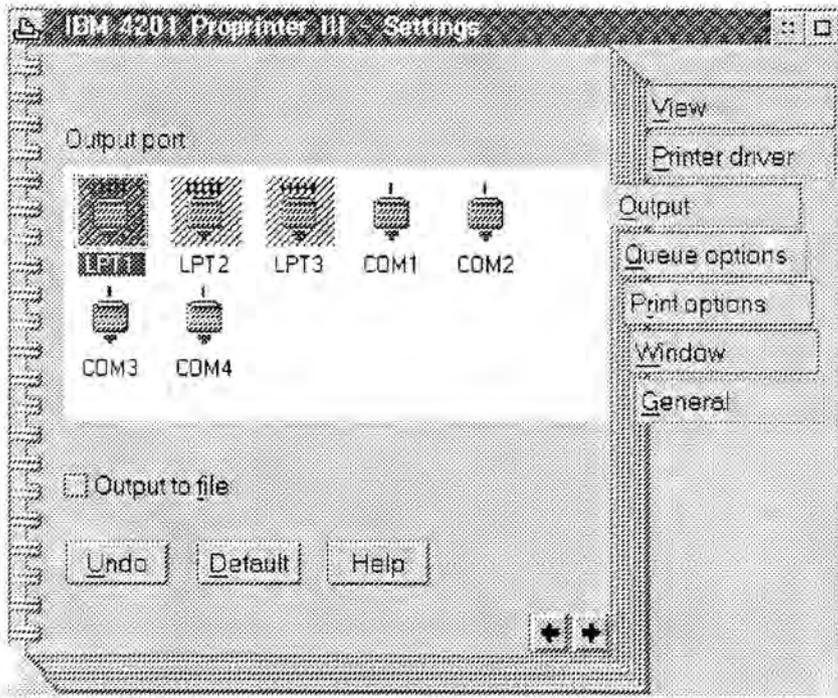


Figure 10-18. Port Driver

- Port drivers are designed to isolate the spooler from the individual settings required by printer device drivers.
- A port driver must be provided for each physical device driver that is used to control printing. OS/2 2.x comes with two:
 1. SERIAL.PDR
 2. PARALLEL.PDR
- Port drivers are DLLs that provide 32-bit functions such as listing the ports that the port driver supports (LPT1-3 or COM1-4).
- The port driver passes values such as:
 - word length
 - baud rate
 - parityto the printer's physical driver.

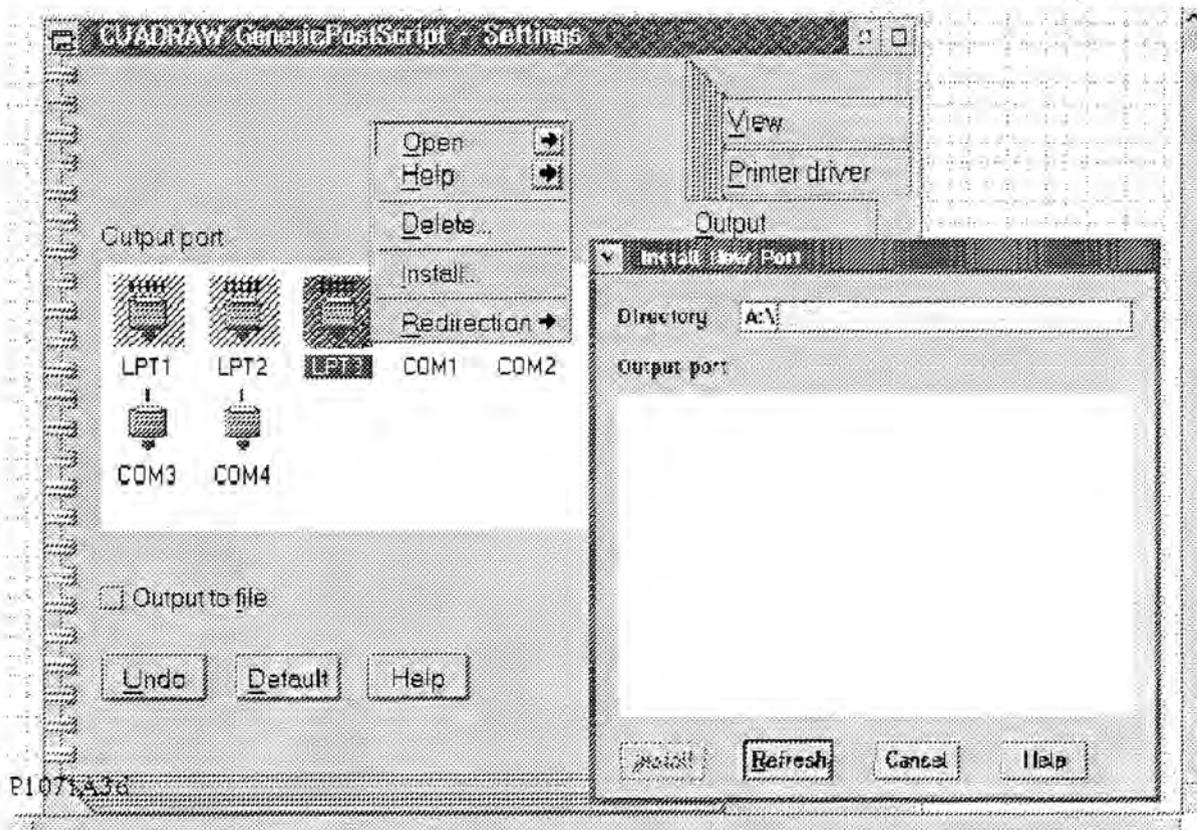


Figure 10-19. Installing Port Objects

Port objects are installed as follows:

1. Insert the diskette containing the port object to be installed into drive A.
2. Double click on *Drive A*
3. Drag one or more port objects from the drive A window to a printer object.

Or, alternatively:

1. Display the context menu for the appropriate printer object.
2. Click on the arrow to the right of *Open*
3. Click on *Settings*
4. Click on the *Output* tab
5. Display the context menu for the port object (use mouse button 2).
6. Click on *Install*
7. If adding ports from a diskette, insert the diskette containing the new port object into Drive A.
8. If installing ports that are supplied with the OS/2 such as LPT1 to LPT3 or COM1 to COM4 then type:

```
\OS2\DLL
```

in the path field

9. Click on *Refresh*.
10. Click on *Install*.

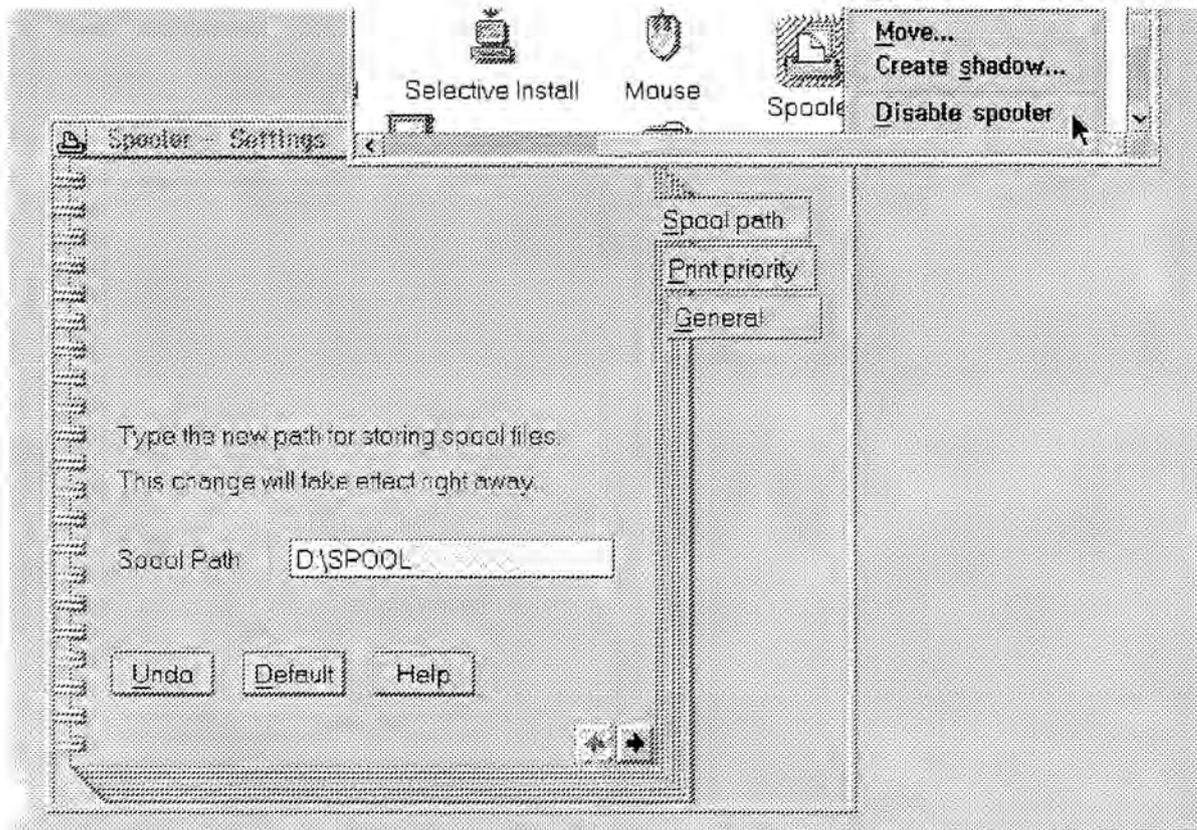


Figure 10-20. Spooler Object

The OS/2 spooler object initially resides in the *System Setup* folder that resides in the *OS/2 System* folder. It represents the settings of the OS/2 spooler and **not** its content.

Through the spooler object, the spooler can be disabled, enabled (default state), and its spool path (to the hard drive) can be changed.

If you disable the spooler the print subsystem will be unable to keep your print-jobs separate. This inability means that if the system has more than one job request, it will send each part of a print-job directly to the printer. Consequently, partial job data from the various jobs arrive at the printer in the order they were sent to it. To keep your print-jobs separate, either enable the spooler or wait for each print-job to finish printing before sending the next.

It is strongly recommended that the spooler be always enabled. An active spooler is especially important when you're working with network devices because then it is very difficult to prevent mixing up print-jobs from different LAN users sent to one network printer or plotter.

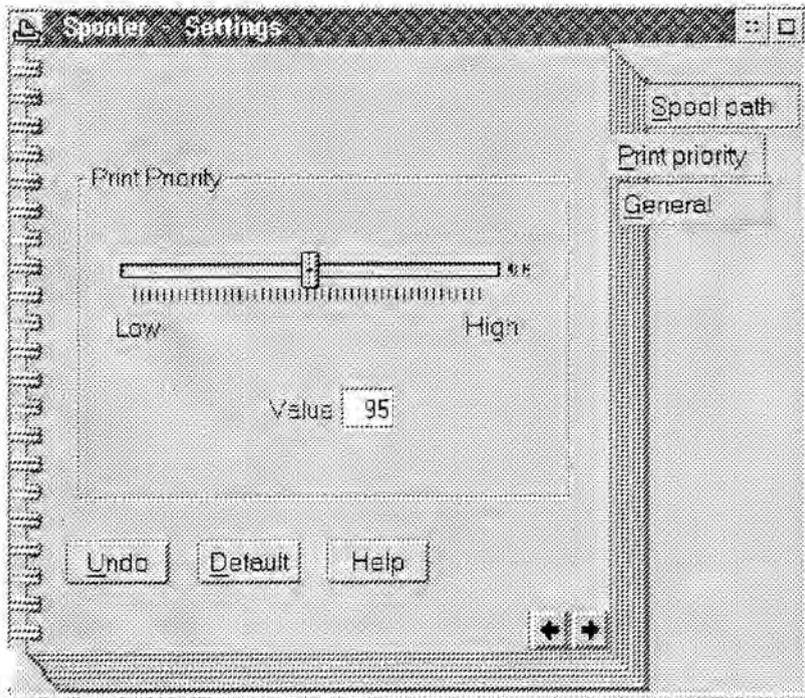


Figure 10-21. The Print Priority Setting

The print priority setting allows you to balance and optimize the system's performance between printing and running applications.

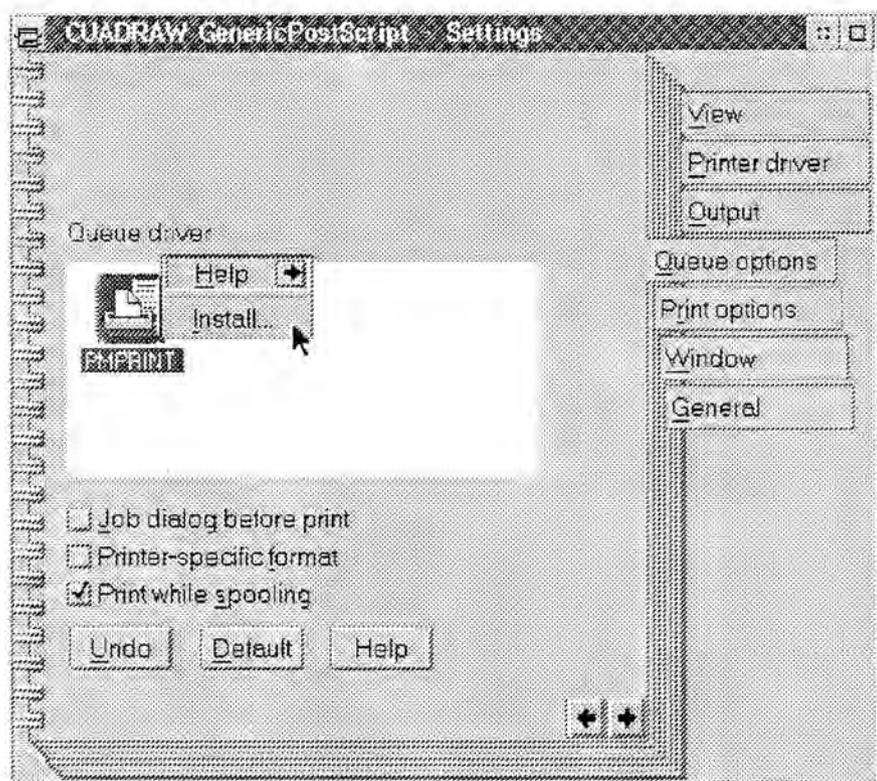


Figure 10-22. The Queue Driver Object

The queue driver is presented to the user as a queue driver object. The queue driver is called upon by the spooler to pass queued print-jobs on to the printer driver. Queued jobs are spooled (written) to a file on the hard drive where they wait until it is their turn to be printed.

Queue driver objects can be installed, deleted and changed. There are two queue drivers shipped with OS/2 2.x, PMPRINT.QPR and PMPLOT.QPR, which have different selectable options. These options can be set through the *Queue Options* in the settings notebook for the particular printer object.

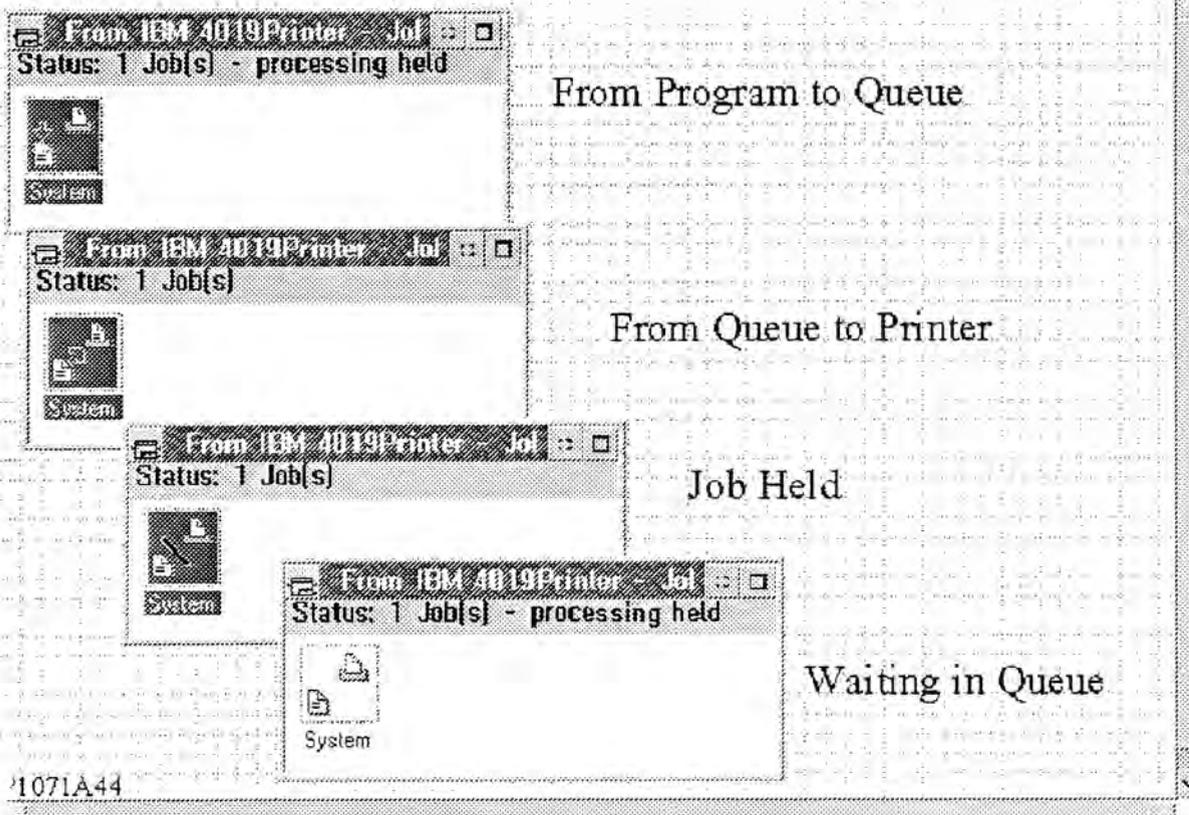


Figure 10-23. Job Object

Depending on the default view setting of a Printer object, the print jobs are represented to the user as either job objects appearing as icons (icon view) or text lines (details view). Both of which display the job print status.

Job objects are visible from within a printer object container window. Double clicking on a printer object brings up (by default) the Job Icon View. Any jobs sent to a printer object will appear. The current status of a job will be indicated graphically by the appearance of the icon (above).

Normally, the local printer object status is updated automatically. Network printer object status is updated at every refresh interval or when you select *Refresh* from the printer object's Job Icon View context menu.

§ Topic Summary

In this subtopic the student learned how to customize print output by manipulating the port, spooler, queue, and job objects.

The student was taught how to:

- Install and customize a port object.
- Enable/Disable and define the path of the spooler.
- Customize a queue by setting its options.
- Identify the status of a job by observation of its icon.

This concludes another subtopic of the topic
"OS/2 Version 2.x Printing".

Win-OS/2 Printing

Subtopic objectives

Terminal objective:

After attending this subtopic the student should be able to customize print output generated by applications running in the Win-OS/2 environment.

Enabling objectives:

After attending this subtopic the student should be able to:

- List the Print Manager, Control Panel, and the Adobe Type Manager as the three major components of Win-OS/2 printing.
- Install a printer using the Control Panel.
- Assign an output port to a Win-OS/2 printer.
- Access a Win-OS/2 printer driver and change its settings.
- Activate a printer.
- Use the Print Manager to rearrange Win-OS/2 print jobs.

Adobe Type Manager™

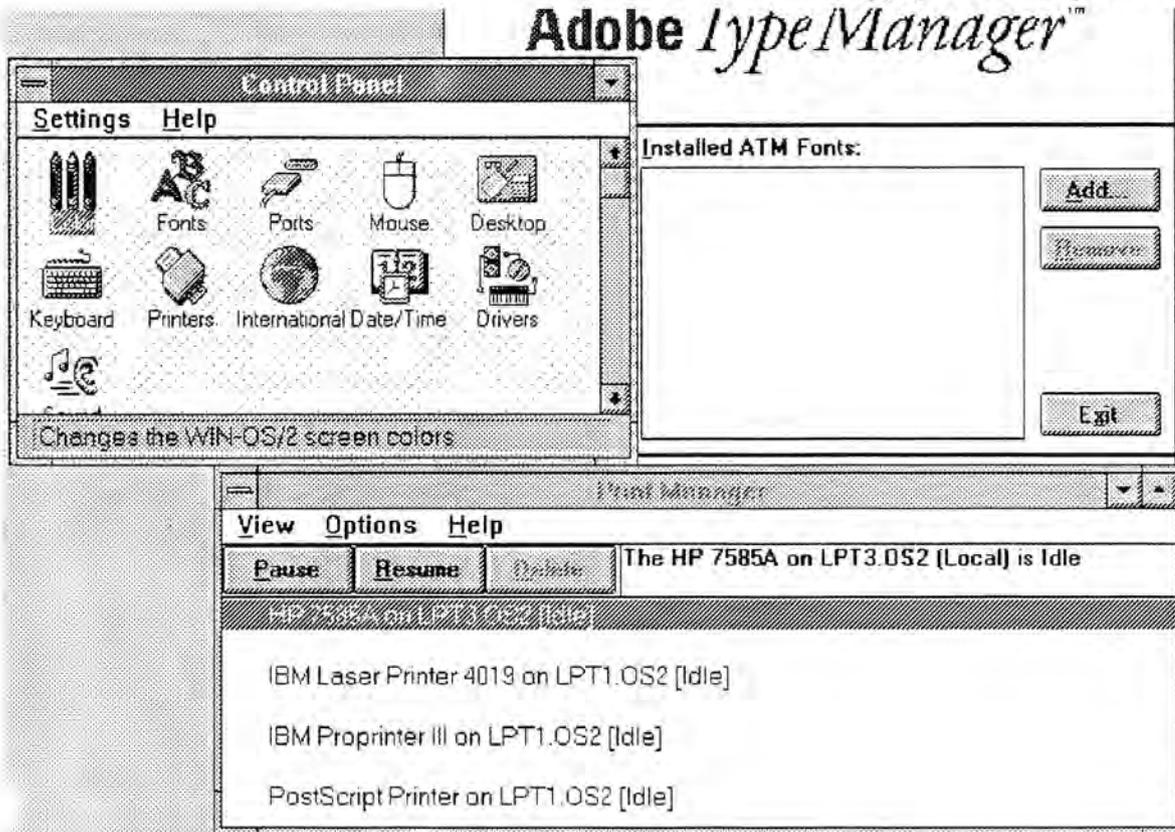


Figure 10-24. Win-OS/2 Print Components

WIN-OS/2 print support is provided through the Win-OS/2 Print Manager and Control Panel. The Win-OS/2 user interface is accessed through a Win-OS/2 full screen or windowed session.

By using the windowed (Seamless) session(s) it is possible to display the Win-OS/2 print subsystem components on the Workplace Shell. These components are:

- Print Manager
- Control Panel
- ATM Control Panel

The Win-OS/2 Print Manager and Control Panel provide the user the management and configuration options (respectively) for printing from Win-OS/2 applications. The ATM Control Panel is used to manage fonts for Win-OS/2 printers.

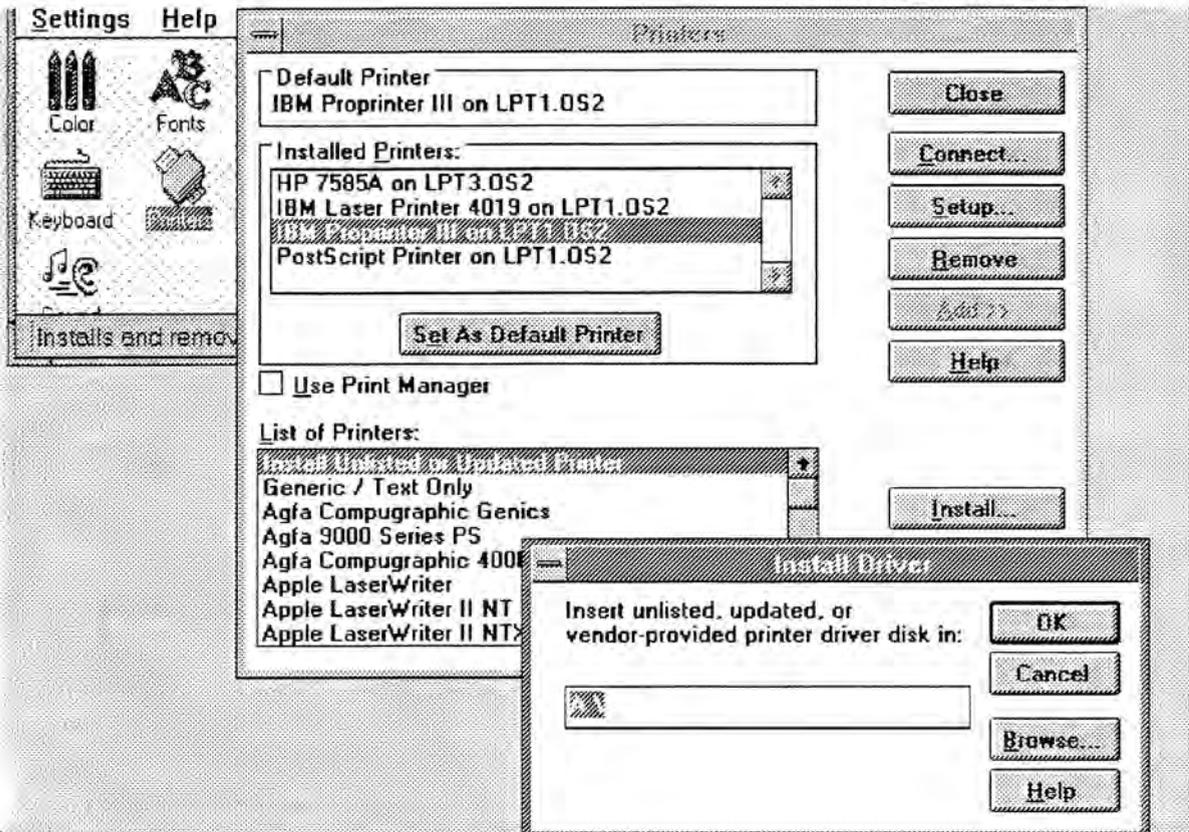


Figure 10-25. Installing a Printer

The Win-OS/2 Control Panel enables you to configure your printers and ports. To configure a printer for use with Win-OS/2, you have to perform the following steps:

1. Install the printer driver
2. Assign an output port
3. Change Printer Settings if required
4. Make the printer active.

To install a printer along with its appropriate driver:

1. Double click on the Win-OS/2 *Control Panel* icon.
2. Double click on the *Printers* icon.
3. From the list, select the printer that you want to install and click on the *Add Printer >>* button.
4. Click on *Install*.

You are prompted to insert a diskette. When done, a list of drivers is presented on the screen.

5. Click on *Install*.

The printer name then appears in the *Installed printers* list. Before you can use the printer, you must complete the other installation steps.

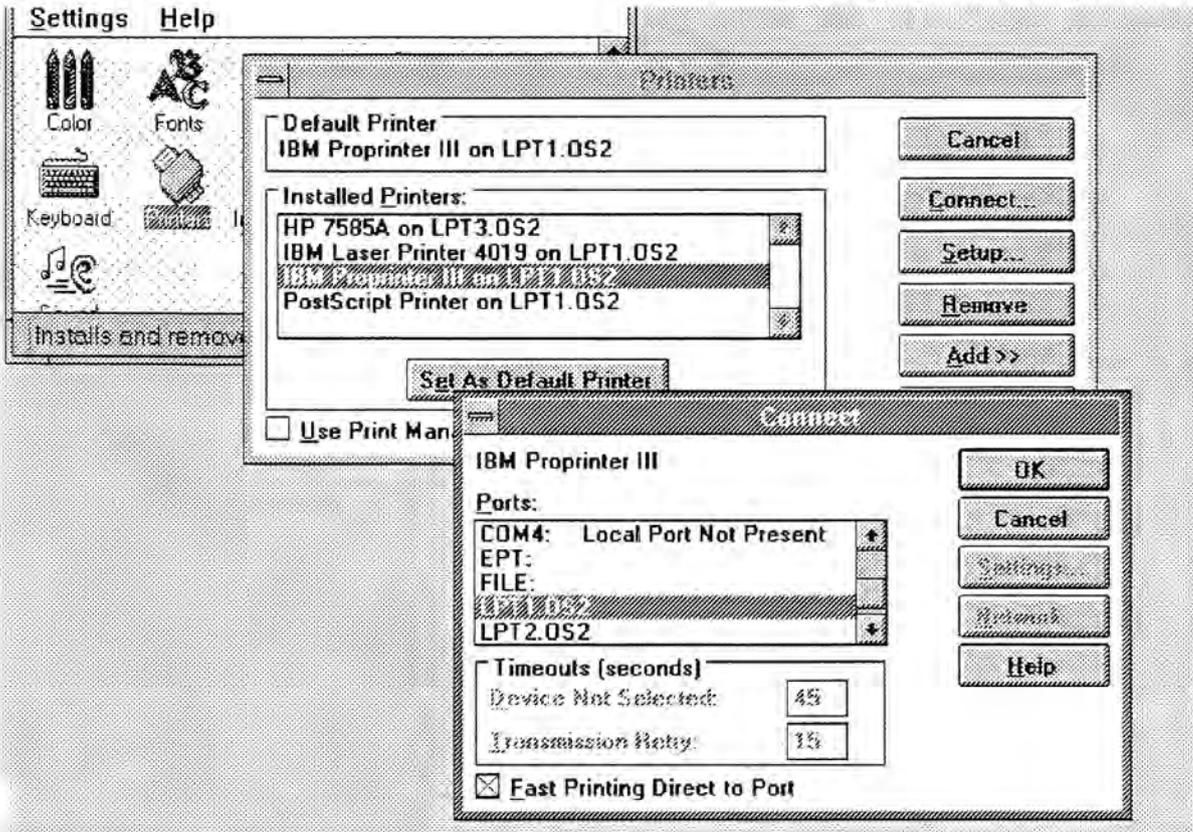


Figure 10-26. Assigning an Output Port

1. Install the printer driver.
2. **Assign an output port.**
3. Change Printer Settings if required
4. Make the printer active.

Double click on the *Printers* icon in the Control Panel window then:

1. Click on the printer in *Installed Printers*.
2. Click on *Connect*.
3. Click on the port.

From the **Ports** listbox, choose the port you want to assign to the printer from the LPT1.OS2 to LPT2.OS2, LPT1 to LPT3, COM1 to COM4, or EPT selections.

4. Click on *OK*.

If you are installing a new printer then you must complete the other installation procedures before you can use the printer.

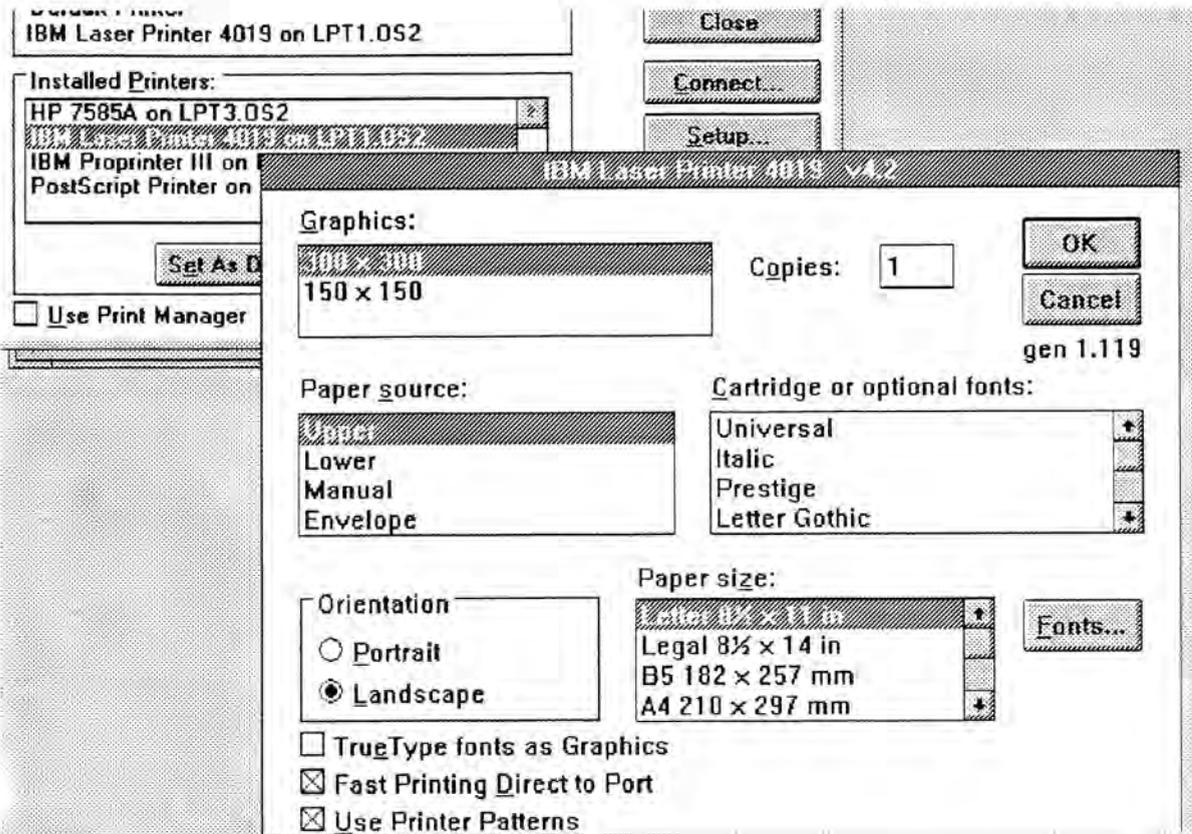


Figure 10-27. Change Printer Settings

1. Install the printer driver.
2. Assign an output port.
3. **Change Printer Settings if required.**
4. Make the printer active.

For most printers you install, you can set options such as page orientation, paper size, and graphics resolution. The precise options available vary from printer to printer. To select Printer Settings:

1. Double click on the *Printers* icon in the *Control Panel* window.
2. Click on the printer in the **Installed Printers** list.
3. Click on *Setup*.
4. Change the desired setup parameters.
5. Click on *OK*.
6. Click on *Close*.

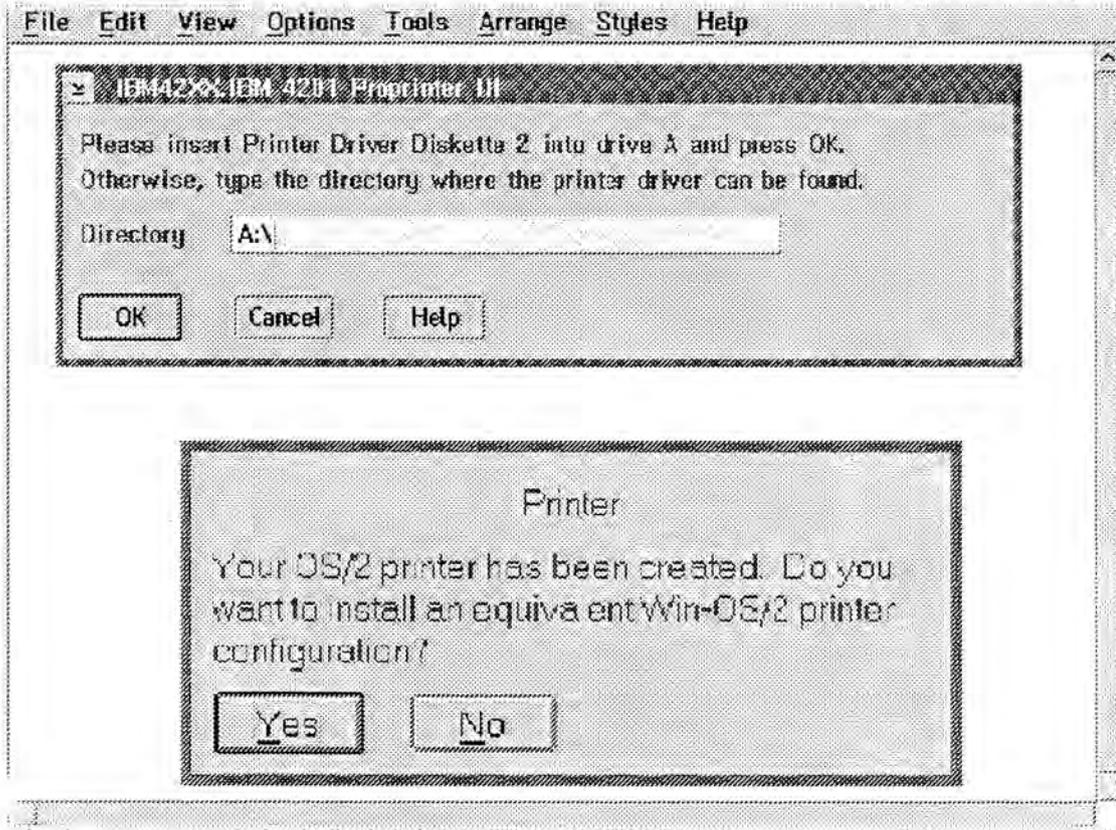


Figure 10-28. WIN-OS/2 Printer Driver

After selecting the desired printer driver(s) from the *Install New Printer Driver* dialog, you are prompted for the printer driver diskette which is a new dialog. After installation, you will sometimes be asked if you require the corresponding printer driver.

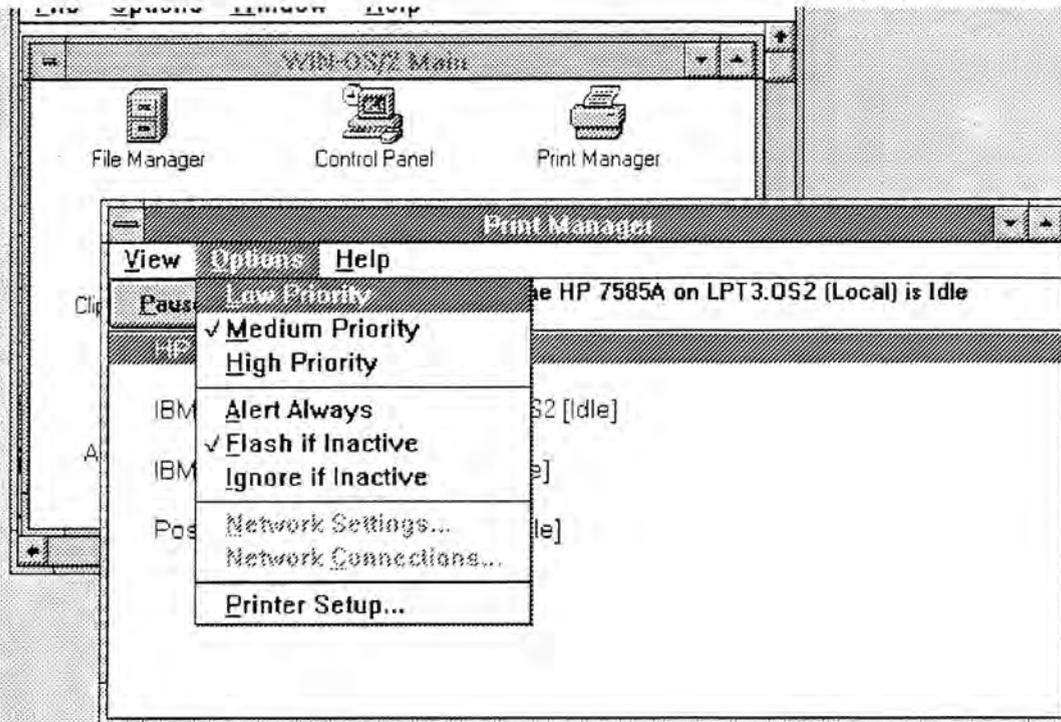


Figure 10-29. Print Manager

The Win-OS/2 Print Manager window displays the current status of all active print queues. Print jobs are listed below the appropriate printer until they have been printed completely. The status of each print queue is displayed on the printer information line. The status of each print job is displayed on the job information line beneath each printer information line.

The *Low*, *Medium*, and *High* settings under *Options* influence the performance of other applications while printing.

To change the order of a job that has not started printing, simply drag and drop it up or down the list of jobs.

Topic Summary

In this subtopic the student learned to customize print output generated by applications running in the Win-OS/2 environment.

The student was taught to:

- List the Print Manager, Control Panel, and the Adobe Type Manager as the three major components of Win-OS/2 printing.
- Install a printer using the Control Panel.
- Assign an output port to a Win-OS/2 printer.
- Access a Win-OS/2 printer driver and change its settings.
- Activate a printer.
- Use the Print Manager to rearrange Win-OS/2 print jobs

This concludes the final subtopic of the topic
"OS/2 Version 2.x Printing".

Topic Summary

In this topic the student learned how to utilize OS/2 Version 2.x's object oriented approach to printing.

The student was taught how to customize print output by manipulating the following components of the OS/2 Version 2.x print subsystem:

- Printer objects
- Printer driver objects
- Port, job, spooler and queue
- Win-OS/2 Printing

This concludes the topic
"OS/2 Version 2.x Printing".





TOPIC 11: Fonts

Topic objective:

Enabling objectives:

Upon completion of this topic the student should be able to:

- Describe the font support provided by OS/2
- Install fonts for use in the OS/2 and WIN-OS2 environments

(

(

(

Helvetica - Normal *Italic* **Bold**
Bold Italic

Times New Roman - Normal *Italic*
Bold ***Bold Italic***

Courier -Normal *Italic* **Bold**
Bold Italic

Symbol - Σψμβολ

Additional fonts are available P1071B00

Figure 11-1. Fonts shipped with OS/2

The IBM core fonts consist of a set of thirteen Adobe Type 1 fonts that work with the Adobe Type Manager (ATM). Unless you specify otherwise, these fonts are installed during installation.

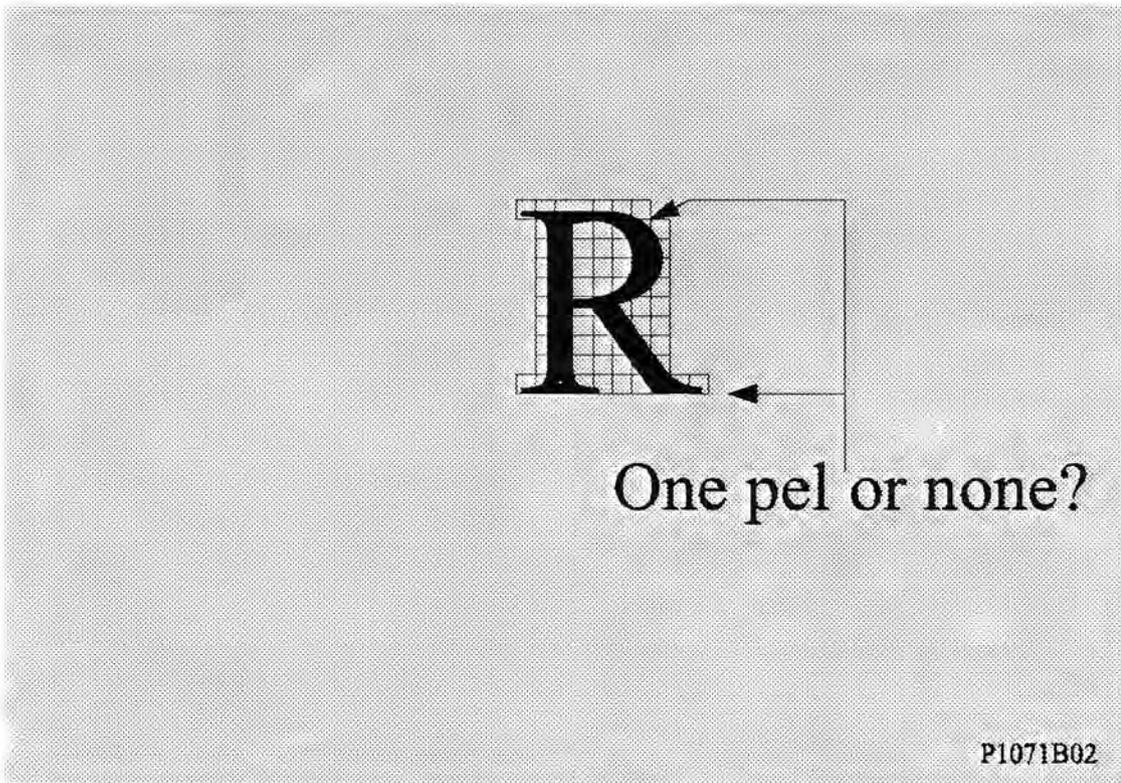


Figure 11-2. Adobe Type Manager

The Adobe Type Manager (ATM) is an integral part of the OS/2 operating system and works with existing OS/2 and WIN-OS2 application programs to produce the sharpest possible fonts on the screen and on printed paper.

The fonts supplied are Type 1 fonts which are rasterized by ATM. This means that they give the output device information on how to draw the character. This is useful because it makes the characters scalable. Instead of having different character definitions for each point size, the basic definition remains unchanged and the information is scaled to produce the desired size.

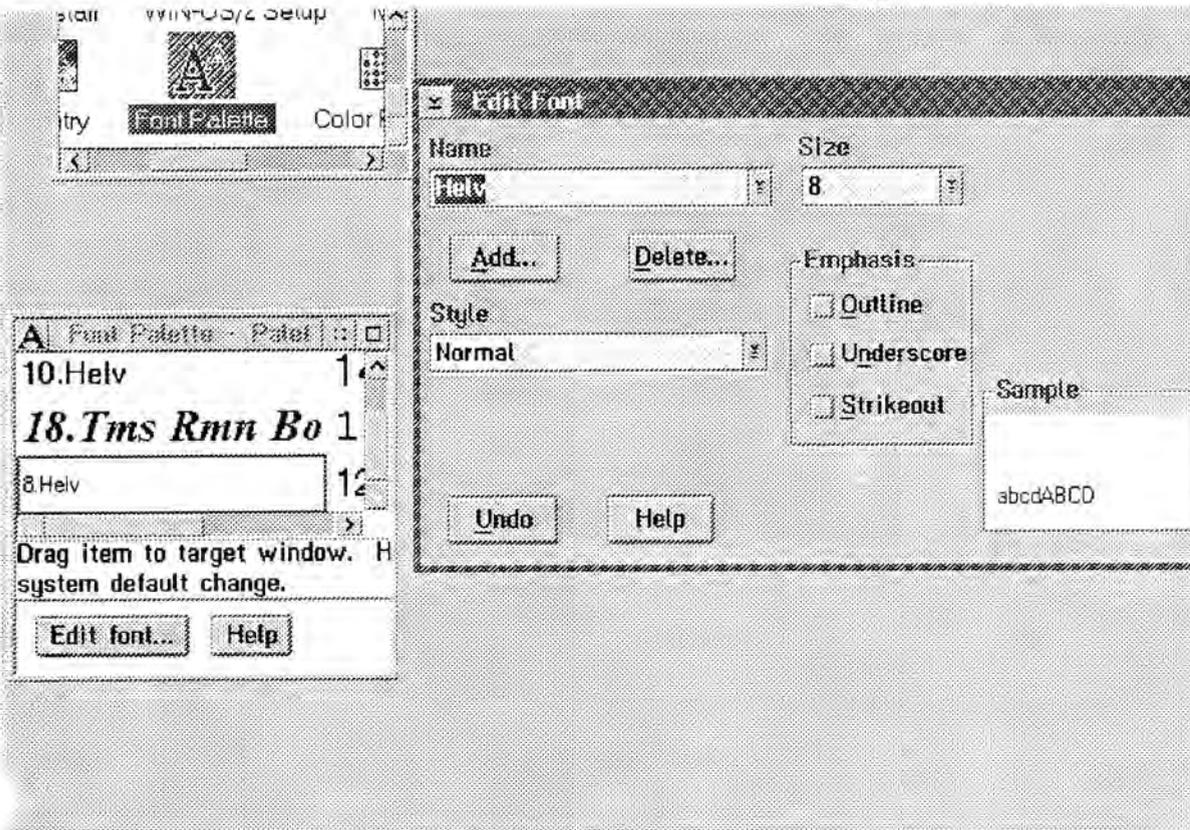


Figure 11-3. Adobe Type Manager

The Font Palette allows you to select display fonts for system objects. If you want a font that is not installed or available on the palette, use the *Edit Font...* button. To add a new font, select one of the fonts indicated on *Edit font...*. You can change the font (the **Name** field), the fonts' style, size and, Emphasis. If you want a base font that is not on your system, select *Add...* and insert the diskette containing the appropriate font.

If you want more fonts than the eight fonts available from the Desktop, you can create additional font palettes, using the Font Palette template in the Templates folder.

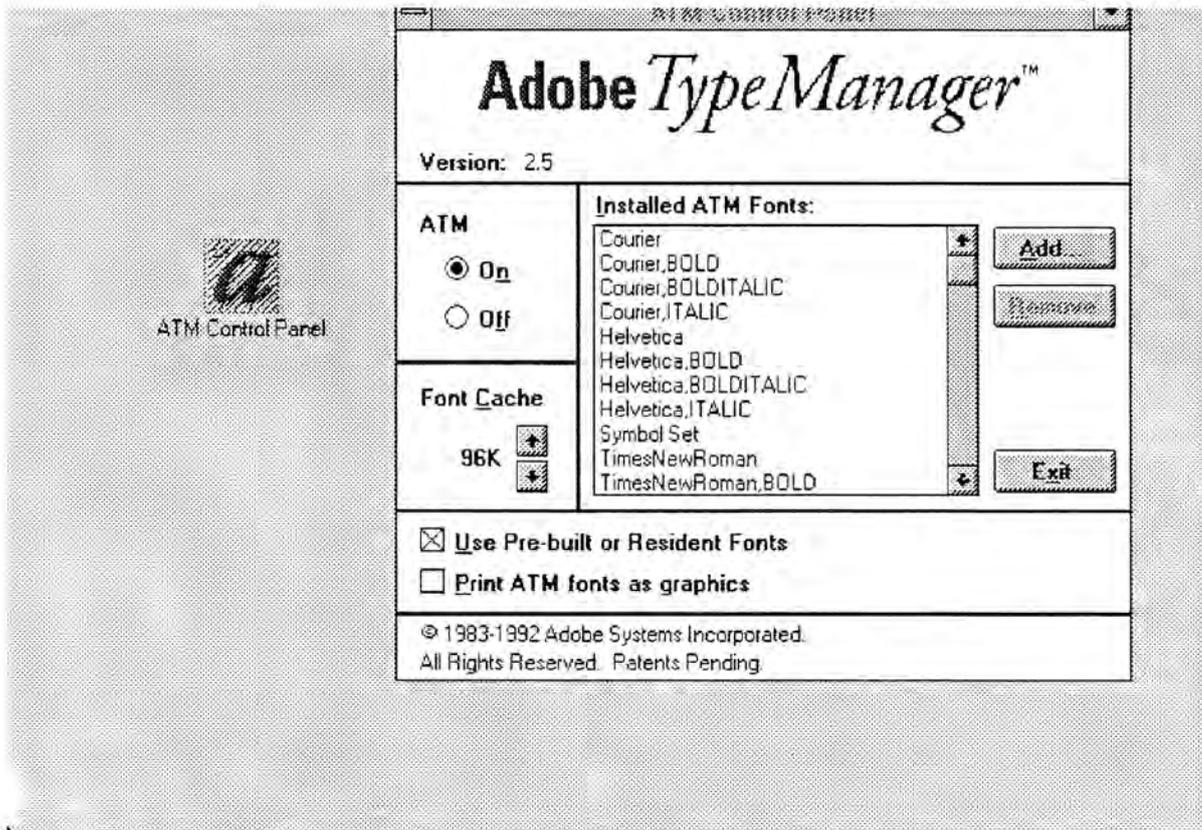


Figure 11-4. Adobe Type Manager

Windows 3.1 includes TrueType** font support. TrueType is an advanced font technology similar in concept to ATM, which enables scalable fonts to be accurately displayed and printed.

WIN_OS2 3.1 includes both TrueType and ATM font technologies. TrueType is installed by default and can be selected using the WIN-OS2 Control Panel - Fonts icon. However, since ATM is also implemented in OS/2, it is recommended that ATM be used in preference to TrueType since this simplifies moving documents and clipboard data between WIN-OS2 and Presentation Manager.

The thirteen IBM Core fonts installed are also available for use in WIN-OS2, but they are not installed at installation time. They must be installed using the WIN-OS2 ATM Control Panel.

The ATM Control Panel is used to add and remove ATM fonts, change the font cache size, use pre-built or resident font software, and turn ATM on or off. When you change any of the choices in the ATM Control Panel, except the **Use Pre-built or Resident Fonts**, you must exit and then restart WIN-OS2 for your changes to take effect.

Summary

In this topic the student learned about:

- What font support is available from OS/2
- How to install fonts for OS/2 and WIN-OS2

This concludes the topic
"Fonts."



P1071 LABS:

LAB: Installation with Dual Boot

OS/2 Installation Lab

Background

One can install OS/2 to share the same partition in which DOS already resides. This procedure is known as Dual Boot. The supported versions are DOS 3.2 or later.

When one plans to use Dual Boot there are certain requirements prior to installing OS/2 that must be met:

- All DOS commands **MUST** be located in a subdirectory such as C:\DOS, and **not** in the root directory.
- All DOS utility programs **MUST** be located in a subdirectory such as C:\DOS, and **not** in the root directory.
- The DOS files CONFIG.SYS and AUTOEXEC.BAT must be placed in the root directory.
- Statements in the DOS CONFIG.SYS and AUTOEXEC.BAT files must be pointing to the new path i.e. C:\DOS.

Objectives

- Verify that the Dual Boot requirements have been met
 - Install OS/2 to share the same partition as DOS
-

Verifying that the Dual Boot requirements have been met

1. Power on your machine
2. When you get the command prompt, verify that the following requirements have been met:
 - Is there a **DOS** subdirectory?
 - try typing in **DIR** to determine this
 - Is **autoexec.bat** in the root?
 - Is **config.sys** in the root?
 - Do statements in autoexec.bat point to C:\DOS?
 - try typing in **TYPE AUTOEXEC.BAT** to determine this
 - Does the **SET COMPSEC** point to C:\DOS?
 - Does the **PATH** point to C:\DOS?
 - Do statements in config.sys refer to the path C:\DOS?
 - try typing in **TYPE CONFIG.SYS** to determine this
 - Is there a **command.com** in C:\DOS?
 - try typing in **CD DOS** and then

— typing in **DIR COMMAND.COM** to determine this

If the appropriate statements are not there, make those changes.

You can add the following statement to your AUTOEXEC.BAT file

- **COPY C:\DOS\COMMAND.COM C:\ > NUL**

This statement is not necessary for Dual Boot to work correctly, it is an optional statement. Some DOS programs will not run correctly if the COMMAND.COM file is not in the root directory. This statement has the effect of copying that file to the root directory each time you start the computer.

Installing the OS/2 operating system, Phase I

1. Insert the **OS/2 Installation diskette**.
2. Restart your system.
 - Press the <Ctrl> + <Alt> + keys

Notice the first screen that appears from **Diskette 1**. It offers you a chance to cancel and stop the installation. Press <Enter> to continue with the installation of OS/2.
3. Follow the instructions until you get to the **Installation Drive Selection**.
4. At that screen, select option 1, **Accept the drive**.
5. Diskette number 2 will be requested.
6. Select **Do not format the partition**.
7. Follow the instructions.

You will be requested for specific diskette numbers to be inserted in drive A. After you have been requested to insert the Installation diskette one more time, the system will ask you to remove the current diskette and press <Enter>. This will indicate the end of the phase I installation.

Installing the OS/2 operating system, Phase II

1. When the graphical installation starts, select **Install preselected features**.

Notice: At any time, you can request help to assist you in your decision making.
2. Accept the default mouse, keyboard, country and display shown.
3. Click on the **OK** push button.
4. Select the printer situated at your desk. If you do not have a printer, accept the default "Do Not Install Default Printer".
5. Click on the **OK** push button once more.

You will be warned that installation is about to start. Afterwards, you will be requested to insert a diskette.

At this time, please let your instructor know you have reached this phase

Notice: An indicator will indicate which diskette is currently being used and which file is currently being copied from diskette to the hard drive.

6. Continue following the instructions.
7. An **Advanced Options** panel will appear.
This screen is used to select one or more of the Advanced Installation options.
8. For this installation, do not select any of the Advanced options so, remove all check marks and press on the **OK** button.
9. Continue following the instructions.
10. The system will indicate when OS/2 is installed and what action to take.

The first time OS/2 runs on your machine....

The first time OS/2 gets started on your machine, it will by default start the **Tutorial**. If you have not gone through the tutorial, do so at this time. Otherwise, press the **Exit** button to exit out of the tutorial.

Depending on the machine you are using, you may have to wait one to several minutes for the Desktop to be created that first time. Notice that although the screen may appear to be doing nothing, the disk light is continuously flashing.

When the Desktop is final, it is important to properly shut down the system. This action should always be performed that first time. It is more of a preventive nature than a mandatory action. Use the proper method to shutting down the system.

PROPERLY shutting down OS/2

If you do not remember how to perform a proper shut down, follow the next five instructions.

1. Move your mouse pointer to an unused area on the Desktop.
2. Press once on button 2.
This will pop-up the Desktops' contextual menu.
3. Move your mouse pointer to the word **Shut down....**
4. Select **OK** to continue the shutdown.
5. Do not use your machine until you see the following message:

Shutdown has completed. It is now safe to turn off your computer, or restart the system by pressing <Ctrl> + <Alt> + .

Using the Dual boot feature to boot from OS/2 to DOS

After you have installed the OS/2 operating system, you can the BOOT command to switch from one operating system to another.

1. Open the **OS/2 System** folder.
2. Open the **Command Prompts** folder.
3. Open **OS/2 Window**.
4. Type **boot /dos** and then press the <Enter>.
5. You will be asked if you are sure you want to continue.
6. If you are ready to boot to DOS, type in **Y**.

A message will appear indicating that the hard disk is being prepared and to wait...

Question: When you use the BOOT command, will it save data from currently running programs?

Question: If you have any running programs, will they be restarted the next time OS/2 is started?

Question: What is the system doing with the config.sys and the autoexec.bat that OS/2 uses?

Using the Dual boot feature to boot DOS to OS/2

Once you have finished using native DOS, the BOOT command is used to return to OS/2.

1. Type **CD \OS2** .
2. Type **boot /os2** and then press <ENTER>.
3. You will be asked if you are sure you want to continue.
4. If you are ready to boot to OS/2, type in **Y** .

Using the Dual boot feature to boot from OS/2 to DOS

There is another way to start the Boot command from OS/2:

1. Open the **OS/2 System** folder.
2. Open the **Command Prompts** folder.
3. Notice the **Dual Boot** Object.
4. Open the Dual Boot object.
5. You will be asked if you are sure you want to continue.
6. To prevent Dual boot from occurring, type **N** at the prompt.

PROPERLY shutting down OS/2

1. Move your mouse pointer to an unused area on the Desktop.
2. Press once on button 2.
This will pop-up the Desktops' contextual menu.
3. Move your mouse pointer to the word **Shut down...**
4. Select **OK** to continue the shutdown.
5. Do not use your machine until you see the following message: can turn off your machine.
Shutdown has completed. It is now safe to turn off your computer, or restart the system by pressing <Ctrl> + <Alt> + .

You have now completed the lab.

LAB: Boot Manager Installation

Boot Manager installation Lab

Background

One can install multiple operating systems on the hard drive. To do so, one must install the Boot Manager which will permit you to select which operating system to activate at system startup.

The Boot Manager requires the use of a primary partition for itself. OS/2 1.x and DOS operating systems must be installed in primary partitions. OS/2 2.x can be installed on either primary or logical partitions.

Objectives

- Install Boot Manager in one partition
- Add partitions.
 - Primary partition for the Boot Manager
 - Primary partition for DOS 5
 - 4 Logical partitions (D, E, F and, G)
- Install DOS 5 in the primary partition NOT utilized by Boot Manager or the existing OS/2 and DOS co-existence.

Recall the previous lab, you installed OS/2 to co-exist with DOS 6 on a primary partition. This partition is NOT to be deleted or altered.

Current status of the machine disk

Start FDISK from an OS/2 Full Screen.

1. Notice the existing partitions, what are they?
 - What type of partition is C? Logical or Primary
 - What file system type is C? _____
 - What size is partition is C? _____
 - What is the Status of partition C? Bootable or Startable or None
 - What other partitions are there? _____
 - What type of partitions are they?
 - _____
 - _____
 - _____

Installing Boot Manager

1. Highlight **Freospace** line.
2. Press the <Enter> key to display the Options menu.
3. Select **Install Boot Manager**.
4. Select **Create at End of Free Space** and press <Enter>.

Notice that automatically the size of the Boot Manager is 1 MB. Also, notice that Help is available at all times to assist in your decision making.

Question: Could you have selected to create Boot Manager at the start of the Freespace?

Question: If you had selected at the start of the freespace, would you have come across any difficulties? Try it and see.

Creating another primary partition

This other primary partition will contain DOS 5

1. Highlight the Free Space line and press the <Enter> key to display the Options menu.
2. Select **Create Partition** from the Options menu.
3. Type **5** as the size of the primary partition and press <Enter>.
4. Select **Primary partition** and press <Enter>.
5. Select **Create at End of Free Space** and press <Enter>.

Question: Could you have selected to create this partition at the start of the Freespace?

Question: If you had selected at the start of the freespace, would you have come across any difficulties? Try it and see.

Creating logical partitions

Create a **D**, **E**, **F** and, **G** logical partition. Use the following sizes:

- D to be 40 MB
- E to be 30 MB
- F to be 25 MB
- G to be what ever is left over

27
20
8

1. Highlight the Free Space line and press <Enter>.
2. Select **Create Partition** from the Options menu.

3. Type the size of the logical partition, as given by your instructor, and press <Enter>.
4. Select **Logical partition** and press <Enter>.
5. Select **Create at Start of Free Space** and press <Enter>.

A D partition should now have been created. Repeat the above steps to create the other logical partitions.

Adding names to the Boot Manager Menu

Now that all of the partitions are created, add the two Primary partitions and the D partition to the Boot Manager Menu.

1. Highlight the new C partition you have created (not the one that was already created for you at the start of the first lab).
2. Press the <Enter> key to display the Options menu.
3. Select **Add to Boot Manager Menu....**
4. Type **DOS 5** and press <Enter>.

Question: Why is this new partition classified as Unformatted?

5. Highlight the other C partition. (the one with OS/2 installed).
6. Press the <Enter> key to display the Options menu.
7. Select **Add to Boot Manager Menu....**
8. Type **OS/2 DOS** and press <Enter>.
9. Highlight the D partition.
10. Press the <Enter> key to display the Options menu.
11. Select **Add to Boot Manager Menu....**
12. Type **OS/2** and press <Enter>.

Question: What would happen if you had added the partition name to the Boot Manager Menu after you had created each individual partition?

Question: Why are these partitions given a status of BOOTABLE?

Setting up "OS/2 DOS" as the default startup Operating system

1. Highlight the OS/2 DOS partition and press the <Enter> key.
2. Select **Set startup values...** from the Options menu.
3. Select option **Default**, and then press <Enter>.
4. Accept the **Timer** to be **Yes**.
5. Accept the **Timeout** to be **30**.
6. Alter the **Mode** to be **Advanced**.
7. Press the <F3> key to save those Startup Values.

Notice: To save these values, one must use the <F3> key.

Question: If you use the <Enter> key instead of the <F3> key, what happens?

Saving and activating your partition information

1. Press the **F3** key to exit FDISK.
2. Select **Save and Exit** from the pop up window.

Notice that you will see a small red box indicating that you have to perform a <Ctrl> + <Alt> + . Do NOT do so, *instead* perform a proper shut down.

CTRL
ESC

Setting up to install DOS 5

A way to set up for installing DOS 5 is to use the FDISK command and then indicating which partition is to be used.

1. Get to an **OS/2 Full Screen**.
2. Type **FDISK** and then press <Enter>.
3. Highlight the line that indicates DOS 5 and press <Enter>.
4. When the Options menu appears, select **Set installable**.
5. Press the <Enter> key to set your choice.

Make sure that the **DOS 5** partition line now has the status of **Installable**! If it does not, return to steps 3 and 4.

Make sure that you have selected the DOS partition that YOU have created not the one that had been created for you for the first lab!

Question: Which primary partition now has the drive letter C assigned to it?

Question: If the "OS/2 DOS" primary partition has the drive letter C assigned to it and you installed DOS, what would happen?

6. Press the **F3** key save and exit out of FDISK.

Question: If you were to re-enter FDISK, how would you know which C partition is the new current C partition?

The system will indicate that you are to do a <Ctl> + <Alt> + , **Instead** perform a proper shut down.

The actual installation of DOS 5

Having shut down OS/2, WAIT until you see that the message on the screen indicates that you can turn off your machine. Until you see that message, WAIT.

1. Insert **DOS disk 1 of 3** in Drive A.
2. Reboot the machine with the DOS disk 1 in the diskette drive.
3. Install DOS as per the instructions on that diskette.
4. If you are asked if you wish to format, select yes.

Question: What partitions would you expect to be formatted?

Note: The installation procedure of DOS 5 refuses to install if it finds any partition unformatted. It will determine what partitions are unformatted and will ask if you want to have them formatted. Say YES. If you select NO, you will then find yourself out of the DOS 5 installation.

Automatic return to the Boot Manager Menu

With DOS installed, the next time you re-boot the system, Boot Manager is supposed to take over. In other words, when you re-boot, you should see the Boot Manager Menu. If this menu does **NOT** appear then follow the instructions in the section called **Making Boot Manager startable**. If you do get the Boot Manager Menu, then you have completed this lab.

Making Boot Manager startable

You will need to see how your hard drive is partitioned. This is achieved through the FDISK command.

To get to the FDISK screen, you have a **CHOICE** of using either the DOS 5 FDISK or OS/2 FDISK.

The OS/2 fdisk can be obtained by booting from the OS/2 Installation diskette and Diskette 1 and then pressing the <Esc> key to get to a command prompt.

Making Boot Manager startable using the DOS fdisk command

1. Get to a DOS command prompt:
 - a. Use the **F10** key to get to the Action Bar.
 - b. Press the <Enter> key.
 - c. Highlight the word **Exit**.
2. Type FDISK from the DOS Command prompt.
3. Select option 2 **Set active partition**.
4. Type in the number of the partition that contains the Boot Manager and press <Enter>.
Recall, Boot Manager is 1 MB in size.
5. Press the <Esc> key to return to the FDISK main menu.
6. Press the <Esc> key to activate the changes and exit FDISK.
DOS considers itself to be the only operating system, it will ask you to insert a DOS system diskette, ignore the message and press <Enter>.
7. Remove any diskette from Drive A.
8. The next screen you will see is the Boot Manager menu.

You have now completed this lab.

Making Boot Manager startable through the OS/2 fdisk command

1. Insert the **OS/2 Installation Diskette** into Drive A.
2. Press and hold <Ctrl> + <Alt> + keys to restart the system.
3. When prompted to do so, remove that diskette and insert **Diskette 1**.
4. When the Welcome screen is displayed, press the <Esc> key to display an OS/2 command prompt.
The action of booting from these two diskettes automatically makes Boot Manager given the status of Startable.
5. Remove the diskette from Drive A.
6. Press and hold <Ctrl> + <Alt> + keys to restart the system.

You have now completed this lab.

LAB: Selective Install

OS/2 Selective Installation Lab

Background

One can alter the setup or add a feature to OS/2 any time after OS/2 has already been installed.

From the first lab you were requested to select the **Install preselected features** option. That option copies only the most commonly used features of OS/2 to your hard disk. It does not copy all the features of OS/2. For instance, it does not copy the OS/2 Command Reference.

Objectives

Install the following additional features of OS/2:

1. Documentation - OS/2 Command Reference
2. Tools and Games
3. High Performance File System
4. REXX

Getting to the OS/2 System Configuration Panel

Start the *OS/2 Setup and Installation* program

1. Get the Desktop context menu (also known as the pop-up menu).
2. Select the option **System setup**.
3. Start the **Selective Install** program.

Question: What other method could you have used to start the selective install?

Selecting additional features for OS/2

The first panel is known as **System Configuration**. Selecting the **OK** button, will automatically bring the next panel known as **OS/2 Setup and Installation**.

1. As per the objectives of this lab select the following features:

- Documentation - only the OS/2 Command Reference
- Tools and Games
- High performance File System
- REXX
- Optional Bit Maps

If you accidentally choose an unwanted option, clicking once with the mouse on that unwanted option will de-select it.

Question: What could happen if you did not select the **More** option for Documentation?

2. When your options are selected, press the **Install** button.

Follow the instructions on the screen. Accept the Source Directory to be the A drive. Click on the **Install...** button.

3. You will be requested for various OS/2 diskettes.

4. A message will appear indicating that the a shut down is required to make the changes active. Remember to perform a proper shut down.

Properly shutting down your system

1. Get the Desktop's contextual menu (also known as the pop-up menu)
2. Select the option **Shut down...**
3. Select **OK** at the warning message to close all windows and active programs.
4. The message **Please wait, shutdown still in progress** will appear. Wait.
5. A message will appear indicating that **Shutdown has completed. It is now safe to turn off the computer or to restart it.**
6. At that time, you may now turn your computer off.

You have now completed this lab.

LAB: Response File Installation

Installing OS/2 Using a Response File

Background

An alternative method of OS/2 system installation is Response File Installation. The response file allows for unattended installation of OS/2 from diskette, diskette images on an alternative medium (i.e., hard file), or from a code server on the Local Area Network.

Objectives

1. Review requirements for a response file.
2. Create a response file for unattended installation.
 - ✗ a. A "tester", OS/2 is to be placed in the **D** partition. You must be able to boot from this partition.
 - b. This "tester" must be available as an option from the Boot Manager menu.
 - ✓ c. A 10MB swap file is to be located in the **E** partition.
 - ✓ d. Provide ATM font support for "tester" and WIN-OS2 sessions.
 - e. Windows applications must start from the OS/2 Desktop.
 - ✓ f. The WIN-OS2 environment is to be located in the **E:** partition.
 - ✓ g. Provide the necessary drivers for an IBM 4029 LaserPrinter 10P.
 - ✓ h. This "tester" OS/2 is to have all features installed.
3. Add the newly created response file to Diskette number 1.

This will be supplied by your instructor.
4. Install OS/2 on the **D** partition using a response file.

Existing partitions cannot be deleted or resized. Any data in these partitions is to be preserved, and the OS/2 in the existing C: partition is not to be altered.

Before beginning the installation, planning must be done. Complete the following planning chart to determine the configuration of the hard disk.

Hand copy
Pack copy

Planning chart

Component Required	Disk Space Install	Disk: Primary or Logical
Existing OS/2 2.1	?	Primary
"Tester" OS/2	?	?
Swapper file	10 MB	?
WIN-OS/2	9MB	?
DOS 5	5MB	C:Primary
Boot Manager	1MB	Primary

Creating the response file

You will be using an unattended installation of OS/2 from a code Server. In order to do this, you will need to create a response file that contains all of the parameters that "tester" OS/2 requires for its' installation.

1. Open the USER.RSP file using the system editor.

The USER.RSP file is located in the OS2\INSTALL directory.

2. Modify this file by replacing the question marks with appropriate values.

Refer to your **Planning chart** (Appendix A) and **User Response Changes** (next page) as you complete the response file.

Appendix A contains a copy of the response file (SAMPLE.RSP), which includes a description of each parameter. Use this reference to complete the values in the response file listed on the next page. (This file is also located in you OS2\INSTALL directory).

3. When completing the reponse file, ensure that the following values are set:

- **ExitOnError=0**
- **RebootRequired=0**

4. Look up the desired printer code in the PRDESC.LST file in Appendix A.

This file is also located in you OS2\INSTALL directory.

5. To specify that the swap file is to go on other than your OS/2 boot drive, you must modify the ConfigSysLine parameter.

- **ConfigSysLine=E:\2048 10240**

6. Save the reponse file with your changes as:

- **C:\DEFAULT.RSP**

User Response File Changes

AlternateAdapter = 0
APM = 0
BaseFileSystem = ? Specify FAT File System
CDROM = 0
SCSI = Leave this parameter unchanged if present
CountryCode = 001
CountryKeyboard = US
DefaultPrinter = ? Use the number from PRDESC.LST (Appendix A)
DiagnosticAids = ? Install
DisplayAdapter = 0
Documentation = ? Install all
DOSSupport = ?
WIN-OS/2Support = ? All available groups and components
DPMI = 1
ExitOnError = 0
Fonts = ? All available fonts
FormatPartition = ? Do not format the partition
MigrateConfigFiles = 0
***MigrateApplications** = Leave this parameter unchanged if present
MoreBitmaps = ? Install more Bitmaps
Mouse = 1
MousePort = 0
OptionalFileSystem = ? Install optional file systems
OptionalSystemUtilities = ? Install all utilities
PCMCIA = 0
PrimaryCodePage = 1
PrinterPort = 1
ProcessEnvironment = 1
ProgressIndication = 1
RebootRequired = 0
REXX = ? Install REXX
SerialDeviceSupport = ? Install Serial Device Support

SourcePath = X:\IMG\OS2V21 *This will be A:\ if you are using diskettes.

TargetDrive = ? Target is the D partition

WIN-OS/2TargetDrive = ? Target is the E partition

ToolsAndGames = ? Install all

ConfigSysLine = ? Specify the location of the Swap file

Check your results with the completed response file located in Appendix A.

Setting up the appropriate diskette

- You will be given a diskette, by your instructor. Use it instead of the diskette number 1.
This new diskette #1, also known as the LT (LAN Transport) diskette, has been altered to permit access to a CID server.
 - The executable file (RSPINST.EXE), has been placed on this diskette as well as a modified Config.sys file.
RSPINST.EXE will read the information from your response file to get directions as to how to install OS/2.
1. Copy the C:\DEFAULT.RSP to the new diskette #1.
 2. Edit the config.sys to see that the proper response file is being used.

The actual installation of the Operating System

1. Boot the machine with your OS/2 Installation Diskette.
2. when prompted, insert the LT diskette #1 in drive A: and press <Enter>.
System installation will now begin.
Question: Did you receive any prompts during the system installation? Should you have?
 - a. When the installation is complete, reboot and test your setup.
 - b. Check to make sure that the components of OS/2 have been installed and located as you specified.You have now completed the lab.

LABS ON: OS/2 Internals

Memory Over-Commitment

Background

This exercise demonstrates how OS/2 2.x page addressing handles memory over-commitment. The growth as well as the decrease in the size of the swapper file will be shown.

Objective

Visualize that the SWAPPER.DAT file is a dynamic file.

Memory Over-Commitment

1. Start the program *swapmon2*

You will get a message that *Theseus0.dll can not be found*. That file is not required for the purpose of this lab.

2. Select the **OK** button to remove that message box.

The program SWAPMON2 will be started.

3. You will be asked to enter the profile path. Use **E:\labs**

This path is where the swapmon2 will store its' ini information.

4. Experiment with the options of this file.

5. Select the context menu of blue box of that program.

It will indicate the following options which you can experiment with.

- Settings
- Colors
- Drives
- Logging

6. The current size of the swapper file is 2048 K

7. Start committing memory.

You could start the following applications:

- OS/2 Chess - computer against computer
- Klondike - Solitaire - auto play
- Alarms
- Calendar
- Calculator
- Seek and Scan
- PMChart

8. The current size of the swapper file is now: 2048 K

9. End the applications you started for this program.

Depending on what else your system is doing, it may take some minutes before the swapper file decreases. If after a few minutes the swapper file has not decrease, end the program and restart it. Then wait a few more minutes.

10. The reduced size of the swapper file is: 2048K

Expected Results

After completing this lab, you should have seen the increase and decrease in the size of the swap file. Notice also, that is will always be in multiples of 512 KB.

Optional - Memory Allocation Lab

Background

This exercise focuses on the `DosAllocMem()` function and its usage. The sample program (`MEMLAB1.EXE`) used in the exercise allocates and uses memory in order to illustrate the memory allocation mechanism, particularly with regard to paged memory.

In this lab, the student is required to run the program ***MEMLAB1.EXE***. This program does the following:

1. It asks for "how many long integers should memory be allocated".

The application has to indicate to OS/2 how much memory it will require in total.

2. The system designates that amount of memory for the program to use whenever it requires it.

3. It will then ask "how many long integers to actually write to this memory".

When the application requires part of its' memory to actually do something with it, it has to tell OS/2 how much of it's total allocated it requires at that moment of time.

4. It performs writes to fill the requested amount of memory.

The message will be indicate "inserting integers into memory". At time, OS/2 will verify that the amount of memory the application is asking to use is actually within the limit of the amount of memory it told OS/2 it would require.

5. Checks whether the written values are correct.

If it had no problem it will indicate "All memory checked out OK, xxx integers successfully inserted into memory"

6. Frees the allocated memory.
-

Objective

To explain how the operating system manages the memory based on the different requests from programs.

Normal Memory Allocation

1. Open an *OS/2 window*.
2. Get to the *Labs* sub-directory.
3. Execute the program *MEMLAB1.EXE*.
4. When prompted *For how many long integers should memory be allocated*, type a **1** to allocate one double-word

The way this program has been written, the program will request one double-word (which is four bytes) of memory.

A 4KB page will actually be given by OS/2. (1 KB is 1024 bytes and 4KB is 4096 bytes)

5. When prompted *how many long integers should be written into this memory*, type **1024**.

The way this program has been written, it will request 4096 bytes of memory.

The program should execute without error. Explain how this is possible.

QUESTION: Why is it possible to access more memory during the write operation (in step 5) than was originally allocated (in step 4)? This is explained on the next page.

Memory Protection Violation

1. Open an *OS/2 window*.
2. Get to the *Labs* sub-directory.
3. Execute the program *MEMLAB1.EXE*.
4. When prompted *For how many long integers should memory be allocated*, type a **1** to allocate one double-word

The way this program has been written, the program will request one double-word (which is four bytes) of memory.

A 4KB page will actually be given by OS/2. (1 KB is 1024 bytes and 4KB is 4096 bytes)

5. When prompted *how many long integers should be written into this memory*, type **1025**.

The way this program has been written, it will request 4100 bytes of memory.

QUESTION: Why was an error given in this part of the lab? This is explained on the next page.

Large Memory Allocation

1. Open an *OS/2 window*.
2. Get to the *Labs* sub-directory.
3. Execute the program *MEMLAB1.EXE*.
4. At the first prompt, type **1048576** to allocate 4 MB (4,194,304 bytes) of memory.
5. At the second prompt, type **67584** to specify 264 KB (270,336 bytes) of memory.

Notice that you have requested a read/write action to an area well beyond the traditional 64KB (segment) boundary of OS/2 1.x.

Note

This may take a few seconds to complete.

If the program does not work, check for the free disk size on the logical drive where the SWAPPER.DAT file is located. Erase excess files and rerun the program. Contact your instructor for further assistance.

Expected Results

After successfully completing the lab, you will have tried a number of memory allocation options available that utilize the `DosAllocMem()` function. The results from each step are explained below.

Part 1

You will notice that the operating system always allocates at least 4KB (1 page) of memory, even though only 40 bytes were specified in the allocation request. The same would have been true if only 1 byte had been requested. This is because all memory management in OS/2 Version 2.x is handled on a per-page basis, using the flat memory model. The page is the lowest level of granularity in OS/2 2.x. This granularity differs from previous versions of OS/2, which had a byte-level granularity.

This principle applies to *all* memory protection and memory access types (read, write, etc.) operations. A programmer is able to use more memory for read/write than is actually requested in an allocation request if that memory exists within the page boundary.

CNTL ALT NUMLOCK NUMLOCK,
invokes dump.

dump's data
to file

Part 2

In part (2), a general protection exception occurs when attempting to access 4100 integers for read/write. Attempting to access 4100 integers resulted in a general protection exception (Trap 000D) when an access was attempted on the 4100th location. This program will indicate "message file not found".

Part 3

The flat memory model allows a programmer to address any location within a memory object on a contiguous basis. The DosAllocMem() function returns a 32-bit memory address, and *not* a segment selector as in previous versions of OS/2. The programmer, therefore, does not have to consider the 64KB segment boundary limitation. This freedom allows the programmer to allocate a memory object of an arbitrary size up to 512MB (the process space address limit).

Optional - Memory Protection Lab

Background

This exercise demonstrates the different memory attributes used for different types of memory access and the resulting impact on the allowed usage of memory (read or write). The access types used in this lab exercise are READ, WRITE, and EXECUTE.

In this exercise, the student will be given a chance to see how dynamic the operating system is in dealing with an executing program.

Program **Memlab2.exe** does the following:

1. It asks for the amount of memory (in KB) you wish to allocate
2. It asks for the type of allocation you want that memory to have
 - a. PAG_COMMIT and PAG_READ
Means to commit the page to memory and allow it only to be read from.
 - b. PAG_COMMIT and PAG_WRITE
Means to commit the page to memory and allow it only to be written to.
 - c. PAG_COMMIT and PAG_EXECUTE
Means to commit the page to memory and allow it only to be executed.
 - d. PAG_COMMIT and PAG_GUARD and PAG_WRITE
Means to commit the page to memory, place a page guard and allow it to be written to.
 - e. PAG_WRITE
Means to write to the page.
3. You will be asked to enter your selection (from 1 to 5).
4. It will allocate the memory (in pages) with the requested attributes.
5. It will ask you if you now want to READ (R) or WRITE (w) to these pages.
6. It will either read or write to the pages. Depending on the selection you made it will result in either an error or not.

For instance, if you had indicated the pages were for reading and you attempted to write to it, the result would be an error.

Objective

Demonstrate that the operating system does provide memory protection on a continuous basis.

Attempting to read on a PAG_COMMIT and a PAG_READ

Open an *OS/2 Window* and carry out the following steps:

1. Get to the *Labs* sub-directory.
2. Execute *MEMLAB2*.
3. Type the amount of memory (in KB) that you would like to allocate.
4. Type **1** for PAG_COMMIT and PAG_READ.
5. Type **r** for read.

Expected Result

The READ request, on a PAG_READ committed page, will execute without a problem.

Attempting to write on a PAG_COMMIT and a PAG_READ

Using the same *OS/2 Window*,

1. Execute *MEMLAB2*.
2. Type the amount of memory (in KB) that you would like to allocate.
3. Type **1** for PAG_COMMIT and PAG_READ.
4. Type **w** for write.

Expected Result

The WRITE request, on a PAG_READ committed page, will issue an error.

Attempting to read on a PAG_COMMIT and a PAG_WRITE

Using the same *OS/2 Window*,

1. Execute *MEMLAB2*.
2. Type the amount of memory (in KB) that you would like to allocate.
3. Type **2** for PAG_COMMIT and PAG_WRITE.
4. Type **r** for read.

Expected Result

The READ request, on a PAG_WRITE committed page, will execute without a problem.

Attempting to write on a PAG_COMMIT and a PAG_WRITE

Using the same *OS/2 Window*,

1. Execute *MEMLAB2*.
2. Type the amount of memory (in KB) that you would like to allocate.
3. Type **2** for PAG_COMMIT and PAG_WRITE.
4. Type **w** for write.

Expected Result

The WRITE request, on a PAG_WRITE committed page, will execute without a problem.

Attempting to read on a PAG_COMMIT and a PAG_EXECUTE

Using the same *OS/2 Window*,

1. Execute *MEMLAB2*.
2. Type the amount of memory (in KB) that you would like to allocate.
3. Type **3** for PAG_COMMIT and PAG_EXECUTE.
4. Type **r** for read.

Expected Result

The READ request, on a PAG_EXECUTE committed page, will execute without a problem.

Attempting to write on a PAG_COMMIT and a PAG_EXECUTE

Using the same *OS/2 Window*,

1. Execute *MEMLAB2*.
2. Type the amount of memory (in KB) that you would like to allocate.
3. Type **3** for PAG_COMMIT and PAG_EXECUTE.
4. Type **w** for write.

Expected Result

The WRITE request, on a PAG_EXECUTE committed page, fails. This behavior occurs because the 80386 processor does not distinguish between READ and EXECUTE access.

Furthermore, READ or EXECUTE access is allowed even for memory objects specified with WRITE access. WRITE access implies both READ and EXECUTE access. WRITE access, however, must be specified in order for the application to write into a memory object.

LABS ON: OS/2 Internals (second group)

Dynamic Linking Lab

Background

Application programs may be making use of dynamic linking. It can not be assumed that these applications will automatically update the OS/2 configuration file to denote this. The **LIBPATH** parameter indicates to OS/2 the location of existing Dynamic libraries.

The program you will be using is called **BIGBEN**. This is an OS/2 Full screen application that indicates the current time in a digital format.

Programmers can also create programs that use the same executable file but depending on the parameter, entered with that exe, will result in different DLL(s) being used. Either case, the path to these DLLs must be in the **LIBPATH** statement.

Objectives

- Verify that a program using dynamic linking will not run if its' path to the dynamic module is not given.
 - Copy a specified dynamic module into an existing DLL directory.
 - Change the **LIBPATH** statement to show the path of a new dynamic module.
 - Verify the effect of changing the **LIBPATH** parameter.
-

Working with the Default LIBPATH parameter

1. Open an **OS/2 Full screen**
2. Go to **Labs**
3. Run **BigBen**

Notice the error message that you receive. This occurs because the program BigBen requires a dynamic link module which it can't find.

Copying a dynamic module into an existing Dynamic Module directory

The DLL the program requires is **Crtlib.dll**

1. The default OS/2 DLL sub-directory is **D:\OS2\DLL**
2. Search for the location of the **CRTLIB.DLL** file.
3. When found, copy it to the **\OS2\DLL** sub-directory.
4. Start **BigBen**
5. Close the **BigBen** program.
Notice from the **config.sys** file that the **OS2\DLL** is in the **LIBPATH** path.
6. Carefully erase the **Crtlib.dll** file from **OS2\DLL**.
Be extremely careful not to erase any other DLLs!

Updating the path of the LIBPATH statement

Your **Config.sys** file has **E:\LABS** in the **LIBPATH** path.

1. Review your **config.sys** file to ensure that the **E:\LABS** is in the **LIBPATH** path. Edit it if the path is not shown.
2. Copy the **CRTLIB.DLL** file to **E:\LABS**
3. Open an OS/2 full screen.
4. Start **BigBen**.
5. Close the **BigBen** program.

Using the same executable file but different DLLs

1. Open an OS/2 window.
2. Go to the **LABS** directory.
3. Start the program **res**
4. Select **English**.
The messages should now appear in English.
5. Go to its action bar and select **File**
6. Select **Exit** from its pull down menu.
7. End that application.
8. Start the program **res** again.

9. Select French.

The messages should now appear in French.

10. Go to its action bar and select **Fichier**

11. Select **Sortir** from its pull down menu.

Notice that this SAME executable file is now giving its' messages in French, where as in the other, the messages are in English.

12. Exit out of the application.

With this method, programmers are now able to create one executable file. 'Language sensitivity' can be placed in DLLs.

Multithreading Lab

Background

Multithreading programs use multiple threads to perform different tasks simultaneously. Since the processor cannot actually perform more than one task at a time, the processor is "shared" between tasks.

The Priority program allows you to start two threads.

Objective

- Demonstrate the capabilities and effects of multithreading.
-

A program that has two threads

1. Go to the **Labs** directory.
2. Run the program **Priority**
3. Select **Thread** from the action bar of the **Priority** program.
4. Select **Start 1** from that pull down menu.
This will be one thread.
5. Move the **Thread #1** window over to the right of your screen.
6. Once more, select **Thread** from the action bar of the **Priority** program.
7. Select **Start 2** from that pull down menu.
This will be another thread.
Do **not** set both threads to time critical. The way this program has be written, this may end up 'hogging' the system.
8. Experiment keeping both threads at **Regular** and changing only the Priority Level. Notice when the Priority boost occurs.
Do **not** set both threads to time critical. The way this program has be written, this may end up 'hogging' the system.
9. When you are finished, close the program.

Multithreading Lab

Background

This exercise demonstrates that an OS/2 program can start a very large number of threads (up to a maximum of 4095). This program uses the `DosExecPgm()` function.

In this exercise, the student will run the program *memlab4.exe*. The syntax of the program is:

- `MEMLAB4 <number of threads>`

Where **number of threads** is the number of threads that will be started for this process.

Depending on how many threads your system can support, the following message may be generated: **_beginthread error**. This typically occurs when the maximum allowed number of threads specified in `CONFIG.SYS` is less than the number of threads requested.

Note

Remember that the specified number of threads in the `CONFIG.SYS` file includes threads used by the operating system.

Objective

Demonstrate what occurs when one changes the `THREADS` parameter in the `config.sys` file to a number smaller than is required by the operating system.

Running the program MEMLAB4

1. Open an OS/2 Window
2. Go to the **Labs** sub-directory.
3. Execute MEMLAB4 and indicate a thread number.
 - Type *memlab4 10* and then press the <Enter> key.
4. Experiment using different thread values, such as 300.

If the required number of threads cannot be started, check the `CONFIG.SYS` file. This file contains a specification for the maximum number of threads allowed in the system.

Changing the Thread value in Config.sys

1. Change the Thread = value to 70 in the OS/2 Config.sys file.
2. Properly shut down the system and then re-start OS/2.
3. Go to the **Labs** sub-directory.
4. Execute MEMLAB4 and indicate a thread number.
 - Type **memlab4 35** and then press the <Enter> key
5. Experiment using different thread values
6. When you have finished, place the Thread value back to what it originally was before you started this lab.
7. Properly shut down the system and then re-start the system.

Expected Results

After successfully completing the exercise, you should have observed that the number of threads **per process** is not limited to 53 as it is in OS/2 1.x. The maximum number of threads **system wide** is now 4095 threads.

LAB ON: HPFS

Creation of a High Performance File System Partition

Background

In situations where fixed-disk drives are expected to contain large files accessed by OS/2 applications, a user can install the HPFS. It supports as many as 16 partitions, with each partition supporting up to 2 Gigabytes.

The HPFS maintains compatibility with the FAT system at the API level. It is less performance sensitive than the FAT system when working with large files and directories and can also handle files that use the traditional or new long file names.

Objective

- To set up the **G** partition as an HPFS partition.
-

Verifying the HPFS Configuration file parameter

1. Ensure that the following statement is in config.sys:

- **IFS=D:\OS2\HPFS.IFS /CACHE:64**

Note: If this statement is not there, chances are that you have not installed the files to support HPFS. In that case, do a Selective Installed and install HPFS support.

2. Run the CACHE.exe program from an OS/2 command prompt.

- **cache.exe**

Question: What are the default parameters?

1000, 5000, 500, 1000

Question: What is another way of altering these values without going to the command prompt?

CONFIG.SYS

Formatting G as HPFS

1. On an OS/2 command prompt enter the following statement:

- **FORMAT G:/FS:HPFS**

Note: You will be asked if you are sure you want to format. Formatting a partition will erase all the contents of that partition.

2. **Note:** You will be asked for a volume label. If you press the <Enter> key it will indicate no volume label, please select no volume label.

The HPFS file system will now be using that partition.

Question: What would be another way of formatting this partition as HPFS?

cut start=1

USING the High Performance File System Partition

Background

Because DOS only understands the FAT file system naming convention, you will NOT be able to see some of the files that are in G. The DOS prompt will be able to see the HPFS partition thus, you can work within HPFS.

As long as the file name is within the FAT file system naming convention you will be able to work with the file even if it is in an HPFS partition.

Objectives

- To create files with the long file names and to see how non-HPFS partitions deal with the long file names.

Creating a long file name and viewing it

1. Copy and rename the following files to G
 - *Copy d:\config.sys g:\config.new*
 - *Copy d:\config.sys g:\"configurationfile from Dp"*
 - *Copy d:\config.sys g:\"configurationfile from Dd"*
 - *Copy d:\config.sys g:\configurationfilep*
 - *Copy d:\config.sys g:\configurationfiled*
 - *Copy d:\autoexec.bat g:\autoexec.new*
 - *Copy d:\autoexec.bat g:\"autoexecution from DOSp"*
 - *Copy d:\autoexec.bat g:\"autoexecution from DOSd"*
 - *Copy d:\autoexec.bat g:\autoexec.batttpp*
 - *Copy d:\autoexec.bat g:\autoexec.battttd*
2. From an **OS/2** command prompt do a **Dir** of G:
 - Type **G:**
 - Type **Dir**
3. From an **DOS** command prompt do a **Dir** of G:
 - Type **G:**
 - Type **Dir**

Question: What is different between the DOS and the OS/2 command prompt?

Copying a long file name from an HPFS partition to a FAT partition

1. From an OS/2 command prompt, copy *configurationfilep* from **G:** to **E:**
2. From an OS/2 command prompt, copy *autoexecution from DOSp* from **G:** to **E:**
3. From an OS/2 command prompt, copy *autoexec.battttp* from **G:** to **E:**

Question: Did you receive any errors? How did you resolve them?

1. Open the Drives folder
2. Open the **E** and the **G**
3. Copy *configurationfiled* from **G:** to **E:**
4. Copy *autoexecution from DOSd* from **G:** to **E:**
5. Copy *autoexec.batttd* from **G:** to **E:**

Question: Did you receive any errors?

1. Perform a **DIR** on **E:**
Question: Notice any new files?
2. Return to the **E** drives object and open its' **Details view**

Question: Notice the real names of those new files. What does this signify?

Doing a BACKUP from an HPFS partition to a FAT partition

1. From an OS/2 command prompt use the BACKUP command to backup the root of **G**
 - a. Type **Backup G:\ A:**
 - b. Follow the instructions.
Notice You will be able to see the files that are being backed up. These files will include the long file name files.
2. From an OS/2 command prompt RESTORE the backed up files to **D**
 - a. Type **Restore A: D:**
 - b. Follow the instructions.

Notice You will get an error indicating that a file name or extension is too long and that file will not be restored.

Only the files that follow the FAT naming convention will get restored to the **D** drive.

3. Erase the files that you created on **G**

4. Type **Restore A: G:**
5. Follow the instructions.

LABS ON: VDMs

VDM Configuration

Background

Recall that each object (including program objects) has settings. Settings are properties or characteristics of an object. In the settings of a program object, one of the tabs shown will be **Session**. The SESSION tab indicates the program type for a program object. The type determines how the program runs. The Types are classified as either OS/2 Full Screen, OS/2 Window, DOS Full Screen, DOS Window, WIN-OS2 Full Screen or, WIN-OS2 Window.

When the program object is a DOS program, a push button in the Session page will be accessible. Selecting that push button will give the user a chance to tailor how they wish the VDM to execute.

Objective

In this exercise, the student will create a new folder and configure a VDM within that folder according to their specified parameters.

Working with certain settings

1. Make a copy of the **autoexec.bat** (file found on D), and place a copy of it in the **Labs** sub-directory.
2. Alter that copied autoexec.bat file to have the prompt command as indicated:

```
PROMPT ←-[33;41m[DOS] $i$P$G
```

To get the arrow, use the keypad numbers. Press down on the **Alt** key and the numbers **2 7**

3. Create a folder and call it **Test**
4. Make a copy of a DOS Full Screen and a DOS window and place them in the Folder - TEST.
5. Working within the **Test** folder, open the **DOS settings** of the **DOS Window**
6. Alter the following;
 - **DOS_AUTOEXEC** to point to E:\LABS\autoexec.bat
 - **DOS_Device** add the following path
D:\OS2\MDOS\ANSI.SYS
 - **DOS_RMSIZE** to some other value than the default.
 - **EMS_Memory_limit** to its maximum
 - **XMS_Memory_limit** to 0
7. **Save** these changes.
8. Change this DOS Window name to the name **My Window**
9. Start **My Window**

You should get a command prompt that looks like this:

- DOS D:\>

The text will be yellow and it will be in a red block.

10. Type in the *mem* command and verify how much EMS and XMS is available for this VDM.
11. Close *My Window*
 - Type *Exit*

Changing the settings

1. Working within the *Test* folder, open the **DOS settings** of *My Window*.
2. Alter the following:
 - **DOS_AUTOEXEC** to point to E:\LABS\autoexec.bat
 - **DOS_Device** DELETE the following path
D:\OS2\MDOS\ANSI.SYS
 - **DOS_RMSIZE** to 640KB
 - **EMS_Memory_limit** to 1024
 - **XMS_Memory_limit** to 2948
3. **Save** these changes.
4. Start *My Window*.

You should get a different looking command prompt than before.
5. Type in the *mem* command and verify how much EMS and XMS is available for this VDM.
6. Close *My Window*.

Expected Results

After successfully completing the exercise, check the result by double-clicking on *My Window* in the folder *Test*. A VDM should start with a DOS command prompt, which should look like the following example.

```
←[33;41m[DOS] D:\>
```

The 33 and 41 represent colors. To be able to see these colors, you have to load the device driver ANSI.SYS. When the DOS prompt looks like this, the ANSI.SYS is NOT active.

Rebooting a Virtual DOS Machine

Background

When running DOS, if an INT 19h is called, the result is a reboot of the **entire system**. This lab will show that the execution of an INT 19h in a VDM is handled by the Virtual DOS Machine Manager (VDMM).

Objective

The objective here is to show what occurs in OS/2 Version 2.x when a program running in a VDM issues an INT 19h.

Issuing an INT19h

1. Open a DOS Window.
2. Go to the *Labs* sub-directory.
3. Run the program **INT19**
4. When prompted, press the <Enter> key to issue the INT 19h.

Consider what has happened. Although the input would normally cause the system to reboot, now only that VDM is terminated.

Expected Results

After you successfully completed the exercise, note that interrupt INT 19h did not reboot the system. Instead, the interrupt was routed to the Virtual DOS Machine Manager (VDMM) by the General Protection Handler. The VDMM terminated the VDM when receiving the INT 19h.

If you start a DOS application program from an OS/2 command prompt, control is passed to the Virtual DOS Machine Manager which then starts the VDM. Execution of INT 19h does **NOT** terminate OS/2. Instead, INT 19h terminates the VDM session only.

MVDM Cut and Paste

Background

One can share information between sessions. One can copy or cut information from one session and then paste the same information to different session (or the same session).

OS/2 2.x accesses two clipboards:

1. The OS/2 Clipboard accepts information from OS/2 window sessions, PM sessions, DOS sessions, one of more programs in a WIN-OS2 session, or any combination of these.
2. The WIN-OS2 Clipboard exchanges data between WIN-OS2 sessions.

One can make the Clipboard *private* if one wishes to prevent programs in WIN-OS2 from copying, cutting, or pasting from the OS/2 Clipboard. If you make the Clipboard private but leave the WIN-OS2 Clipboard *public* you can copy , paste, cut between different WIN-OS2 sessions.

If one makes the WIN-OS/2 Clipboard private, only programs within that session can copy, cut, or paste from that sessions' private WIN-OS2 clipboard.

You can always exchange information if either Clipboard is private as long as you click on the IMPORT or EXPORT choices to transfer information between clipboards.

Objectives

1. Fill a VDM window session with text, copy it and, paste the contents into another program.
2. Place a graphic in a VDM session, copy part of it and, paste the contents into another program.

Using the COPY ALL

1. Start a DOS window.
2. Go to the **Labs** sub-directory.
3. Put some text in this VDM, for example, use the DIR /W command.
4. Get to the Context menu of that window (the title bar icon).
5. Select the option **Copy All**
6. Open the **EPM** program.
7. From it's Action Bar select **Edit**
8. From the pull down menu of EDIT, select **Paste**

This will paste the text that you copied from the VDM window into the OS/2 program.

9. Close EPM without saving the information.

Copying and Pasting Graphics

1. Start the DOS window.
2. Go to the **Labs** sub-directory.
3. Start the program **Graphic**
4. Get to the Context menu of that window (the title bar icon).
5. Select the option **Mark**
6. Use your mouse to select the "area" that you wished marked.
7. Return to the Context menu of that window (the title bar icon)
8. Select the option **Copy**
9. Start the OS/2 program **PMCHART**
 - This program can be found in the Productivity folder.
10. From it's Action Bar select **Edit**
11. From the pull down menu of EDIT, select **Paste**
 - It may take a few moments to show the results. The results are the graphics that you marked and copied from the VDM window.
12. Close PMCHART without saving the information.

VDM Interprocess Communications

Background

In this exercise, you are required to start an OS/2 application program that first creates a number of named pipes. The OS/2 application then waits for a DOS BASIC program to connect to the pipe. This connection is performed by one thread. Afterward, the main OS/2 program sends data to change the screen colors of various (connected) DOS BASIC programs.

Objective

The objective of this lab is to show that an OS/2 session can exchange data with a DOS session in the same system.

Steps

1. Start an OS/2 Window.
2. Go to the **Labs** sub-directory.
3. Start the program PIPEOS2 using a parameter of 1.
 - Type *pipeos2 1*
4. Start a DOS window.
5. Go to the **Labs** sub-directory.
6. Type **BASICA PIPEDOS**
7. Position both windows such that you can see parts of both windows.
8. Return to the OS/2 program.
9. Enter a letter that corresponds to a color you want the DOS window to display (this is done from the OS/2 program).

For instance, type the letter **B** for a blue screen, or **G** for a green screen.
10. Inside the DOS window, you should see it changed to the color you indicated via the OS/2 program.
11. Try other colors.
12. To end the DOS program, type the letter **Q** from the OS/2 program.

This will end the DOS program even though the screen will maintain the same color as was last requested.
13. To end the OS/2 program, press the <Enter> key.

This will provide the command prompt.

Expected Results

The DOS session should have been able to connect to a single or many named pipe(s) that were created and maintained by the OS/2 session. Afterward, data was passed to the DOS session(s) in the form of single characters that altered the color of the DOS screen.

After you have successfully completed the exercise, please note that this is only one way of communicating between DOS and OS/2 sessions.

Booting Native DOS while under OS/2

Background

An important goal of OS/2 2.x is the ability to run past, current, and future PC DOS programs. Most DOS applications available today run unchanged in the VDM DOS emulation environment. It should be remembered that the DOS which runs in this case is highly optimized to an OS/2 2.x virtual 8086 machine. Because of this there are subtle differences between DOS Emulation and the real DOS. For instance, some DOS applications may be inherently bound to a specific DOS version, relying on internal DOS structures or features not present in the VDM DOS Emulation, such as internal DOS tables or undocumented features. Another restriction of VDM DOS Emulation is that only DOS character device drivers can be loaded. The user may own a BLOCK Device for which no OS/2 driver is available.

Virtual Machine Boot allows the user to boot "off the shelf" DOS 3.x, 4.0 or 5.0 in a VDM session, including DOS Block Device Drivers. This gives the user the greatest possible compatibility with PC DOS.

Objectives

In this exercise, the student will create a new **DOS Window** object in the command prompts folder. This new DOS session will be configured so as to boot a shrink wrap version of DOS 5.0 instead of utilizing OS/2s' emulated version of DOS.

In order to boot an 8086 kernel into a VDM, that kernel's boot record must be obtained from either a bootable diskette, an image file of that diskette, or a DOS hard disk partition.

The student will be required to configure a VDM which can, in turn, boot DOS from any of these sources.

Running the VER command from a DOS window

1. Open an DOS window.
2. Type **Ver**

The answer should be "The Operating System/2 version is 2.10".

Setting up a VDM to boot from a DOS partition

1. Make a copy of a **DOS Window** and call it *My DOS 5 partition from C:*
2. Get into the Settings of **My DOS 5 partition from C:**
3. Go to **DOS Settings**
4. Search for the option called **DOS_Startup_Drive**
5. Type **C:**
6. Save your changes.
7. Start "My DOS 5 partition from C:".
There will be errors regarding memory. Ignore them for now.
8. From the DOS Shell, press the <F10> key.
9. Highlight the **File** on that Action Bar.
10. Select **Exit** from the pull down menu.
11. Verify that you have indeed started from C: by typing **Ver**
The response should be "IBM DOS Version 5.00"

Avoiding multiple config.sys and autoexec.bat files on C:

Loading DOS from a DOS partition presents one significant problem. It might appear that the user would have to maintain multiple configuration files and rename or copy them depending upon whether the partition was being booted into a VMB session or directly from Boot Manager. The key is to specify both sets of drivers in the correct order in the CONFIG.SYS and AUTOEXEC.BAT located in C:

1. Using an OS/2 Editor, edit the CONFIG.SYS file situated on the C:\ drive and add the following lines **after** the native DOS HIMEM.YS and EMM386.EXE found in the CONFIG.SYS file. It should appear something like this:
 - REM native DOS drivers first
 - DEVICE = C:\DOS\HIMEM.SYS
 - DEVICE = C:\DOS\EMM386.EXE
 - REM OS/2 Version 2.x drivers second
 - DEVICE = D:\OS2\MDOS\HIMEM.SYS
 - DEVICE = D:\OS2\MDOS\EMM386.SYS
 - ... etc ...
2. Ensure that the paths are pointing to the proper locations.

These statements may not all appear in your config.sys file. It depends on what memory has been set up for native DOS situated in C:

When this file is processed in an OS/2 VMB, the DOS HIMEM load fails due to no available extended memory. EMM386 fails as it sees protect-mode memory software already running. Then, the OS/2 Version 2.1 HIMEM and EMM386 stub device drivers load as normal. So you can expect to see at least one error message, indicating that the DOS drivers are not loaded.

When this file is processed as part of a native DOS boot, the DOS HIMEM and EMM386 load as normal, but the OS/2 stub device drivers detect that they are not running under OS/2 and do not load themselves. So you can expect to see at least one error message, indicating that the OS/2 drivers are not loaded.

3. Using an OS/2 Editor, edit the AUTOEXEC.BAT file situated on the C:\ drive and add the following **before** the native DOS mouse driver.
 - REM load the stub mouse driver from OS/2 Version 2.1 first
 - LH D:\OS2\MDOS\MOUSE
 - REM load the stub mouse driver from native DOS second C:\DOS\MOUSE

Ensure that the paths are pointing to the proper locations.

Note that here the OS/2 driver is listed first. When this file is processed in an OS/2 VMB session, the OS/2 stub loads first. The DOS mouse driver sees that another mouse is already present and hence does not install itself.

When booting DOS natively, the OS/2 mouse stub device driver detects that it is not running under OS/2 and does not load itself. The DOS mouse driver then loads as normal.

Preparing for a diskette boot

1. Boot to the DOS 5 Operating system using the Boot Manager menu.
2. Place a diskette in the diskette drive and format it. Ensure that the hidden files are also placed on that diskette.
 - Format a: /s
3. **Copy** the following files *to the diskette*
 - **Copy C:\Config.sys A:**
 - **Copy C:\Autoexec.bat A:**
 - **Copy D:\OS2\MDOS\FSFILTER.SYS A:**
 - **Copy C:\DOS\him*. * A:**
 - **Copy C:\DOS\set*. * A:**
 - **Copy C:\DOS\emm*. * A:**
 - **Copy C:\DOS\mou*. * A:**
 - **Copy C:\DOS\edi*. * A:**
 - **Copy C:\DOS\doss*. * A:**
 - **Copy C:\DOS\comman*. * A:**

This will copy the files that this lab might require.

4. Remove that diskette from the diskette drive.
5. Start OS/2.
6. Place the diskette back into the diskette drive and start an OS/2 editor.

You will be editing the files from the diskette NOT the files from the OS/2 D: partition.

7. Add a new line in the **A:\CONFIG.SYS** file prior to the first DEVICE = statement.
8. On this line type **DEVICE = A:\FSFILTER.SYS**

FSFILTER is a device driver that provides access to OS/2 disk partitions when running a specific version of DOS. This statement must precede any statements with a reference to a file stored in an HPFS partition.

9. Ensure that any commands referred to from this file are pointing to A:\
10. Save those changes.
11. Edit the **A:\autoexec.bat** file and ensure that all commands are pointing to A:\

Setting up the VDM to boot from a DOS diskette

1. Make a copy of a **DOS Window** and call it *My DOS 5 diskette*
2. Get into the Settings of **My DOS 5 diskette**
3. Select **DOS Settings**
4. Search for the option called **DOS_Startup_Drive**
5. Type in **a:**
6. Save your changes.

Booting from the diskette and using it

1. Place your bootable diskette in the diskette drive.
2. Start **My DOS 5 diskette**

This will take a few moments. The hidden files from the diskette will be used to boot that VDM session. If all goes well, what you should see is the DOS 5 shell you set up.

3. Verify that you have indeed started from A: by typing **Ver**

The response should be "IBM DOS Version 5.00" and you should have seen the diskette drive light go on.

Creating an image copy of that diskette

1. Start an OS/2 window.
2. Make a directory and call it *images*
3. Type **VMDISK A: D:\IMAGES\DOS5.IMG**

The image file created will be a complete binary "dump" of the diskette. Its file size will correspond to the diskette capacity regardless of the amount of space actually used on the diskette.

Setting up the VDM to boot from a DOS image

1. Make a copy of a **DOS Window** and call it *My DOS 5 image*
2. Get into the Settings of **My DOS 5 image**
3. Select **DOS Settings**
4. Search for the option named **DOS_Startup_Drive**
5. Type **D:\images\dos5.img**
6. Save your changes.

Booting from the DOS image

1. Start **My DOS 5 image**

Notice that this will take less time than booting from the diskette.

Results

This completes the lab. You should have been able to:

- Boot DOS from the C: partition while OS/2 was active.
- Boot DOS from a diskette while OS/2 was active.
- Boot DOS from an image while OS/2 was active.

LABS ON: WIN-OS2

Defining WIN-OS2 Applications in a VDM

Background

This exercise focuses on SAVDM and MVDM. The sample program Fpamort.exe used is an application developed for use on a DOS/Windows platform. You will be asked to create Workplace Shell objects and then configure some of them for use as a SAVDM and others as MVDM.

- SAVDM - Single application in a virtual DOS Machine
- MAVDM - Multiple applications in a virtual DOS Machine

Objectives

1. Create two program reference objects for the program FPAMORT.EXE
 2. Using one of the created objects, set the program settings for Fpamort.exe and use as a SAVDM
 3. Using the other created object, set the program settings for Fpamort.exe and use as a seamless SAVDM
 4. Create a program reference object that references WINOS2.COM and configure it to MVADM
-

Defining a Single Application VDM (SAVDM)

1. Open the **Templates** folder.
2. Drag a copy of the **Program** object and place it on the Desktop.
3. On the **Program** page, type the path to the FPAMORT.EXE file.
This program is in LABS.
4. Go to the **Session** page.
5. Select the **WIN-OS/2 Full Screen** radio button.
6. Name the program - Loan Repayment.
7. Close the **Program-Settings** notepad.
8. Start the program.

The intent of this section is to set up a SAVDM. If you were successful, you will be able to start the program and it will be in a WIN-OS2 full screen. It will also be the only program in that session.

Defining a "Seamless" WIN-OS2 VDM

1. Open the **Templates** folder.
2. Drag a copy of the **Program** object and place it on the Desktop.
3. On the **Program** page, type the path to Fpamort.exe.
4. Turn to the **Session** page.
5. Select the **WIN-OS/2 window** radio button.
6. Name the program - Loan Repayment.
7. Close the **Program-Settings** notepad.
8. Start the program.

The intent of this section is to set up a Seamless VDM. If you were successful, the program will be right on the Desktop.

Defining a Multiple Application VDM (MAVDM)

1. Open the **Templates** folder.
2. Drag a copy of the **Program** object and place it on the Desktop.
3. On the **Program** page, type the path to the WINOS2.COM file.
Remember that WIN-OS2 support has been installed on the **E:** drive.
4. Tab down to the **Parameters:** field.

In this field, you will enter (as parameters to WINOS2.COM) the full paths of the Windows programs you would like running when this MAVDM is started.

- Type **/S drive1:\path1\program1.exe , drive2:\path2\program2.exe**

/S Indicates Standard Mode.

drive1: The drive letter where the EXE file is located.

\path1\program1.exe ie. **E:\LABS\FPAMORT.EXE** The full path to the executable file fpamort.exe

drive2: The drive letter where the second EXE file is located.

\path2\program2.exe ie. **E:\OS2\MDOS\WINOS2\CLOCK.EXE** The full path to the second executable file clock.exe

5. Go to the **Session** page and select **DOS full screen**
6. Name the program - Loan Repayment and Clock in an MAVDM
7. Close the **Program-Settings** notepad.
8. Start the program.

The intent of this section is to set up multiple windows application in one VDM. If you were successful, you will see the Clock and the Loan Repayment programs started in that VDM.

Starting OS/2 and DOS from WIN-OS2 Applications

Background

In this version of OS/2, you can start DOS and OS/2 applications from a WIN-OS2 session.

Objectives

1. Start an OS/2 application from a WIN-OS2 File Manager
 2. Start an DOS application from a WIN-OS2 File Manager
-

Setting the stage

1. Create a WIN-OS2 File Manager program reference
 - Open the **Templates** folder.
 - Drag a copy of the **Program** object and place it on the Desktop.
 - On the **Program** page, type the path, E:\OS2\MDOS\WINOS2\WINFILE.EXE.
 - Tab down to the **Working Directory:** field.
 - At this field, enter E:\OS2\MDOS\WINOS2.
 - Name the program WIN-OS2 File Manager.
2. Start WIN-OS2 File Manager.
3. View the files in **D:\OS2\APPS**
4. Find Klondike.exe.
5. Start Klondike.exe.
6. Get to the Window List.

Notice that Klondike Solitaire appears separate from the WIN-OS2 file Manager.
7. Close the program Klondike.
8. View the files in **E:\LABS**
9. Find Graphic.exe.
10. Start Graphic.exe.
11. Get to the Window List.

Notice that Graphic appears separate from the WIN-OS2 file Manager.
12. Close the program Graphic.

Expected Results

If you were able to complete the steps above, then you have successfully demonstrated OS/2s' support for launching OS/2 and DOS applications from a Windows application.

Automatically starting a WIN-OS2 Application from the group

Background

In this version of OS/2, when you start a WIN-OS2 session you can indicate what programs are to be automatically started.

Objectives

1. Customize the Full Screen WIN-OS2 session to automatically start the Clock program upon initialization of the WIN-OS2 session.
-

Setting the stage

1. Start the **WIN-OS/2 Full Screen** object in the *Command Prompts* folder.
2. Open the *WIN-OS/2 Accessories* group.
3. Select **Clock**
4. Select **File** from the Program Manager window.
5. Select **Copy...**

A dialog **Copy Program Item** will appear. The COPY Program Item field should indicate *CLOCK* and the From Program Group: field should indicate *WIN-OS/2 Accessories*

6. Ensure that the *To Group:* indicates *StartUp*
7. Select the *OK* pushbutton.
8. Close the Full Screen WIN-OS/2 session
9. Open the Full Screen WIN-OS/2 session

After completing the above steps, the WIN-OS/2 full screen session should have initialized and automatically started the clock.

LAB ON: Migration Database

Creating a Migration Database

Background

You can create your own database of DOS, Windows, and OS/2 applications and then specify that database (instead of DATABASE.DAT) when you run the Migrate Applications program. Your database would include specific settings for your DOS and Windows programs.

The first step in creating a migration database is to create a text file that lists the settings of the programs that you will want to migrate into the OS/2 environment. Once the text file has been created, you will use a utility, PARSEDB, to compile the file into a binary database. The binary database is used by the Migrate Applications utility to migrate applications to OS/2.

You will use SAMPLE.TXT, located in the Labs directory on the E drive as the basis for your database text file. SAMPLE.TXT is actually a subset of the file DATABASE.TXT file, located in the \OS2\INSTALL sub-directory.

Once the database text is completed, you will run the PARSEDB utility program to compile the file into a binary database. This binary file will be used by the Migrate Applications utility.

Objective

You will build a database to be used by the Migrate Applications utility which will automatically configure the applications:

- Graphic
- Fpamort

Revising an existing database

1. Use the OS/2 System Editor to modify the SAMPLE.TXT file as shown.

The file will be in LABS. SAMPLE.TXT may not require any modifications to run.... You might want to experiment with adding or changing some of its' values.

Read the body of the database text file for an explanation of the database fields. Do NOT delete the blank line after the IDLE_SECONDS statement, it indicates end of input for PARSEDB.

```

-----
REM Graphic.exe for DOS
REM -----
NAME                Graphic.EXE
TITLE               DOS Graphic
TYPE                DOS
ASSOC_FILE          NULL
DEF_DIR             \LABS
MOUSE_EXCLUSIVE_ACCESS  OFF
KBD_CTRL_BYPASS    NONE
KBD_ALTHOME_BYPASS OFF
DOS_RMSIZE          412
VIDEO_8514A_XGA_IOTRAP OFF
VIDEO_SWITCH_NOTIFICATION ON
DPMI_MEMORY_LIMIT  0
DOS_FILES           20
EMS_MEMORY_LIMIT    0
XMS_MEMORY_LIMIT    0
IDLE_SENSITIVITY   95
IDLE_SECONDS        4

REM ****You MUST leave a blank line at the end of the input.
REM
    
```

```

-----
REM FPAMORT.exe for Windows
REM -----
NAME                FPAMORT.EXE
TITLE               Loan Repayment
TYPE                Windows
ASSOC_FILE          NULL
DEF_DIR             \LABS
MOUSE_EXCLUSIVE_ACCESS  OFF
KBD_CTRL_BYPASS    CTRL_ESC
COMMON_SESSION      ON
KBD_ALTHOME_BYPASS  ON
DOS_RMSIZE          640
VIDEO_8514A_XGA_IOTRAP OFF
VIDEO_SWITCH_NOTIFICATION ON
DPMI_MEMORY_LIMIT  512
DOS_FILES           20
EMS_MEMORY_LIMIT    0
XMS_MEMORY_LIMIT    0
IDLE_SENSITIVITY   95
IDLE_SECONDS        4

REM ****You MUST leave a blank line at the end of the input.
REM
REM End of Database.
    
```

2. Save the file as **E:\LABS\SAMPDB.TXT**
3. Get to an OS/2 command prompt

4. Compile SAMPDB.TXT into a binary database
 - **PARSEDB D:\OS2\INSTALL\DBTAGS.DAT E:\LABS\SAMPDB.TXT E:\LABS\SAMPDB.DAT**

You should see the following text after pressing the <Enter> key:

- Lines: 23 Applications: 1
 - Writing "D:\labs\sampdb.dat" ...done!
 - Processed 1 DOS apps, 1 Windows apps, and 0 OS/2 apps
5. If any error messages are displayed, notify your instructor.
 6. Close the OS/2 command prompt.

Migrating applications using the default database

For your own satisfaction, run the Migrate Applications utility as indicated below to ensure that it would not automatically find FPAMORT or GRAPHIC.

1. Start the Migrate Applications utility.
This will be in the System Setup Folder.
2. Set the program to search the E, F and, G drives for applications.
3. Select Find.
4. Search for the program FPAMORT.
5. Did you find it? You should not find it.
6. Search for the program GRAPHIC.
7. Did you find it? You should not find it.
8. Continue with the migration.
9. Exit the Migration Application Utility.

Migrating the applications using the new database SAMPDB.DAT

1. Start the Migrate Applications utility.
2. Replace the text in the Database Used for Find Option entrybox with the path and filename of your binary database: E:\LABS\SAMPDB.DAT
3. Set the program to search the E drive for DOS and Windows applications.
4. Select Find.
5. Highlight only FPAMORT and GRAPHIC applications
6. Select to Migrate.

7. Exit when completed.

Verify that your application runs with the new settings

1. Get the pop-up menu for graphic.
2. Go to the DOS settings.
3. Verify that the settings you specified are in effect.
4. Exit the settings.
5. Get the pop-up menu for fpamort.
6. Go to the Win-OS2 settings.
7. Verify that the settings you specified are in effect.
8. Exit the settings.

LABS ON: Commands and retrieval

OS/2 Help Command Lab

Background

OS/2 offers a help facility. When you receive an error in the form of **SYSxxxx** you can get more detail information about this error. You can also get, more detail on OS/2 commands and parameters.

Objectives

- Verify that detail help exists for all **SYS** error codes.
 - Verify that one can get information about OS/2 commands without having to personally start the OS/2 On-line Command Reference.
-

SYS error codes

1. Open an **OS/2 Window**.
2. Type in **Help sys1041** and **<Enter>**.

Notice You will be given a repeat of the error code, a more detail explanation of the error and, suggested action(s).

3. Try the following system error codes;
 - a. **sys39**
 - b. **sys1719**
 - c. **sys25**

4. At this time do not close this window. You will be using this window for another lab.

Help Information

1. Open an **OS/2 Window**.
2. Type in **Help help** and **<Enter>**.
3. **Notice** that the system will start the OS/2 On-Line Command Reference and show you the page that explains this topic.
4. Minimize the window that contains the OS/2 On-Line Command Reference.

Note: The window that has the OS/2 On-Line Command Reference is a separate window than the one that you were using.

5. Using the same window that you used for entering the help command, request help for the following topics;
 - a. **Batch**

- b. **Path**
- c. **Format**

The system will not restart the OS/2 On-Line Command Reference if it is already 'started'.

- 6. To review the topics you have selected:
 - a. Select **Options** from the action bar of the OS/2 On-Line Command Reference.
 - b. From its' pull down menu, select **Viewed pages**

Notice: You can review all of the topics you had currently requested help for.

- 7. Do not close the window that you have been requesting help topics from.
You will be using this window for another lab.

OS/2 Commands and Command Retrieval

Background

OS/2 commands and syntax are nearly identical to those in DOS. There are additional OS/2 commands. The OS/2 Command Reference indicates which commands are unique to OS/2.

The **SET KEYS=ON** statement in the CONFIG.SYS file will allow the user to retrieve and edit the commands that were previously issued from the OS/2 Full Screen or Windowed command prompt.

The retrieval function can be enabled or disabled any time by typing **KEYS ON** or **KEYS OFF** respectively, even though SET KEYS=ON is in the CONFIG.SYS file.

The command queue is stored in a "wraparound buffer" which has a maximum size of 64KB. When the queue is full, the oldest commands will be discarded as new commands are entered from the keyboard.

The command, CHCP changes the code page for the current session, it is intended to be used at the command prompt. Any programs started under the primary command processor inherit the command processor code page. One can change the code page support in different sessions. For instance code page 437 can be activated in one session while code page 850 could be the active code page in another session. In OS/2 sessions, CHCP checks that the proper DEVINFO= statement is included in the CONFIG.SYS file for the code page being requested. If not, a message is displayed.

Objectives

- Review some existing DOS commands (for redirection, piping and filtering) that are also available in OS/2.
- Experiment with some OS/2 commands.
- Become familiar with the use of KEYS, KEYS LIST, KEYS ON and KEYS OFF commands.
- Be aware of the active code page on your system.

Redirection, Piping and Filtering

1. Redirect output from the DIR command to a file called MYFILE.NEW instead of displaying it to the screen.
 - a. Open an **OS/2 Full Screen** session.
 - b. From the E: Drive type **dir > myfile.new**
2. Pipe output from the TYPE command into the MORE command in order to see the contents of the file you created in the previous step.
 - a. Type **type myfile.new | more**

If the contents of the file could not be shown on 1 screen, then you would see the -- **MORE**-- appear at the bottom of the screen. Pressing the enter key would be your way of indicating to the computer to continue onto the next screen.

The MORE command sends output from a file to the screen, one full screen at a time.

3. Use the FIND command to filter MYFILE.NEW for a specific string of text.

- a. Type **find "labs" myfile.new**

No lines should appear as a result of that find.

Question: You have been using the LABS sub-directory. Why were there no findings of the word 'labs'?

Other OS/2 Commands

1. Use the OS/2 Command Reference to find information on the MOVE command. Note that this command is used for relocating one or more files from one directory to another on the **same partition (drive)**.
2. Move the MYFILE.NEW file from where it is currently located (the root of the E: Drive) to another directory and rename it.
 - a. Open an OS/2 session.
 - b. From the E: Drive command prompt type **move myfile.new E:\labs\newname.txt**
Note the error code you receive.
 - c. This time type **move myfile.new \labs\newname.txt**
3. Verify in LABS that it now includes a file called **NEWNAME.TXT**
4. Use the **OS/2 Command Reference** to recall information on the PSTAT command then try entering **PSTAT** from an OS/2 command prompt. Note that even if you are not running applications, OS/2 is busy executing OS/2 operating system related threads in the background.
5. Testing PSTAT;
 - a. Start another OS/2 Window.
 - b. Type in the PSTAT command but do NOT press the <Enter> key yet.
 - c. Start another OS/2 window and now run **MEMLAB4 5**.
 - d. Quickly return to the PSTAT window and execute it.
Notice what appears now that MEMLAB4 is running.
6. Use the OS/2 Command Reference to find information on the TYPE command. Note that this command in OS/2 displays the contents of **one or more** files. In DOS mode only one file is displayed.
7. Use the TYPE command to consecutively display the contents of all CONFIG and AUTOEXEC and BAK files in your root directory.
 - a. From the D: Drive command prompt type **type config.* autoexec.* *.bak**
 - b. Press the <Enter> key.

c. Watch as the contents of all the requested files are displayed.

Note that global file name characters (* and ?) can be used with the file names you specify.

8. Repeat the previous exercise but use the MORE command in conjunction with the TYPE command to display the output **one full screen at a time**.
9. Use the HELP OFF command to set the prompt back to the default system prompts for OS/2 and DOS modes.
 - a. Open an **OS/2 Full Screen** session.
 - b. From the D: Drive command prompt type **help off**
 - c. Press the <Enter> key.
 - d. Watch as the blue bar containing basic help information is removed.
 - e. Press **Ctrl + Esc** to get the Window List.
 - f. Open a **DOS Full Screen**.
 - g. From the D: Drive command prompt type **help off**
 - h. Press the <Enter> key.
 - i. Watch as the white bar containing basic help information is removed.

Note that with HELP OFF it is difficult to determine if you are in DOS mode or OS/2 full screen mode.

10. Use the HELP ON command to provide a help line as part of the command prompt.
 - a. From the **DOS Full Screen** type **help on**
 - b. Press the <Enter> key.
 - c. Watch as the white bar containing basic help information is added.
 - d. Open the **OS/2 Full Screen** session you used in the last step.
 - e. From the D: Drive command prompt type **help on**
 - f. Press the <Enter> key.
 - g. Watch as the blue bar containing basic help information is added.
11. Use the MKDIR (or MD) command in OS/2 to create multiple directories in one step.
 - a. From the D: Drive command prompt type **md Jim Barbara Camille**
 - b. Press the <Enter> key.
 - c. From the D: Drive command prompt type **dir ***. to see the 3 new directories you have created.

The following commands can have multiple arguments on one line -- DIR, DEL, TYPE, MD, RD, VOL.
12. Use the **RMDIR** (or RD) command in OS/2 to remove the directories you just created. Be sure to do this in just one step!
13. Enter multiple commands on one line using the command separator operator "&".
 - a. Open an OS/2 window.
 - b. From the D: Drive command prompt type **md newdir & copy D:\config.sys D:\newdir**
 - c. Press the <Enter> key.

- d. Verify there is a new directory called NEWDIR and that CONFIG.SYS has been copied into it.
14. Try using the command separator operator "&" from a DOS command prompt. Is it possible to enter multiple commands on a single line in DOS mode?

KEYS Commands

1. Retrieve the commands you have entered during this lab.
 - a. Return to the OS/2 session you have been working with.
 - b. Type **keys list** and press **Enter**.
 - c. You should see a list of all the commands you have entered in this OS/2 session.
 - d. Press the **Up Arrow** of the Cursor Movement Keys to retrieve the previous command in the queue.
 - e. Press the **Down Arrow** of the Cursor Movement Keys to retrieve the next command in the queue.
2. Turn OFF the retrieve key support for this OS/2 session and see how commands can be retrieved.
 - a. Type **keys off** and press <Enter>.
 - b. Try using the **Up Arrow** and **Down Arrow** to retrieve commands.
 - c. What happens?
 - d. To obtain the command you entered last press <F3>.
 - e. To obtain the last entry one character at a time press <F1>.
 - f. Type **keys list** to see the list of commands that were in the buffer **before the KEYS OFF command was executed**.
3. Start another OS/2 session to see if retrieve command support is active.
 - a. Start an **OS/2 Window** session.
 - b. Type **keys** to see if **KEYS is on**.
 - c. Type **keys list** and press <Enter>.
 - d. You should see a list of the commands that have been entered in **this OS/2 session**.

Notice that when KEYS OFF has been entered in one session, all other sessions are unaffected.
4. Retrieve the last command within A DOS Command prompt.
 - a. Start a DOS Command prompt.
 - b. Go to E:
 - c. Type **dir** and press <Enter>.
 - d. Type **cd\labs** and press <Enter>.
 - e. Press <F3> to retrieve only the last command entered.

OS/2's enhanced retrieval function is not supported in a DOS environment.. Within the DOS environment, the <F3> key is used to re-display only the last command that was issued.

5. Return to the first OS/2 session used in the KEYS Commands section of this lab (the one that currently has KEYS OFF) and enter more commands then turn KEYS ON again.
 - a. Make the **OS/2 Full Screen** session active.
 - b. Type **keys** to ensure **KEYS is off**.

- c. Type **dir** and press <Enter>.
- d. Enter any other OS/2 commands you wish.
- e. Type **keys list**
- f. Are the last few commands you entered in the list?
- g. Type **keys on** to make retrieve key support active.
- h. Type **cd** and press <Enter>.
- i. Type **keys list** to ensure that the last command you entered has been added to the command queue.

What Version? What CSD Level?

1. Use the VER command to determine what version of OS/2 is installed on your system.
 - a. Open an OS/2 session.
 - b. Type **ver** and press <Enter>.
 - c. Open a **DOS Full Screen** and type **ver** then press <Enter>.

Note that the DOS environment is actually part of OS/2 so the same version information is displayed.
2. Use the SYSLEVEL command to determine what Corrective Service Diskettes (CSDs) have been applied to your system.
 - a. From an OS/2 session type **syslevel** and press <Enter>.
 - b. Try this command from a **DOS Command prompt**.

Note that the CSD levels are displayed for each component of OS/2 installed.

Which is the active Code Page?

- Determine your sessions' active code page.
 1. Start an OS/2 window.
 2. Type **chcp** and press **Enter**.
A message will indicate which code page is the active one in this session.
 3. Open another **OS/2 window** and type **chcp**
Notice that the active code page will be the same as the previous sessions'.
- Changing the active code page in a window.
 1. Get to one of your opened windows.
 2. Type **chcp 850** and then press <Enter>.

If you press the <Alt> key and keep it pressed down while you press the following numbers on the keypad 2 , 1 , 0 (one after the other) and then release the <Alt> key, notice what character appears on the screen.

3. Type *chcp 437* and then press <Enter> .

Try the <Alt> key and keep it pressed down while you press the following numbers on the keypad 2 , 1 , 0 (one after the other) and then release the <Alt> key, notice what character appears on the screen.

This experiment is meant to show that you can have different characters appear on the screen in different sessions without having to change the physical keyboard. For example if one were to create a letter using the special French characters one could change to that active code page while still having the English keyboard layout as your physical keyboard.

DDE Lab

Background

You can use the dynamic data exchange (DDE) feature to change data in one program or session and have those changes take effect in another programs or session. Your program must be capable of using the DDE feature. The program can be a PM or a Windows application.

Some programs let you select either *on-demand* or *automatic* DDE. On-demand means that you must request the change between sessions. Automatic means that information is changed as soon as changes occur in one session.

This program, *Server.exe* has DDE capability and has been written to talk to a specific client, *Client.exe*.

With these programs, you will be able to see how the Client can request to be continuously updated with information. You will also be able to see, how a Client can send information to a Server.

This Lab is to give you some idea of the power of DDE. What could be created is a spreadsheet program. That program could be set up as the Server. A Client program could be a graphics program that would request data from the Server spreadsheet and then graph the result. As an example, the Client could request information from data file ABC, and from specific cells (A1-A15). The Server would send that data over, and the Client could create a graph based on that data.

Objectives

- Verify how DDE functions.
- Verify how to have a Client request for automatic updates from the Server.
- Verify how to have a Client request for a one time update from the Server.
- Verify how the Client can send information to the Server if required.

Starting the Client and Server

1. Open an OS/2 window session.
2. Get to the *Labs* directory.
3. Using the **start** command, start **Server.exe**
4. Using the **start** command, start **Client.exe**
5. Using the **start** command, start another **Client.exe**
6. Using the **start** command, start a third **Client.exe**

Initiating the link

1. Tile each of the Server and Client windows.
2. From each of the Clients, go to their Action bars and select **DDE**.
3. From the **DDE** pull down menu, select the option **Initiate**.

This is how the Clients request communication with the Server.

Requesting a one time update

1. From the **DDE** pull down menu, select the option **One Time Request**.

Notice that the server will send a message to that Client; *This is a request message 0 from Server.*

2. At the same Client, request another update from the Server.

Notice that the message will change from the number 0 to the number 1; *This is a request message 1 from Server.*

If you wish, try this on another of the Clients.

Requesting a continuous update

1. On another of your Clients, from their **DDE** pull down menu, select the option **Start Continuous**

Notice that the server will send a message to that Client; *This is a request message 0 from Server.* If this is the first request from this Client, then the message will start from the number 0 and increment.

The Server, maintains contact with all of its Clients, it will deal with each one individually. It will know what message number it has for each Client. The Server has been written, such that after 2 seconds it sends another message to any Client that has requested a continuous update.

If you wish, try this on another of the Clients.

A Client sending a message to the Server

1. From one of your Clients,
2. From the **DDE** pull down menu, select the option **Poke**.

Notice that the server will receive a message from that Client; *This is a poke message 0 from Client*.

3. At the same Client, do another Poke.

Notice that the message will change from the number 0 to the number 1; *This is a poke message 1 from Client*.

4. Send a Poke from another Client.

Notice that the message will indicate *This is poke message 0 from Client* This again indicates that the Server is keeping track of each Client individually.

Terminating a Client

1. From one of your Clients,
2. From one of the clients, select the option **Terminate** from the DDE pull down menu.

Notice that the server will stop sending any message to that Client, if it had been receiving it automatically. This does not cause the Client to 'die'.

You can re-initiate the Link to the Server.

LABS ON: Printing

Creating and Installing a Printer

Background

One printer may be installed during OS/2 installation. Additional printers can be added using a technique practiced in this lab.

Objectives

You will be able to:

- Create a new printer object.
- Install printer drivers for OS/2 and WIN-OS/2.

Creating a printer object using Templates

Creation of a printer object: IBM4019 Laser Printer E (IBM4019.DRV), using the diskette path of A:

1. Open the Templates folder and drag a Printer template to the Desktop.
2. Change the name field to **Laser Printer**
3. Select the **LPT2** port object in the Output port window.
4. Select **Install new printer driver...**
5. Verify the Printer driver shipped with OS/2 is selected in the Printer Driver selection box at the top of the panel.
6. Scroll and select the required OS/2 driver, IBM4019 Laser Printer E (IBM4019.DRV), and press **Install**
7. If using the LAN, the correct directory is determined by the system. Select **OK**
8. Select **OK** when the installation of the driver is complete.
9. Select **Create** to complete the definition of the printer object.
10. If prompted to install the equivalent WIN-OS/2 printer, answer **Yes**.

When the installation completes, your new printer object will be on the Desktop.

Working With Printers

Background

You have installed a printer object on the Desktop. In this exercise, you will explore how printers are used.

Objectives

- You will be able to:
 - Modify printer settings.
 - Modify and manipulate job queues.

Modify the printer settings for the new printer object, Laser printer

1. Change default view of the Laser Printer to a Details view.
 - Open the settings notebook for the Laser Printer Object.
 - On the View page, change the default to Details.
2. Change the paper orientation to Landscape.
 - Turn to the Printer Driver tab.
 - Select **Job Properties**

The Job Properties window is unique for each printer driver. Not all options shown may be available to all printer drivers.
 - Select **Help** to learn about other job properties for your printer driver.
 - Select **Landscape** to change the paper orientation for all jobs sent to this printer object.
 - Select **OK** to confirm the change.
 - Close the Settings notebook.

Exploring Job and Queue options.

Set the status of all printer queues on Hold.

1. Display the pop-up menu for the Laser Printer object and select **Change status**
2. Select Hold
3. Repeat for the other printer.

Making the Laser Printer the default printer

The system will route a print job to the default printer if no other printer is selected. Exceptions: The Print command, entered from a command prompt sends output to LPT1 by default, unless directed to a different port using the Device parameter.

1. Display the pop-up menu for the Laser Printer object.
2. Select **Set Default**
3. Select **Laser Printer**

View the contents of the Laser Printer queue

To view the contents of the queue assigned to Laser Printer, double-click the printer icon. The Job Details View for the printer displayed, should currently be empty.

1. Print C:\AUTOEXEC.BAT by opening Drive C, selecting AUTOEXEC.BAT, and dragging it to the printer.
2. Select **Plain text** for the type of data.
3. Print using the <Print Screen> keystroke.
4. Verify the two jobs are in the Laser printer queue.
5. Display the pop-up menu for the second job.

The job shown in the Job Details View window is an object, like other objects in the Workplace Shell, it can be manipulated. You may perform a number of tasks from the job's pop-up menu:

- Copy - allows you to make an additional copy of the selected object.
 - Delete - cancels the selected object.
 - Change status - allows you to hold (pause) a print job, or release a job that is currently held.
 - Print next - causes a particular job to print before any other pending jobs.
6. Select **Print next** and observe.

Verify a print command goes to LPT1, if no other port is selected

1. Open an OS/2 command prompt.
2. Type **PRINT CONFIG.SYS**
Question: Which queue was CONFIG.SYS print job sent to?
3. Display the job pop-up menu and select Delete.

Delete all jobs in the printer queues and close all printer windows.

WIN-OS/2 Printing and the Print Manager

Background

Generally, you should use the printer objects on the OS/2 Desktop to manage your print jobs. However, you may have customers who do not want to go to the OS/2 Desktop to see their print jobs. For a given port, if there is a printer object on the Desktop, jobs sent to that port from the WIN-OS/2 environment will not appear in the WIN-OS/2 Print Manager. In general, if you have no corresponding printer on the port, the WIN-OS/2 print jobs will be managed by the Print Manager.

You installed a Laser Printer on LPT2, and the corresponding WIN-OS/2 printer driver was installed.

In this exercise you will install WIN-OS/2 printer drivers, using the WIN-OS/2 Print Manager. Although OS/2 has a wide variety of printer drivers for use with WIN-OS/2, there may be times when you have already installed just the OS/2 driver, but not its WIN-OS/2 equivalent, or when you do not have an OS/2 driver.

In this lab, you will work with the WIN-OS/2 Print Manager.

Objectives

You will be able to:

- Install a WIN-OS/2 printer.
- Use and configure the WIN-OS/2 Print Manger.

Installing a WIN-OS/2 printer driver.

You will be installing a WIN-OS/2 printer: IBM Proprinter XLII using a diskette path of A:

1. Start a WIN-OS/2 full screen session.
2. Open the Control Panel.
3. Open Printers.

Note the printers installed previously are available for use in the WIN-OS/2 environment.

4. Select **Add >>** to display the List of Printers.
5. Use the scroll bar to display the required printer driver.

WIN-OS/2 does not create the path for the LAN. Change the drive and directory to the appropriate path and select **OK**

6. Select the printer in the Installed Printers list and select **Connect...**
7. Select **LPT1.OS2** in the Ports listbox. Press **OK**hp3.

Try to keep drivers in the OS/2 and WIN-OS/2 environment in matching pairs. If there is no equivalent OS/2 driver, create another printer on the Desktop, using IBMNULL for the printer driver, and assign it to LPT2.

8. Print job defaults may be modified by selecting the printer and selecting **Setup...**
9. Make the new printer the default printer.
10. Select the newly installed printer in the Installed Printers listbox.
11. Select the **Set Default Printer**

Print using the WIN-OS/2 printer

1. Turn on **Print Manager**
 - Check the Use Print Manager checkbox.
 - Verify the Proprinter or your attached printer is the Default Printer.
 - Close the Printers window and the Control Panel.
2. Start Print Manager in the WIN-OS/2 Main group.
3. Pause all printers. Minimize the Print Manager.
4. Start Notepad in the WIN-OS/2 Accessories group.
5. Enter some keystrokes and print the data.
6. Select Print Setup and verify that the default printer is your printer on LPT1. Select **OK**
7. Select **File and Print**

Question: Does the job appear in the Print Manager? It will not appear even though Print Manager is checked because there is a corresponding OS/2 port.
8. Return to the OS/2 Desktop. Note that your job has been sent to the OS/2 print system.
9. Delete the job.
10. Return to WIN-OS/2 and print the Notepad file again.

Question: Is your print job in the WIN-OS/2 Print Manager queue?
11. Delete all jobs in the Print Manager.
12. Select the Notepad (untitled) item and press Delete.
13. Close all open windows and close the WIN-OS/2 Program Manager.

LAB ON: Fonts

Installing OS/2 and WIN-OS2 Fonts

Background

Although OS/2 fonts can be installed during installation, you may not have installed all needed fonts or you may subsequently purchase additional ones.

Objectives

You will be able to:

- Add new fonts to be used by OS/2 applications.
- Add fonts to be used by a Windows application.
- Install printer drivers for OS/2 and WIN-OS/2.

Installing a Type 1 font for the WIN-OS2 Environment

ATM fonts for WIN-OS2 shipped with OS/2 are installed in \PSFONTS\PFM. Additional fonts have been installed in \LABS.

1. Start a WIN-OS2 full screen
2. Start the ATM Control Panel
3. Add a font of your choice:
 - Select **Add...** to produce the ATM Fonts window.
 - Use the directories listbox to build a source directory path for D:\PSFONTS\PFM
Other available fonts are displayed in the Available Fonts listbox.
 - Select any entry in the Available Fonts listbox.
 - Select **Add**
4. Activate ATM.

Selecting the **ON** activates ATM.

5. Select **Exit** to close the ATM Fonts window.

If installing a sans-serif (block-ended) font, only install the plain version of the font (do not install bold or italic versions). This will save space. ATM uses the plain version to create bold and italics.

6. Start WRITE and experiment with the new font you have selected.
7. Close the WIN-OS2 session.

Installing additional OS/2 fonts

Fonts available in the public domain have been installed in the \LABS directory for you.

1. Open the OS/2 *Font palette*
2. Select *Edit font...*
3. Select *Add...*
4. Change to the E:\LABS\FONTS and select *Add...*
5. Select a few fonts in the Font files window.
6. Select *Add*

To make use of the new added fonts you have to close the Fonts palette and re-open it. That is how the font list will be updated.

7. Experiment with some of the fonts.
8. Close all windows.

LABS ON: Installing applications

Installing AMIPRO for Windows

Background

AmiPro is a Windows application by Lotus. This version, that you will be installing, is actually a Working Model. This means that not all of the features of this application will be installed. Recall, the intent of this lab is not to learn how to use the application but to see how easy it is to install any application. The instructions for installing the entire package would not be any different.

AmiPro, is a windows word processor.

Objective

- To see one of the methods of installing a Windows application.
-

Installing AMIPRO

1. Start a Windows Full screen session.
2. Insert diskette entitled *AmiPro disk 1 of 2*
3. Select the word *File* from the Action Bar of WIN-OS2.
4. From its pull down menu, select the option **Run**
5. Type **A:install**
 - A pop-up menu will appear requesting for a name and initials. Select:
 - NAME to be **IBM**
 - INITIALS to be **IBM**
6. Select to *Install Ami Pro Working Model*.
7. Install this application on the **G** partition.
 - **G:\AMIPROWM**
8. Follow the instructions.

After the installation of this application, the WIN-OS2 session will remain open.

Starting AmiPro Working Model

1. Go to the **G** partition from a command prompt.
2. Go to `\amiprowm`
3. Type in `amipro.exe` to start the program.

Introducing AmiPro

1. Start Amipro.
2. Do the *QuickStart Tutorial* OR
3. From the AmiPro action bar, select **File**
4. From that pull-down menu, select **Open...**
5. Select the file *Mercury.sam*
6. Click on the **OK** button.
7. Read that file.
8. Experiment with other files.

Installing Describe 32 bit for OS/2

Background

Describe OS/2 application. The version you will be installing is their 32 bit application. Describe is a word processor. This version, is a Working Model.

Objective

- To see what is involved in installing a 32 bit OS/2 application.
-

Installing DESCRIBE

1. Open an **OS/2 session**.
2. Insert diskette entitled *DesScribe*.
3. Get to the **a** prompt.
4. Type **setupos2**.
5. Select the option **Options** from the Action Bar.
6. From the pull down menu, select the option to **Install DeScribe**.
A warning will appear, indicating that the stand-alone software is about to be installed.
7. Click on the **Yes** button to continue the installation.
8. Enter the destination of where this application will be installed.
Use the **G** drive.
9. Type in **G:\DesGold**
10. Select the **Cancel** button not to install this application in the group MAIN.
The installation program will take a few minutes to install its' files.
11. You should receive the message that the DeScribe installation has been successfully completed.
Select the **OK** button to get rid of that message.
12. Select the option **Options** from the Action Bar.
13. From the pull down menu, select the option to **Exit Installation**.
This will return you to the command prompt session.

Starting DeScribe

1. Go to the **G** partition from a command prompt.
2. Go to *\Desgold*
3. Type in *Describe*

Introducing DeScribe

1. Start Describe.
2. From the Describe action bar, select *File*.
3. From that pull-down menu, select **Open....**
4. Select **Directories** to be *Data* and double click on it.
5. Select *Document* and press the **OK** button.
6. Read that file.
7. Experiment with other files.

Installing WordPerfect for DOS

Background

WordPerfect is a DOS application.

Objective

- To see how one can install a DOS application.
-

Installing WORDPERFECT V5.1

1. Open an OS/2 window.
2. Insert diskette entitled *Install/Learn/Utilities*
3. Get to the **a** prompt.
4. Type **install**

OS/2 will determine that this application is actually not an OS/2 application and will open a DOS session.

A menu will appear asking if you wish to continue with the installation. Select **Y**
5. Select Yes when asked if installing to a Hard Drive

A menu will appear with default options.
6. Type the number 2.

The default of the installation is the C partition, which you will not be using. Option 2 will introduce another panel called *Custom Installation*
7. Type the number 2.

This will produce a panel called **Custom Install: Location of Files**
8. Type the number 1. (This will show a current path of where the installation program thinks it will be installing this application.)
9. Type the path **G:\WP51** and then press the Enter key.
10. If the WP51 directory does not exist, you will be asked if you wish it to be created. Answer **Y**
11. Type in the number 9.

This will exit out of your customized path information.
12. You will be returned to the **Custom Installation** panel.
13. Type the number 3, to install Disks.

14. Select YES to install the Utility Files.
15. Select YES to install the Learning Files.
16. Select YES to install the Help File.
17. Select YES to install the Keyboard Files.
18. Select YES to install the Style Library.
19. Select YES to install the Printer Test File.
20. Select YES to install the Wordperfect Program.
21. Insert diskette as prompted.
22. Select YES to install the Speller.
23. Insert diskette as prompted.
24. Select YES to install the Thesaurus.
25. Select NO to install the French Speller.
26. Select YES to install the PTR Program.
27. Insert diskette as prompted.
28. Select YES to install the Graphic Drivers.
29. Select NO to install the small.DRS file.
30. Select YES to install the Graphic Images.
31. Select Option 5 to select and install Printer and Exit.
32. Insert diskette as prompted.
33. Select the printer that is attached to your system.
34. There is 1 printer attached to your system.
35. When prompted for license number press enter.

Starting WORDPERFECT V5.1

1. Go to the **G** partition.
2. Go to **WP51**
3. Type in **wp.exe** to start the program.

Phone Numbers

IBM Service	Phone number
OS/2 Support	1-800-992-4777
IBM Direct	1-800-465-7999 x345
Customer Relations	1-800-465-6600
Customer Assistance	1-800-465-1234 (905) 316-9000
Education Direct	1-800-661-2131 (905) 946-1100
IBM Dispatch (hard)	1-800-465-6666 (905) 316-1222
IBM Dispatch (soft)	1-800-465-2222 (905) 316-1333
IBM BBS Montreal	(514) 938-3022
IBM BBS Vancouver	(604) 664-6466
IBM BBS Toronto	(905) 316-4255 (416) 492-1823
IBM BBS Atlanta	(404) 835-6600

Appendix A. Appendix A - Installation Files

Response file topics:

There are five sections in this Appendix:

- Section 1 -- The contents of a SAMPLE.RSP
- Section 2 -- The contents of a USER.RSP
- Section 3 -- The contents of a CONFIG.SYS file for Response File Install
- Section 4 -- The contents of a CONFIG.SYS file for Panel Install
- Section 5 -- The contents of a PRDESC.LST file

Section 1 -- SAMPLE.RSP

The following pages contain the contents of the SAMPLE.RSP file. The file can also be found in the OS2\INSTALL subdirectory of OS/2 2.1.

```
*
* Advance Power Management
*
* Specifies whether or not to install APM.
*
* Valid Params:
*
* 0= Don't install
* 1= Autodetect (DEFAULT)
* 2= Install
*
*****

APM= 1

*****
*
* AlternateAdapter
*
* Specifies secondary adapter for two display systems.
* This should be a lower or equal resolution display since
* the highest resolution display will be primary for PM.
*
* Valid Params:
*
* 0= None (DEFAULT)
* 1= Other than following (DDINSTAL will handle)
* 2= Monochrome/Printer Adapter
* 3= Color Graphics Adapter
* 4= Enhanced Graphics Adapter
* 5= PS/2 Display Adapter
* 6= Video Graphics Adapter
* 7= 8514/A Adapter
```

- * 8 = XGA Adapter
- * 9 = SVGA Adapter
- *

AlternateAdapter = 0

* BaseFileSystem

* Specifies which file system should be used to format
* the install partition

* Valid Parms:

- * 1 = HPFS (DEFAULT)
- * 2 = FAT
- *

BaseFileSystem = 1

* CDROM

* Specifies which, if any, CD ROM devices you wish to
* install support for.

* Valid Parms:

- * 0 = None
- * 1 = Autodetect
- * 2 = CDTechnology T3301
- * 3 = HitachiCDR-1650,1750,3650
- * 4 = HitachiCDR-3750
- * 5 = IBMCD-ROM I
- * 6 = IBMCD-ROM II
- * 7 = NEC25,36,37,72,73,74,82,83,84
- * 8 = NECMultiSpin 38,74,84
- * 9 = PanasonicCR-501,LK-MC501S
- * 10 = PioneerDRM-600
- * 11 = PioneerDRM-604X
- * 12 = SonyCDU-541,561,6211,7211
- * 13 = SonyCDU-6111
- * 14 = TexelDM-3021,5021
- * 15 = TexelDM-3024,5024
- * 16 = Toshiba3201
- * 17 = Toshiba3301,3401
- * 18 = OTHER
- *

* NOTE: Autodetection is enabled
only when all scsi

* device drivers are loaded.

CDROM = 0

* CountryCode

* Specifies which country should be installed. This
* causes all country information to be installed.

* Valid Parm:

* 3 digit country code (DEFAULT shipped version)

CountryCode = 001

* CountryKeyboard

* Specifies which country keyboard should be installed.
* This causes all keyboard information to be installed.

* Valid Parm:

* 2-5 character keyboard code (DEFAULT = "US")

CountryKeyboard = US

* DefaultPrinter

* Specifies which default printer to install

* Valid Parm:

* 0 = None
* or

* Keyvalue = printer driver index (DEFAULT = line # of
* 42XX) in PRDESC.LST shipped on first printer diskette

* NOTE: the driver index is the same as the line
* number in the ASCII PRDESC.LST file that
* the desired printer name appears on

DefaultPrinter = 0

```
*****
*
* DiagnosticAids
*
* Specifies whether or not to install certain RAS
* utilities.
*
* Valid Params:
*
* 0 = Don't install
* 1 = Install (DEFAULT)
*
*****
```

DiagnosticAids = 1

```
*****
*
* DisplayAdapter
*
* Specifies which adapter should override the primary
* adapter detected by the install process
*
* Valid Params:
*
* 0 = Accept as correct (DEFAULT)
* 1 = Other than following (DDINSTAL will handle)
* 2 = Color Graphics Adapter
* 3 = Enhanced Graphics Adapter
* 4 = Video Graphics Adapter
* 5 = 8514/A Adapter
* 6 = XGA Adapter
* 7 = SVGA Adapter
*
*****
```

DisplayAdapter = 0

```
*****
*
* Documentation
*
* Specifies which documentation should be installed
*
* Valid Params:
*
* 0 = None
* 1 = All (DEFAULT)
* 2 = OS/2 Command Reference
* 3 = OS/2 Tutorial
* 4 = Rexx Documentation
*
*****
```

Documentation = 1

```
*****
*
* DOSSupport
*
* Specifies whether or not to install DOS Box.
*
* Valid Parms:
*
* 0 = Don't install DOS
* 1 = Install DOS (DEFAULT)
*
*****
```

DOSSupport = 1

```
*****
*
* WIN-OS/2Support
*
* Specifies whether or not to install WIN-OS/2
* Environment. If do, select WIN-OS/2 groups or
* other components. This option is valid only
* when option 1 (DOSSupport) is selected for
* the DOSSupport keyvalue.
*
* Valid Parms:
*
* 0 = Do NOT install WIN-OS/2
* ---- Followings INSTALL WIN-OS/2 -----
* 1 = All available groups and components (DEFAULT)
* 2 = WIN-OS/2 Readme File
* 3 = WIN-OS/2 Accessories Group
* 4 = WIN-OS/2 Screen Save Utility
* 5 = WIN-OS/2 Sound Utility
* 6 = WIN-OS/2 Main and Startup Group ONLY (Minimum support)
*
* Note:
* * WIN-OS/2 Main Group and StartUp Group will be
* installed mandatorily when WIN-OS/2 supported
* ( case 1,2,3,4,5 ).
* * Case 6 is minimum WIN-OS/2 support.
*
* Example:
*
* WIN-OS/2Support = 3,4
* would install WIN-OS/2 Main Group, StartUp Group and
* WIN-OS/2 Accessories and Screen Save Utility.
*
*****
```

WIN-OS/2Support = 1

```

*****
*
* WindowedWIN-OS/2
*
* Specifies whether Windows** applications should run in
* windowed sessions on the Presentation Manager desktop
* or in Full Screen sessions. This option is valid only
* when option 1 (WIN-OS/2 Support) is selected for the
* DOSSupport keyvalue.
*
* Valid Parms:
*
* 0= Windowed WIN-OS/2 sessions
* 1= Full Screen WIN-OS/2 sessions
*
*****

```

```
*WindowedWIN-OS/2 = 1
```

```

*****
*
* WIN-OS/2Desktop
*
* Specifies what the WIN-OS/2 desktop should look like.
* This option is valid only when option 1 (WIN-OS/2
* Support) is selected for the DOSSupport keyvalue.
* Option 1 should be selected only if Windows** currently
* exists (two related options follow this one).
* Option 2 should be selected only if WIN-OS/2 has
* previously been installed.
*
* Valid Parms:
*
* 0= Install standard WIN-OS/2 desktop (DEFAULT)
* 1= Copy existing Windows** desktop and use as the
* WIN-OS/2 desktop (two related options follow)
* 2= Preserve WIN-OS/2 desktop currently installed
*
*****

```

```
*WIN-OS/2Desktop = 0
```

```

*****
*
* ExistingWindowsPath
*
* Specifies the path to an existing Windows** system.
* This option is valid only when option 1 is selected
* for the WIN-OS/2Desktop keyvalue.
*
* Valid Parms:
*
* A string that specifies the path to the existing
* Windows** system (Example: C:\WINDOWS)

```

```

*
*****
*ExistingWindowsPath =
*****
*
* ShareDesktopConfigFiles
*
* Specifies that the desktop configuration files should
* be shared between an existing Windows** system and the
* WIN-OS/2 system being installed. If this option is
* selected, the Windows** desktop will be updated when
* changes are made to the WIN-OS/2 desktop. This
* option is valid only when option 1 is selected for the
* WIN-OS/2Desktop keyvalue.
*
* Valid Parm:
*
* 0 = Do not share the Windows** desktop configuration
* files
* 1 = Share the Windows** desktop configuration files
*
*****

*ShareDesktopConfigFiles = 1
*****
*
* DPMI
*
* Specifies which DPMI options to install.
*
* Valid Parm:
*
* 0 = none
* 1 = All (DEFAULT)
* 2 = Virtual DOS Protect Mode Interface
* 3 = Virtual Expanded Memory Management
* 4 = Virtual Extended Memory Support
*
*****

DPMI = 1
*****
*
* ExitOnError
*
* Specifies if the install program should exit with an
* error code if an error occurs. This also determines
* whether the installation process will exit with a return
* code when it completes rather than the C-A-D panel.
*
* Valid Parm:

```

```
*
* 0 = Do not exit when error occurs; display panel
* (DEFAULT)
* 1 = Exit quietly with a return code
*
*****
```

ExitOnError = 0

```
*****
*
* Fonts
*
* Specifies which fonts should be installed
*
* Valid ParmS:
*
* 0 = None
* 1 = All (DEFAULT)
* 2 = Courier (Bitmap)
* 3 = Helvetica (Bitmap)
* 4 = System Mono-spaced (Bitmap)
* 5 = Times Roman (Bitmap)
* 6 = Courier (Outline)
* 7 = Helvetica (Outline)
* 8 = Times New Roman (Outline)
*
*****
```

Fonts = 1

```
*****
*
* FormatPartition
*
* Specifies whether or not to format the install
* partition
*
* Valid ParmS:
*
* 0 = Do not format (DEFAULT)
* 1 = Format
*
*****
```

FormatPartition = 0

```
*****
*
* Include
*
* For a description of the function of this keyword,
* see IncludeAtEnd which is functionally equivalent
* to this keyword.
*
```

```
* Valid ParmS:
*
*   KEYWORD = valid filename
*
*****
```

```
* Include = include.rsp
*****
```

```
* IncludeAtEnd
*
* Specifies another response file to process along
* with the current one. There may be multiple
* occurrences of this keyword. The "included"
* response file is appended to the end of all
* response files that have been processed before
* this one.
```

```
* eg.
* - File1.RSP           Processing
*
*   IncludeAtEnd = File2.RSP      Mouse = 1
*   IncludeAtEnd = File4.RSP      Mouse = 2
*   Mouse = 1                    Mouse = 4
*                               Mouse = 3
* - File2.RSP
*
*   IncludeAtEnd = File3.RSP
*   Mouse = 2
* - File3.RSP
*
*   Mouse = 3
* - File4.RSP
*
*   Mouse = 4
```

```
* No validity checking is done.
*
* Valid ParmS:
*
*   KEYWORD = valid filename
*
*****
```

```
* IncludeAtEnd = atend.rsp
*****
```

```
* IncludeInLine
*
* Specifies another response file to process along
```

* with the current one. There may be multiple
* occurrences of this keyword. The "included"
* response file is processed immediately when the
* keyword is found.
* No validity checking is done.

* eg.
* - File1.RSP Processing
* IncludeInLine = File2.RSP Mouse = 3
* IncludeInLine = File4.RSP Mouse = 2
* Mouse = 1 Mouse = 4
* Mouse = 1

* - File2.RSP
* IncludeInLine = File3.RSP
* Mouse = 2

* - File3.RSP
* Mouse = 3

* - File4.RSP
* Mouse = 4

* Valid Params:
* KEYWORD = valid filename

* IncludeInLine = inline.rsp

* MigrateConfigFiles
* Specifies whether or not to migrate configuration files
* from a previous release of the operating system.

* Valid Params:
* 0 = Don't migrate
* 1 = Migrate files (DEFAULT)

MigrateConfigFiles = 1

* MigrateApplications

*

```

* Specifies whether or not to migrate existing DOS,
* Windows** and OS/2 applications. Only those
* applications listed in the database specified will
* be migrated.
*
* Valid Parms:
*
*   Drives to search, database to use for search
*   (Example: C:D:,C:\OS2\INSTALL\DATABASE.DAT)
*
*****
* MigrateApplications =
*****
*
* MoreBitmaps
*
* Specifies whether or not to install more bitmaps.
*
* Valid Parms:
*
*   0 = Don't install More Bitmaps
*   1 = Install More Bitmaps (DEFAULT)
*
*****
MoreBitmaps = 1
*****
*
* Mouse
*
* Specifies which mouse device driver, if any, to
* install
*
* Valid Parms:
*
*   0 = No pointing device support
*   1 = PS/2 Style Pointing Devicee      (DEFAULT)
*   2 = Bus Version
*   3 = Serial Version
*   4 = InPort Version
*   5 = Logitech (tm) 'C' Series Serial Mouse
*   6 = IBM PS/2 Touch Display
*   7 = Logitech 'M' Series Mouse
*   8 = PC Mouse Systems (tm) Mouse
*   9 = Other Pointing Device for Mouse Port
*
*****
Mouse = 1
*****
*
* MousePort

```

*
* Specifies to which port a serial-type mouse should
* be attached (valid for serial or Logitech(tm) mice)

* Valid ParmS:

- * 0 = No port necessary (DEFAULT)
- * 1 = COM1
- * 2 = COM2
- * 3 = COM3
- * 4 = COM4

MousePort = 0

* OptionalFileSystem

* Specifies whether or not to install optional file
* system(s) i.e. HPFS

* Valid ParmS:

- * 0 = Do Not Install Optional File System(s)
- * 1 = Install Optional File System (DEFAULT)

OptionalFileSystem = 1

* OptionalSystemUtilities

* Specifies whether or not to install the following
* system utilities.

* Valid ParmS:

- * 0 = Install none
- * 1 = Install all (DEFAULT)
- * 2 = Backup Hard Disk
- * 3 = Change File Attributes
- * 4 = Display Directory Tree
- * 5 = Manage Partitions
- * 6 = Label Diskettes
- * 7 = Link Object Modules
- * 8 = Picture Utilities
- * 9 = PMREXX
- * 10 = Recover Files
- * 11 = Restore Backed-up Files
- * 12 = Sort Filter
- * 13 = Installation Aid

*
 * Example:
 * OptionalSystemUtilities = 2,9,4
 * would install Backup,
 PMREXX
 and Tree utilities.
 *

OptionalSystemUtilities = 1

* OS2IniData
 *
 * Specifies a profile string to be written to the
 * user configuration file OS2.INI. There may be
 * multiple occurrences of this keyword.
 *
 * Valid Params:
 *
 * KEYWORD = /AppName/KeyName/KeyValue/
 *
 * NOTE: Since each of these names can contain
 * imbedded blanks and whitespace, the "slash"
 * character must be used as a delimiter. There
 * must be three tokens delineated on all sides or
 * this keyword will be ignored.
 *

OS2IniData = /AppName/KeyName/KeyValue/

* PCMCIA
 *
 * Specifies whether or not to install PCMCIA.
 *
 * Valid Params:
 *
 * 0 = Don't install
 * 1 = Install (DEFAULT)
 *

PCMCIA = 1

* PrimaryCodePage
 *
 * Specifies whether "national" or "multi-lingual" code
 * page is primary (first active code page before
 * switching).

*
* Valid Parms:
*
* 1 = National (DEFAULT)
* 2 = Multilingual
*

PrimaryCodePage = 1

*
* PrinterPort
*
* Specifies to which printer port the default printer
* should be attached
*
* Valid Parms:
*
* 1 = LPT1 (DEFAULT)
* 2 = LPT2
* 3 = LPT3
* 4 = COM1
* 5 = COM2
* 6 = COM3
* 7 = COM4
*

PrinterPort = 1

*
* ProcessEnvironment
*
* Each of the Keyword/Keyvalue statements specified in
* this response file may be added to the environment as
* environment variables.
* This makes it possible for user programs, batch files,
* etc. (UserExit) to access response file settings.
*
* Valid Parms:
*
* 0 = Do not add keyword/keyvalue statements specified
* in this response file to environment.
* 1 = Add all keyword/keyvalue statements specified
* in this response file to environment (DEFAULT).
*

ProcessEnvironment = 1

*
* ProgressIndication

*
* Specifies whether or not to display progress indicators
* during the installation. Disabling this will allow a
* frontend program to display something else while we do
* our job in an unattended environment.
*

* Valid Params:

* 0 = No progress indication
* 1 = Progress indication (DEFAULT)
*

ProgressIndication = 1

* RebootRequired

* Specifies if the machine should be automatically
* warm booted when installation is complete. This is
* ignored if the ExtendedInstall response is specified.
*

* Valid Params:

* 0 = Ask user to reboot (DEFAULT)
* 1 = Auto-reboot
*

RebootRequired = 0

* REXX

* Specifies whether or not to install REXX
*

* Valid Params:

* 0 = Don't Install REXX
* 1 = Install REXX (DEFAULT)
*

REXX = 1

* SCSI

* Specifies which, if any, CD ROM adapter support you
* wish to install support for.
*

* Valid Params:

- * 0 = None
- * 1 = Autodetect
- * 2 = Adaptec1510, 1520, 1522
- * 3 = Adaptec1540, 1542
- * 4 = Adaptec1640
- * 5 = Adaptec1740, 1742, 1744
- * 6 = DPTPM2011, PM2012
- * 7 = FutureDomain 845,850,850IBM,860,875,885
- * 8 = FutureDomain 1650,1660,1670,1680,MCS700
- * 9 = FutureDomain 7000EX
- * 10 = IBMPS/2 SCSI Adapter
- * 11 = IBM16-Bit AT Fast SCSI Adapter

SCSI = 1

* SerialDeviceSupport

* Specifies whether or not to install the serial device driver.

* Valid Parms:

- * 0 = Don't install
- * 1 = Install (DEFAULT)

SerialDeviceSupport = 1

* SourcePath

* Specifies a single media (no disk switching) that should be used as a source drive and directory from which to install.

* Valid Parms:

- * KEYVALUE = drive and optional path (D:\OS2SE20\...)
- * DEFAULT = A:\

* SourcePath = D:\os2se21

* TargetDrive

*

- * Specifies the target drive to which OS/2 should be installed. This drive is assumed to be a valid partition. If a partition other than C: is specified, it is assumed that MOST support is already installed to enable booting an operating system from any partition.

* Valid Params:

* KEYVALUE = d:

* where "d:" is a valid partition that OS/2 may be installed to.

* DEFAULT = first acceptable partition

 * TargetDrive = C:

* WIN-OS/2TargetDrive

* Specifies which valid partition drive to install WIN-OS/2.

* Valid Params: any valid FORMATTED partition.

* C: (DEFAULT)

* D:

* .

* .

* Z:

* Example:

* WIN-OS/2TargetDrive = D:

* would install WIN-OS/2 to partition D: located in

* \OS2\MDOS\WINOS2

*WIN-OS/2TargetDrive = C:

* ToolsAndGames

* Specifies whether or not to install tools and games such as editors and jigsaw.

* Valid Params:

* 0 = Install none

* 1 = Install all (DEFAULT)

* 2 = Enhanced Editor

- * 3 = Search and Scan Tool
- * 4 = Terminal Emulator
- * 5 = Chart Maker
- * 6 = Personal Productivity
- * 7 = Solitaire - Klondike
- * 8 = Reversi
- * 9 = Scramble
- * 10 = Cat and Mouse
- * 11 = Pulse
- * 12 = Jigsaw
- * 13 = Chess

* Example:

- * ToolsAndGames = 2,8,13
- * would install the Enhanced Editor, Reversi and
- * Chess.

ToolsAndGames = 1

* ConfigSysLine

- * Specifies a text line to be appended to CONFIG.SYS.
- * There may be multiple occurrences of this keyword.
- * No validity checking is done.

* Valid Params:

- * KEYWORD = a valid CONFIG.SYS statement

* ConfigSysLine = REM This is a CONFIG.SYS remark line.

* Copy

- * Specifies a source file and destination directory
- * of a file to be copied during install. Errors are
- * ignored, though they will be logged. Packed files
- * are acceptable since UNPACK will do the copy.
- * There may be multiple occurrences of this keyword.
- * No validity checking is done.

* Valid Params:

- * KEYWORD = source file destination

- * where source file = valid filename
- * and destination = valid directory name

*

```

*      ex: Copy = readme.dat c:\os2
*
*****

* Copy = vga c:\ /n:ini.rc
*****

*
* EarlyUserExit
*
* Specifies the name of a program that Install will
* DosExec after the target drive is prepared. Install
* waits for the program to return. This keyword may occur
* more than once. Each will be executed in the order that
* they appear at the end of OS/2 Install. The only
* difference between this keyword and the UserExit keyword
* is that this one is executed early in the installation
* process while the latter is executed at the very end.
*
* Valid Parms:
*
*      KEYVALUE = user exit program name (DEFAULT = none)
*
*****

* EarlyUserExit = T c:\config.sys
*****

*
* ExtendedInstall
*
* Specifies program to be run asynchronously while SE
* Install DosExits
*
* Valid Parms:
*
*      KEYVALUE = full pathname of program
*      (DEFAULT = none)
*
*****

* ExtendedInstall = PROGRAM.EXE
*****

*
* ID
*
* Specifies some identification string which may be
* used by install or UserExit to identify the
* response file(s) used for this installation
*
* Valid Parms:
*
*      KEYWORD = ASCII string

```

```

*
*****
*ID = OS2SE20 Sample Response File
*****
*
* SeedConfigSysLine
*
* Specifies a text line to be appended to the CONFIG.SYS
* written to the seed system from which PM Install boots.
* This will allow device drivers (that may be required) to
* become part of that seed system.
* There may be multiple occurrences of this keyword.
* No validity checking is done.
*
* Valid Params:
*
*   KEYWORD = a valid CONFIG.SYS statement
*
*****
* SeedConfigSysLine = REM This is a remark line in the seed CONFIG.SYS.
*****
*
* UserExit
*
* Specifies the name of a program that Install will
* DosExec before exiting memory. Install waits for the
* program to return. This keyword may occur more than
* once. Each will be executed in the order that they
* appear at the end of OS/2 Install.
*
* Valid Params:
*
*   KEYVALUE = user exit program name (DEFAULT = none)
*
*****
* UserExit = T.EXE C:\OS2\INSTALL\INSTALL.LOG
*****
*
* Version
*
* Specifies specific version of the operating system for
* which this file is intended. The file can be used for
* future versions, though some keywords may no longer
* be valid.
*
* Valid Params:
*
*   KEYWORD = some version string (determined later)
*
*****

```

*Version = OS2SE20

```
*****
*
* DDInstall
* Use OS/2 Device Driver Installation to install external
* loadable device drivers. A Device Driver Profile ( a
* text file with a .DDP file name extension) must be
* provided by the device driver author to control the
* installation of the device driver.
*
* Valid Parms:
*   DDISrc = Directory where the .DDP files are.
*   DDIDest = Directory where to copy the device driver
*             files.
*   DDIDDP = List of .DDP files to install.
*             (example: file1.DDP,file2.DDP)
*
*
*
*****
```

```
*DDISrc = Z:\DDP
*DDIDest = C:\
*DDIDDP = *.DDP
```

Section 2 -- USER.RSP

This is the file built when you completed your initial installation.

```
AlternateAdapter = 0
APM = 0
BaseFileSystem = 2
CDROM =
SCSI =
CountryCode = 001
CountryKeyboard = US
DefaultPrinter = 0
DiagnosticAids = 1
DisplayAdapter = 0
Documentation = 1
DOSSupport = 1
WIN-OS/2Support = 1
*WindowedWIN-OS/2 = 1
*WIN-OS/2Desktop = 0
*ExistingWindowsPath =
*ShareDesktopConfigFiles = 1
DPMI = 1
ExitOnError = 0
Fonts = 1
FormatPartition = 1
MigrateConfigFiles = 0
*MigrateApplications =
MoreBitmaps = 1
```

```
Mouse = 1
MousePort = 0
OptionalFileSystem = 1
OptionalSystemUtilities = 1
PCMCIA = 1
PrimaryCodePage = 1
PrinterPort = 1
ProcessEnvironment = 1
ProgressIndication = 1
RebootRequired = 0
REXX = 1
SerialDeviceSupport = 1
SourcePath = X:\
*WIN-OS/2TargetDrive = e:
TargetDrive = F:
ToolsAndGames = 1
*ConfigSysLine =
```

Section 3 -- CONFIG.SYS file for a Response File Install

```
buffers = 32
iopt = yes
memman = noswap
protshell = sysinst1.exe
set os2_shell = cmd.exe /K A:\startup.cmd
diskcache = 64,LW
protectonly = yes
libpath = .;\os2\dll;X:\img\lcu;
ifs = hpfs.ifs /c:64
pauseonerror = no
codepage = 850
devinfo = kbd,us,keyboard.dcp
devinfo = scr,ega,vt1850.dcp
device = \dos.sys
set path = \;\os2;\os2\system;\os2\install;A;;
set dpath = \;\os2;\os2\system;\os2\install;A;;
set keys = on
basedev = print01.sys
basedev = ibm1flpy.add
basedev = ibm1s506.add
basedev = ibm2flpy.add
basedev = ibm2adsk.add
basedev = ibm2m57.add
basedev = ibm2scsi.add
basedev = ibmint13.i13
basedev = os2dasd.dmd
device = \testcfg.sys
device = \refpart.sys

rem *** Start of ThinLaps additions ***
device = lanmsgdd.os2
device = protman.os2
device = netbeui.os2
```

```

device = netbios.os2
device = ibmtok.os2
run = netbind.exe
run = lanmsgex.exe
call = a:\srvattch.exe y: server1
call = a:\srvattch.exe z: \\server1\lculog
device = a:\srvifs.sys
ifs = a:\srvifsc.ifs * /s:2 /a:0
call = a:\srvattch.exe x: \\server1\gdrive
run = x:\img\lcu\svrrexx.exe

```

Section 4 -- CONFIG.SYS file for a Panel Install

```

iopt = yes
memman = noswap
protshell = sysinst1.exe
set os2_shell = x:\img\os2v20\disk_1\sysinst2.exe x:\img\os2v20
diskcache = 64,LW
protectonly = yes
libpath = .;\os2\dll;X:\img\lcu;
ifs = hpfs.ifs /c:64
pauseonerror = no
codepage = 850
devinfo = kbd,us,keyboard.dcp
devinfo = scr,ega,vtbl850.dcp
device = \dos.sys
set path = \;\os2;\os2\system;\os2\install;A;
set dpath = \;\os2;\os2\system;\os2\install;A;
set keys = on
basedev = print01.sys
basedev = ibm1flpy.add
basedev = ibm1s506.add
basedev = ibm2flpy.add
basedev = ibm2adsk.add
basedev = ibm2m57.add
basedev = ibm2scsi.add
basedev = ibmint13.i13
basedev = os2dasd.dmd
device = \testcfg.sys
device = \refpart.sys

rem *** Start of ThinLaps additions ***
device = lanmsgdd.os2
device = protman.os2
device = netbeui.os2
device = netbios.os2
device = ibmtok.os2
run = netbind.exe
run = lanmsgex.exe
call = a:\srvattch.exe y: server1
call = a:\srvattch.exe z: \\server1\lculog
device = a:\srvifs.sys
ifs = a:\srvifsc.ifs * /s:2 /t /a:0

```

```
call = a:\srvattch.exe x: \\server1\gdrive
run = x:\img\lcu\srvrexx.exe
buffers = 32
```

Section 5 -- Printer description list

1. AST TurboLaser: AST TurboLaser (PSCRIPT.DRV)
2. Agfa Matrix ChromaScript v51_8: Agfa Matrix ChromaScript v51_8 (PSCRIPT.DRV)
3. Agfa-Compugraphic 9400PS v49_3: Agfa-Compugraphic 9400PS v49_3 (PSCRIPT.DRV)
4. Agfa/Compugraphic 400PS: Agfa/Compugraphic 400PS (PSCRIPT.DRV)
5. Apple LaserWriter: Apple LaserWriter (PSCRIPT.DRV)
6. Apple LaserWriter II NT: Apple LaserWriter II NT (PSCRIPT.DRV)
7. Apple LaserWriter II NTX: Apple LaserWriter II NTX (PSCRIPT.DRV)
8. Apple LaserWriter Plus: Apple LaserWriter Plus (PSCRIPT.DRV)
9. Apple LaserWriter Plus v42_2: Apple LaserWriter Plus v42_2 (PSCRIPT.DRV)
10. COMPAQ PAGEMARQ 15: COMPAQ PAGEMARQ 15 (PSCRIPT.DRV)
11. COMPAQ PAGEMARQ 20: COMPAQ PAGEMARQ 20 (PSCRIPT.DRV)
12. Citizen PN48: Citizen PN48 (EPSON.DRV)
13. Colormate PS v51_9: Colormate PS v51_9 (PSCRIPT.DRV)
14. Dataproducts LZR 1260 v47_0: Dataproducts LZR 1260 v47_0 (PSCRIPT.DRV)
15. Dataproducts LZR-2665: Dataproducts LZR-2665 (PSCRIPT.DRV)
16. Digital LN03R ScriptPrinter: Digital LN03R ScriptPrinter (PSCRIPT.DRV)
17. Digital LPS PrintServer 40: Digital LPS PrintServer 40 (PSCRIPT.DRV)
18. Epson 24 pins - 136 columns: 24-pin 136 Col (EPSON.DRV)
19. Epson 24 pins - 80 columns: 24-pin 80 Col (EPSON.DRV)
20. Epson 9 pins - 136 columns: 9-pin 136 Col (EPSON.DRV)
21. Epson 9 pins - 80 columns: 9-pin 80 Col (EPSON.DRV)
22. Epson AP-2250 9 pins - 80 columns: AP-2250 (EPSON.DRV)
23. Epson AP-3250 24 pins - 80 columns: AP-3250 (EPSON.DRV)
24. Epson AP-5000 24 pins - 80 columns: AP-5000 (EPSON.DRV)
25. Epson AP-5500 24 pins - 136 columns: AP-5500 (EPSON.DRV)
26. Epson ActionLaser 1000/1500: Epson ActionLaser 1000/1500 (LASERJET.DRV)
27. Epson ActionLaser II: Epson ActionLaser II (LASERJET.DRV)
28. Epson DFX-5000 9 pins - 136 columns: DFX-5000 (EPSON.DRV)
29. Epson DFX-8000 9 pin - 136 column: DFX-8000 (EPSON.DRV)
30. Epson EPL-6000 Laser: EPL-6000 (EPSON.DRV)
31. Epson EPL-7000: Epson EPL-7000 (LASERJET.DRV)

32. Epson EPL-7500 v52_3: Epson EPL-7500 v52_3 (PSCRIPT.DRV)
33. Epson EPL-8000: Epson EPL-8000 (LASERJET.DRV)
34. Epson EPL-8000 PS Card 82605: Epson EPL-8000 PS Card 82605 (PSCRIPT.DRV)
35. Epson EX-1000 Color 9 pins - 136 columns: EX-1000 (EPSON.DRV)
36. Epson EX-800 Color 9 pins - 80 columns: EX-800 (EPSON.DRV)
37. Epson FX-1050 9 pins - 136 columns: FX-1050 (EPSON.DRV)
38. Epson FX-1170 9 pins - 136 columns: FX-1170 (EPSON.DRV)
39. Epson FX-286e 9 pins - 136 columns: FX-286e (EPSON.DRV)
40. Epson FX-850 9 pins - 80 columns: FX-850 (EPSON.DRV)
41. Epson FX-870 9 pins - 80 columns: FX-870 (EPSON.DRV)
42. Epson JX-80 Color 9 pins - 80 columns: JX-80 (EPSON.DRV)
43. Epson LQ-1010 24 pin - 132 column: LQ-1010 (EPSON.DRV)
44. Epson LQ-1050 (N9) 24 pins - 136 columns: LQ-1050 (N9) (EPSON.DRV)
45. Epson LQ-1050 24 pins - 136 columns: LQ-1050 (EPSON.DRV)
46. Epson LQ-1070 24 pins - 136 columns: LQ-1070 (EPSON.DRV)
47. Epson LQ-1170 24 pins - 136 columns: LQ-1170 (EPSON.DRV)
48. Epson LQ-2500 Color 24 pins - 136 columns: LQ-2500 (EPSON.DRV)
49. Epson LQ-2550 Color 24 pins - 136 columns: LQ-2550 (EPSON.DRV)
50. Epson LQ-500 24 pins - 80 columns: LQ-500 (EPSON.DRV)
51. Epson LQ-510 24 pins - 80 columns: LQ-510 (EPSON.DRV)
52. Epson LQ-570 24 pins - 80 columns: LQ-570 (EPSON.DRV)
53. Epson LQ-850 (N9) 24 pins - 80 columns: LQ-850 (N9) (EPSON.DRV)
54. Epson LQ-850 24 pins - 80 columns: LQ-850 (EPSON.DRV)
55. Epson LQ-860 Color 24 pins - 80 columns: LQ-860 (EPSON.DRV)
56. Epson LQ-870 24 pins - 80 columns: LQ-870 (EPSON.DRV)
57. Epson LQ-950 (N9) 24 pins - 110 columns: LQ-950 (N9) (EPSON.DRV)
58. Epson LX-800 9 pins - 80 columns: LX-800 (EPSON.DRV)
59. Epson LX-810 9 pins - 80 columns: LX-810 (EPSON.DRV)
60. Epson Stylus 800 Inkjet: Stylus 800 (EPSON.DRV)
61. Generic PostScript Printer: Generic PostScript Printer (PSCRIPT.DRV)
62. HP 7470A Plotter: HP7470A (PLOTTERS.DRV)
63. HP 7475A Plotter: HP7475A (PLOTTERS.DRV)
64. HP 7550A Plotter: HP7550A (PLOTTERS.DRV)
65. HP 7580A Plotter: HP7580A (PLOTTERS.DRV)
66. HP 7580B Plotter: HP7580B (PLOTTERS.DRV)
67. HP 7585A Plotter: HP7585A (PLOTTERS.DRV)

68. HP 7585B Plotter: HP7585B (PLOTTERS.DRV)
69. HP 7586B Plotter: HP7586B (PLOTTERS.DRV)
70. HP ColorPro: HP7440A (PLOTTERS.DRV)
71. HP DeskJet: HP DeskJet (HPDJPM.DRV)
72. HP DeskJet 1200C: HP DeskJet 1200C (PSCRIPT.DRV)
73. HP DeskJet 500: HP DeskJet 500 (HPDJPM.DRV)
74. HP DeskJet 500 in Epson EPL-6000 mode: HP DeskJet 500 (EPSON.DRV)
75. HP DeskJet 500C: HP DeskJet 500C (HPDJPM.DRV)
76. HP DeskJet 510: HP DeskJet 510 (HPDJPM.DRV)
77. HP DeskJet 550C: HP DeskJet 550C (HPDJPM.DRV)
78. HP DeskJet Plus: HP DeskJet Plus (HPDJPM.DRV)
79. HP DeskJet Portable: HP DeskJet Portable (HPDJPM.DRV)
80. HP DraftMaster I: HP7595A (PLOTTERS.DRV)
81. HP DraftMaster II: HP7596A (PLOTTERS.DRV)
82. HP DraftPro: HP7570A (PLOTTERS.DRV)
83. HP LaserJet 2000: HP LaserJet 2000 (LASERJET.DRV)
84. HP LaserJet 4: HP LaserJet 4 (LASERJET.DRV)
85. HP LaserJet 4/4M PS v2011_110: HP LaserJet 4/4M PS v2011_110 (PSCRIPT.DRV)
86. HP LaserJet 4L: HP LaserJet 4L (LASERJET.DRV)
87. HP LaserJet 4M: HP LaserJet 4M (LASERJET.DRV)
88. HP LaserJet 4Si: HP LaserJet 4Si (LASERJET.DRV)
89. HP LaserJet 4Si Mx: HP LaserJet 4Si Mx (LASERJET.DRV)
90. HP LaserJet 4Si/4Si Mx PS v2011: HP LaserJet 4Si/4Si Mx PS v2011 (PSCRIPT.DRV)
91. HP LaserJet 500 Plus: HP LaserJet 500 Plus (LASERJET.DRV)
92. HP LaserJet Classic: HP LaserJet Classic (LASERJET.DRV)
93. HP LaserJet IID: HP LaserJet IID (LASERJET.DRV)
94. HP LaserJet IID v52_2: HP LaserJet IID v52_2 (PSCRIPT.DRV)
95. HP LaserJet III: HP LaserJet III (LASERJET.DRV)
96. HP LaserJet III Cartridge Plus: HP LaserJet III Cartridge Plus (PSCRIPT.DRV)
97. HP LaserJet III v52_2: HP LaserJet III v52_2 (PSCRIPT.DRV)
98. HP LaserJet IIID: HP LaserJet IIID (LASERJET.DRV)
99. HP LaserJet IIID Cartridge Plus: HP LaserJet IIID Cartridge Plus (PSCRIPT.DRV)
100. HP LaserJet IIID v52_2: HP LaserJet IIID v52_2 (PSCRIPT.DRV)
101. HP LaserJet IIIP: HP LaserJet IIIP (LASERJET.DRV)
102. HP LaserJet IIIP Cartridge Plus: HP LaserJet IIIP Cartridge Plus (PSCRIPT.DRV)
103. HP LaserJet IIIP PS v52_2: HP LaserJet IIIP PS v52_2 (PSCRIPT.DRV)

104. HP LaserJet IIISi: HP LaserJet IIISi (LASERJET.DRV)
105. HP LaserJet IIISi PS v52_3: HP LaserJet IIISi PS v52_3 (PSCRIPT.DRV)
106. HP LaserJet IIP: HP LaserJet IIP (LASERJET.DRV)
107. HP LaserJet IIP Plus: HP LaserJet IIP Plus (LASERJET.DRV)
108. HP LaserJet IIP v52_2: HP LaserJet IIP v52_2 (PSCRIPT.DRV)
109. HP LaserJet Plus: HP LaserJet Plus (LASERJET.DRV)
110. HP LaserJet Series II: HP LaserJet Series II (LASERJET.DRV)
111. HP PaintJet Driver by Micrografx: Paintjet (SMGXPJET.DRV)
112. HP PaintJet Driver by Micrografx: Paintjet XL (SMGXPJET.DRV)
113. HP PaintJet XL300 PS v2011_112: HP PaintJet XL300 PS v2011_112 (PSCRIPT.DRV)
114. IBM 2380 PPS II: IBM 2380 PPS II (IBM42XX.DRV)
115. IBM 2381 PPS II: IBM 2381 PPS II (IBM42XX.DRV)
116. IBM 2390 PPS II: IBM 2390 PPS II (IBM42XX.DRV)
117. IBM 2390 PS/1: IBM 2390 PS/1 (IBM42XX.DRV)
118. IBM 2391 PPS II: IBM 2391 PPS II (IBM42XX.DRV)
119. IBM 3816 - 01D: IBM 3816 - 01D (IBM52XX.DRV)
120. IBM 3816 - 01S: IBM 3816 - 01S (IBM52XX.DRV)
121. IBM 4019 LaserPrinter: IBM 4019 LaserPrinter (IBM4019.DRV)
122. IBM 4019 LaserPrinter E: IBM 4019 LaserPrinter E (IBM4019.DRV)
123. IBM 4019 Laserprinter: IBM 4019 Laserprinter (LASERJET.DRV)
124. IBM 4019 Laserprinter E: IBM 4019 Laserprinter E (LASERJET.DRV)
125. IBM 4019 v52_1 (17 Fonts): IBM 4019 v52_1 (17 Fonts) (PSCRIPT.DRV)
126. IBM 4019 v52_1 (39 Fonts): IBM 4019 v52_1 (39 Fonts) (PSCRIPT.DRV)
127. IBM 4029 (17 Fonts 300 Dpi): IBM 4029 (17 Fonts 300 Dpi) (PSCRIPT.DRV)
128. IBM 4029 (17 Fonts 600 Dpi): IBM 4029 (17 Fonts 600 Dpi) (PSCRIPT.DRV)
129. IBM 4029 (39 Fonts 300 Dpi): IBM 4029 (39 Fonts 300 Dpi) (PSCRIPT.DRV)
130. IBM 4029 (39 Fonts 600 Dpi): IBM 4029 (39 Fonts 600 Dpi) (PSCRIPT.DRV)
131. IBM 4029 LaserPrinter 10: IBM 4029 LaserPrinter 10 (IBM4019.DRV)
132. IBM 4029 LaserPrinter 10L: IBM 4029 LaserPrinter 10L (IBM4019.DRV)
133. IBM 4029 LaserPrinter 10P: IBM 4029 LaserPrinter 10P (IBM4019.DRV)
134. IBM 4029 LaserPrinter 5E: IBM 4029 LaserPrinter 5E (IBM4019.DRV)
135. IBM 4029 LaserPrinter 6: IBM 4029 LaserPrinter 6 (IBM4019.DRV)
136. IBM 4029 LaserPrinter 6P: IBM 4029 LaserPrinter 6P (IBM4019.DRV)
137. IBM 4029 Laserprinter 10: IBM 4029 Laserprinter 10 (LASERJET.DRV)
138. IBM 4029 Laserprinter 10L: IBM 4029 Laserprinter 10L (LASERJET.DRV)
139. IBM 4029 Laserprinter 5E: IBM 4029 Laserprinter 5E (LASERJET.DRV)

140. IBM 4029 Laserprinter 6: IBM 4029 Laserprinter 6 (LASERJET.DRV)
141. IBM 4039 LaserPrinter (300 Dpi): IBM 4039 LaserPrinter (300 Dpi) (PSCRIPT.DRV)
142. IBM 4039 LaserPrinter (600 Dpi): IBM 4039 LaserPrinter (600 Dpi) (PSCRIPT.DRV)
143. IBM 4070 IJ: IBM 4070 IJ (IBM42XX.DRV)
144. IBM 4072 ExecJet: IBM 4072 ExecJet (IBM42XX.DRV)
145. IBM 4079 Color Jetprinter PS: IBM 4079 Color Jetprinter PS (PSCRIPT.DRV)
146. IBM 4201 Proprinter: IBM 4201 Proprinter (IBM42XX.DRV)
147. IBM 4201 Proprinter II: IBM 4201 Proprinter II (IBM42XX.DRV)
148. IBM 4201 Proprinter III: IBM 4201 Proprinter III (IBM42XX.DRV)
149. IBM 4202 Proprinter II XL: IBM 4202 Proprinter II XL (IBM42XX.DRV)
150. IBM 4202 Proprinter III XL: IBM 4202 Proprinter III XL (IBM42XX.DRV)
151. IBM 4202 Proprinter XL: IBM 4202 Proprinter XL (IBM42XX.DRV)
152. IBM 4207 Proprinter X24: IBM 4207 Proprinter X24 (IBM42XX.DRV)
153. IBM 4207 Proprinter X24E: IBM 4207 Proprinter X24E (IBM42XX.DRV)
154. IBM 4208 Proprinter XL24: IBM 4208 Proprinter XL24 (IBM42XX.DRV)
155. IBM 4208 Proprinter XL24E: IBM 4208 Proprinter XL24E (IBM42XX.DRV)
156. IBM 4216-031 v51_4 SheetFeed: IBM 4216-031 v51_4 SheetFeed (PSCRIPT.DRV)
157. IBM 4224 - 01 & 02 & E3: IBM 4224 - 01 & 02 & E3 (IBM42XX.DRV)
158. IBM 4224 - C2: IBM 4224 - C2 (IBM42XX.DRV)
159. IBM 4226 Model 302: IBM 4226 Model 302 (IBM42XX.DRV)
160. IBM 5183 Portable Printer: IBM 5183 Portable Printer (EPSON.DRV)
161. IBM 5201 Quietwriter II (IBM52012.DRV)
162. IBM 5202 QuietWriter III: IBM 5202 QuietWriter III (IBM52XX.DRV)
163. IBM 5204 QuickWriter: IBM 5204 QuickWriter (IBM52XX.DRV)
164. IBM 6180 Plotter: IBM6180 (PLOTTERS.DRV)
165. IBM 6182 Plotter: IBM6182 (PLOTTERS.DRV)
166. IBM 6184 Plotter: IBM6184 (PLOTTERS.DRV)
167. IBM 6186-1 Plotter: IBM6186-1 (PLOTTERS.DRV)
168. IBM 6186-2 Plotter: IBM6186-2 (PLOTTERS.DRV)
169. IBM 7371 Plotter: IBM7371 (PLOTTERS.DRV)
170. IBM 7372 Plotter: IBM7372 (PLOTTERS.DRV)
171. IBM 7374 Plotter: IBM7374 (PLOTTERS.DRV)
172. IBM 7375-1 Plotter: IBM7375-1 (PLOTTERS.DRV)
173. IBM 7375-2 Plotter: IBM7375-2 (PLOTTERS.DRV)
174. IBM NULL Printer Driver (IBMNULL.DRV)
175. IBM Personal Page Printer II-30: IBM Personal Page Printer II-30 (PSCRIPT.DRV)

176. IBM Personal Page Printer II-31: IBM Personal Page Printer II-31 (PSCRIPT.DRV)
177. IBM Personal Pageprinter: IBM Personal Pageprinter (PSCRIPT.DRV)
178. Kyocera F-1000A/F-1000: Kyocera F-1000A/F-1000 (LASERJET.DRV)
179. Kyocera F-1800A/F-1800: Kyocera F-1800A/F-1800 (LASERJET.DRV)
180. Kyocera F-2000A/F-2200S: Kyocera F-2000A/F-2200S (LASERJET.DRV)
181. Kyocera F-3000A/F-3300: Kyocera F-3000A/F-3300 (LASERJET.DRV)
182. Kyocera F-5000A/F-5000: Kyocera F-5000A/F-5000 (LASERJET.DRV)
183. Kyocera F-800A/F-800: Kyocera F-800A/F-800 (LASERJET.DRV)
184. Kyocera F-820: Kyocera F-820 (LASERJET.DRV)
185. Kyocera FS-1500A/FS-1500: Kyocera FS-1500A/FS-1500 (LASERJET.DRV)
186. Kyocera FS-3500A/FS-3500: Kyocera FS-3500A/FS-3500 (LASERJET.DRV)
187. Kyocera FS-5500A/FS-5500: Kyocera FS-5500A/FS-5500 (LASERJET.DRV)
188. Kyocera FS-850A/FS-850: Kyocera FS-850A/FS-850 (LASERJET.DRV)
189. Kyocera P-2000: Kyocera P-2000 (PSCRIPT.DRV)
190. Kyocera Q-8010: Kyocera Q-8010 (PSCRIPT.DRV)
191. Linotronic 100 v38_0: Linotronic 100 v38_0 (PSCRIPT.DRV)
192. Linotronic 100 v42_5: Linotronic 100 v42_5 (PSCRIPT.DRV)
193. Linotronic 200 v47_1: Linotronic 200 v47_1 (PSCRIPT.DRV)
194. Linotronic 200 v49_3: Linotronic 200 v49_3 (PSCRIPT.DRV)
195. Linotronic 300 v47_0: Linotronic 300 v47_0 (PSCRIPT.DRV)
196. Linotronic 300 v47_1: Linotronic 300 v47_1 (PSCRIPT.DRV)
197. Linotronic 300 v49_3: Linotronic 300 v49_3 (PSCRIPT.DRV)
198. Linotronic 500 v49_3: Linotronic 500 v49_3 (PSCRIPT.DRV)
199. NEC LC-890: NEC LC-890 (PSCRIPT.DRV)
200. Olivetti LP 5000: Olivetti LP 5000 (PSCRIPT.DRV)
201. Panasonic KX-P1123 in Epson LQ-850 mode: Panasonic KX-P1123 (EPSON.DRV)
202. Panasonic KX-P1124 in Epson LQ-2500 mode: Panasonic KX-P1124 (EPSON.DRV)
203. Panasonic KX-P1124i in Epson LQ-850 mode: Panasonic KX-P1124i (EPSON.DRV)
204. Panasonic KX-P1180 in Epson FX-86e mode: Panasonic KX-P1180 (EPSON.DRV)
205. Panasonic KX-P1191 in Epson FX-86e mode: Panasonic KX-P1191 (EPSON.DRV)
206. Panasonic KX-P1624 in Epson LQ-2500 mode: Panasonic KX-P1624 (EPSON.DRV)
207. Panasonic KX-P1654 in Epson LQ-1050 mode: Panasonic KX-P1654 (EPSON.DRV)
208. Panasonic KX-P1695 in Epson FX-1050 mode: Panasonic KX-P1695 (EPSON.DRV)
209. Panasonic KX-P2123 in Epson LQ-860 mode: Panasonic KX-P2123 (EPSON.DRV)
210. Panasonic KX-P2124 in Epson LQ-860 mode: Panasonic KX-P2124 (EPSON.DRV)
211. Panasonic KX-P2180 in Epson LX-850 mode: Panasonic KX-P2180 (EPSON.DRV)

212. Panasonic KX-P2624 in Epson LQ-1050 mode: Panasonic KX-P2624 (EPSON.DRV)
213. Panasonic KX-P4410: Panasonic KX-P4410 (LASERJET.DRV)
214. Panasonic KX-P4420: Panasonic KX-P4420 (LASERJET.DRV)
215. Panasonic KX-P4430: Panasonic KX-P4430 (LASERJET.DRV)
216. Panasonic KX-P4450: Panasonic KX-P4450 (LASERJET.DRV)
217. Panasonic KX-P4450i: Panasonic KX-P4450i (LASERJET.DRV)
218. Panasonic KX-P4455 v51_4: Panasonic KX-P4455 v51_4 (PSCRIPT.DRV)
219. Phaser Card v1_1: Phaser Card v1_1 (PSCRIPT.DRV)
220. QMS 1725 Print System: QMS 1725 Print System (PSCRIPT.DRV)
221. QMS 860 Print System: QMS 860 Print System (PSCRIPT.DRV)
222. QMS ColorScript 100: QMS ColorScript 100 (PSCRIPT.DRV)
223. QMS ColorScript 100 Mod 10: QMS ColorScript 100 Mod 10 (PSCRIPT.DRV)
224. QMS ColorScript 100 Mod 30: QMS ColorScript 100 Mod 30 (PSCRIPT.DRV)
225. QMS ColorScript 100 Mod 30si: QMS ColorScript 100 Mod 30si (PSCRIPT.DRV)
226. QMS ColorScript 210: QMS ColorScript 210 (PSCRIPT.DRV)
227. QMS ColorScript 230: QMS ColorScript 230 (PSCRIPT.DRV)
228. QMS IS X320T: QMS IS X320T (PSCRIPT.DRV)
229. QMS-PS 1500: QMS-PS 1500 (PSCRIPT.DRV)
230. QMS-PS 1700: QMS-PS 1700 (PSCRIPT.DRV)
231. QMS-PS 2000: QMS-PS 2000 (PSCRIPT.DRV)
232. QMS-PS 2200: QMS-PS 2200 (PSCRIPT.DRV)
233. QMS-PS 2210: QMS-PS 2210 (PSCRIPT.DRV)
234. QMS-PS 2220: QMS-PS 2220 (PSCRIPT.DRV)
235. QMS-PS 410: QMS-PS 410 (PSCRIPT.DRV)
236. QMS-PS 800: QMS-PS 800 (PSCRIPT.DRV)
237. QMS-PS 800 Plus: QMS-PS 800 Plus (PSCRIPT.DRV)
238. QMS-PS 810: QMS-PS 810 (PSCRIPT.DRV)
239. QMS-PS 810 Turbo: QMS-PS 810 Turbo (PSCRIPT.DRV)
240. QMS-PS 815: QMS-PS 815 (PSCRIPT.DRV)
241. QMS-PS 815 MR: QMS-PS 815 MR (PSCRIPT.DRV)
242. QMS-PS 820: QMS-PS 820 (PSCRIPT.DRV)
243. QMS-PS 820 Turbo: QMS-PS 820 Turbo (PSCRIPT.DRV)
244. QMS-PS 825: QMS-PS 825 (PSCRIPT.DRV)
245. QMS-PS 825 MR: QMS-PS 825 MR (PSCRIPT.DRV)
246. Qume ScripTEN: Qume ScripTEN (PSCRIPT.DRV)
247. Seiko ColorPoint PS Model 04: Seiko ColorPoint PS Model 04 (PSCRIPT.DRV)

248. Seiko ColorPoint PS Model 14: Seiko ColorPoint PS Model 14 (PSCRIPT.DRV)
249. Seiko Personal ColorPoint PS: Seiko Personal ColorPoint PS (PSCRIPT.DRV)
250. Seiko Personal ColorPoint PSE: Seiko Personal ColorPoint PSE (PSCRIPT.DRV)
251. Silentwriter LC 890XL v50_5: Silentwriter LC 890XL v50_5 (PSCRIPT.DRV)
252. Silentwriter2 290 v52_0: Silentwriter2 290 v52_0 (PSCRIPT.DRV)
253. Silentwriter2 Model 90 v52_2: Silentwriter2 Model 90 v52_2 (PSCRIPT.DRV)
254. TI 2115 (13 fonts) v47_0: TI 2115 (13 fonts) v47_0 (PSCRIPT.DRV)
255. TI OmniLaser 2108: TI OmniLaser 2108 (PSCRIPT.DRV)
256. TI Omnilaser 2115: TI Omnilaser 2115 (PSCRIPT.DRV)
257. TI microLaser PS17 v_52_1: TI microLaser PS17 v_52_1 (PSCRIPT.DRV)
258. TI microLaser PS35 v_52_1: TI microLaser PS35 v_52_1 (PSCRIPT.DRV)
259. Tektronix Phaser 200e 17 fonts: Tektronix Phaser 200e 17 fonts (PSCRIPT.DRV)
260. Tektronix Phaser 200e 39 fonts: Tektronix Phaser 200e 39 fonts (PSCRIPT.DRV)
261. Tektronix Phaser 200i v2011_108: Tektronix Phaser 200i v2011_108 (PSCRIPT.DRV)
262. Tektronix Phaser II PX v2_02: Tektronix Phaser II PX v2_02 (PSCRIPT.DRV)
263. Tektronix Phaser II PXe 17 font: Tektronix Phaser II PXe 17 font (PSCRIPT.DRV)
264. Tektronix Phaser II PXe 39 font: Tektronix Phaser II PXe 39 font (PSCRIPT.DRV)
265. Tektronix Phaser II PXi v2010: Tektronix Phaser II PXi v2010 (PSCRIPT.DRV)
266. Tektronix Phaser III PXi v2010: Tektronix Phaser III PXi v2010 (PSCRIPT.DRV)
267. Tektronix Phaser IISD v2011: Tektronix Phaser IISD v2011 (PSCRIPT.DRV)
268. Varityper VT-600: Varityper VT-600 (PSCRIPT.DRV)
269. Wang LCS15: Wang LCS15 (PSCRIPT.DRV)
270. Wang LCS15 FontPlus: Wang LCS15 FontPlus (PSCRIPT.DRV)

