OREGON STATE UNIVERSITY
COMPUTER CENTER
Corvallis, Oreg.

C S 3 REFERENCE MANUAL

By
Mr. G. Bachelor

March 27, 1968
cc-68-3

Revisions to OS3 User's Manual

March 15, 1968

A number of changes have been made in OS3 since this manual was issued October 25, 1967. Below we list changes to be made in various sections. A new section (11) has been added, describing OS3 error messages. Sections 6 through 10 have not yet been written.

Section 2

Page 2.1. Add to first paragraph: A batch job may also use magnetic tapes. On-line jobs cannot use magnetic tapes.

Page 2.4. Delete the portion headed "Limit on concurrent use" and substitute the following:

Concurrent use.

If two or more users (batch or on-line) are using the same account/user number pair simultaneously, they may not use the same saved file at the same time unless the file is file-protected. File protection may not be removed from a saved file while more than one user is equipped to it.

Section 3

Page 3.1. In the description of the EQUIP statement, add the new "type":

MT, (tape no.) (comments to operator)

to equip a magnetic tape. Available only for batch jobs.

Example:

EQUIP,15=MT,1234 NO WRITE RING, 556 BPI.

The MT equip statement is printed on the console typewriter to tell the operator to mount the specified tape.

Page 3.2. Add to the description of the FP statement:

It is illegal to file protect a logical unit in the range 50 to 59 unless the unit is a saved file.

Page 3.4. Add to the description of the RFP statement:

It is also illegal to remove file protection from a saved file which is simultaneously equipped by another user with the same account/user numbers.

Page 3.5. Delete the last sentence of the first paragraph and add the following: It is illegal to unequip units in the range 59 to 61, inclusive. It is also illegal to unequip units that are file protected but not saved.

Section 4

Page 4.2. The following parameters are now available on the FORTRAN statement, in addition to those listed on page 4.2:

C = (lun) to produce COMPASS source language output on the specified unit. Uses unit 62 if (lun) is omitted.

D = (lun) to cause diagnostic messages to be printed on (lun) instead of the listing (L) unit.

R = (lun) to cause loading and execution of binary output on (lun) if no serious errors are encountered. Uses unit 56 if (lun) is omitted.

Also, the I (input) parameter may specify the name of a saved file instead of a logical unit number. In this case, FORTRAN will use l.u.n. 58 to equip the saved file.

Page 4.3. Some additional features have been added to the loader. These include:

a. Units in the range 50 to 58 are rewound before and after loading.

b. All of lower memory (locations 00000 through 77777 inclusive) is set to zero when the loader is called.

c. The loader can now generate overlays and segments. An overlay prepared by the loader can be loaded by the OVLOAD statement.

d. A BCD card with the word FINIS on it will terminate loading. This is used when one wishes to load a program but not run it.

Section 5

Two new Utility statements have been added:

DATE,(lun)

Writes one record on (lun), giving the current date and time. If (lun) is omitted, unit 61 is used.

OVLOAD,(lun)

If (lun) is omitted, logical unit 0 is used. Rewinds the unit and loads an overlay from it, then jumps to the indicated transfer address. (Overlays are prepared by the loader.)

Section 1

INTRODUCTION

"OS^3" is an abbreviation for "OSOSOS", which is an acronym for "Oregon State Open Shop Operating System". This is a time-sharing operating system developed at Oregon State University for the CDC 3300, which features remote, on-line access by many users at the same time. OS^3 is a general-purpose time-sharing system, which provides its users with numerous facilities. The systems available to the user include the following:

COMPASS	Comprehensive Assembler.
COSY	Compressed Symbolic deck processor.
EDIT	An on-line editing routine.
FORTRAN	Compiler.
LIBEDIT	Prepares program libraries.
LOAD	Loads binary relocatable programs produced by ALGOL, COMPASS, and FORTRAN.
OSCAR	An on-line computing service.
RADAR	An on-line debugging aid, including a simple assembler and dis-assembler.
UTILITY	A routine that provides a number of services, including file manipulation, copying of files, generating octal memory dumps, etc.

OS^3 uses the "executive" hardware of the CDC 3300 and sophisticated software techniques to provide each user with a simulated computer that has 65,536 words of storage and a very reliable, easy-to-use input/output/file-handling system. It can handle up to 32 users at remote stations (teletypes), plus one "batch" user (via the card reader), all at the same time. Core storage and back-up (disk) storage are allocated and released dynamically by the OS^3 system. The user need be concerned only with the total amount of file space and the total time he needs for his job. If he is an on-line (remote) user, he can expand these space and time limits if necessary, and continue running. A batch user must allow enough time and

file space to handle his job, since if either is exhausted, the job is terminated. It does not hurt if a user allows himself more file space than he needs, since only the space he actually uses is allocated to him.

OS^3 control statements are more like those of MSOS than those of MASTER. Files and input/output devices are referred to by logical unit numbers (0 to 99), and names are used to retrieve saved files. FORTRAN users will find little difference in using OS^3, as compared with other operating systems, except that larger programs can be run under OS^3. Programmers who use COMPASS will find input, output, and file manipulation very easy to do. One unusual feature of OS^3 is that there is no "resident" system in the user's memory. The user's memory space consists of 65,536 words, and it is all available for him to use. This feature (among others) makes it possible to run larger programs under OS^3 than can be run under any other operating system available.

OS^3 accomplishes its remarkable feats by the use of list processing techniques to handle storage allocation, and "time-slicing" and "memory swapping" to handle numerous on-line users. Time-slicing means that each active user program is run for a limited period of time. Then another program is run for a while, etc., on a 'round-robin" basis. Programs which are waiting for input or output are temporarily suspended. Memory swapping means that not all of a user's program need be in core memory at once. Portions which are not needed at the moment are "swapped out" (written out on back-up disk memory), and part of another program is "swapped in" to the memory space made available. When an on-line user's program is waiting for him to type an input, it is quite likely that the entire program will be swapped out, to allow other users' programs to be run. A program is also suspended and may be swapped out when it has generated enough output to keep the user's teletype busy for a while. To each user, it seems as though he has a computer

all to himself, which, of course, appears to be somewhat slower than the actual computer, since the user is sharing the computer's ability with other users.

Unlike most other operating systems, the OS[3] system controls all input/output devices and provides error recovery techniques. It can recover from most hardware problems without seriously inconveniencing the users. Even a memory parity error is not necessarily "fatal".

Section 2

OS^3 JOB PARAMETERS

"Jobs" processed by OS^3 are of two kinds: (a) on-line jobs, and (b) batch jobs. A batch job consists of a deck of cards submitted by the user to the Computer Center. The computer operator places this deck in the card reader, along with other job decks. The OS^3 system reads in the decks and processes each job in turn. (Batch jobs are not time-shared with one another.) All outputs generated by a batch job are stored in files (on the disks). When a job is completed, by reading the JOB card for the next job, the output files for the finished job are scheduled for printing, punching, or plotting.

An on-line job is controlled from a remote station (teletype). It is initiated by a LOGIN (or LOGON) statement, typed by the user. The user may call various systems during a session at the teletype. Outputs from the OS^3 system or from user programs that are directed to the user's teletype are typed immediately. Outputs to be printed, punched, or plotted are stored in files (just as for a batch job), and are scheduled for output when the user logs off. The end of an on-line job is signified when the user types the control statement LOGOFF (or LOGOUT).

There are several parameters associated with each job, which control accounting for the job and access to files, and which place limits on time and file space. These parameters are discussed in the following paragraphs.

Account and user numbers

The account number (or "job" number), and user number (or code) appear on the JOB card for a batch job, or in the LOGIN statement for an on-line job. The account number is

issued by the Computer Center and is used for billing purposes. (At the present time, this is a 5-digit number.) The user number or code is selected by the user. It may be a number (less than 8000000), or a code of no more than 4 letters or digits, of which the first character must be a letter. There may be many user numbers associated with a single account number. For example, each student in a class could have his own user number or code, but all the students would use the same account number. Persons wishing to use OS^3 should contact the Computer Center to make arrangements for having their account number and user numbers entered into the OS^3 accounting system.

When a user saves a file under OS^3, it is saved with 3 items of identification: the account number, the user number, and the name given to the file by the user. A saved file can only be accessed by using the same numbers and names under which it was saved. Such files are called "private" files. Only the user who saved the file can read it, change it, or delete it.

There are also "public" files, which can be read by any user, but which cannot be changed or deleted (except by using special "public" account and user numbers). One important public file is FTNLIB, which contains the FORTRAN library routines.

Time limits

Associated with each account/user number pair, is a "total time limit". This sets an upper limit on the amount of computer time which can be used. The user determines this limit, according to the amount of funds available, or other criteria. This limit can be changed when necessary, by contacting the Computer Center.

Each job also has a time limit. This limit is normally 60 seconds (or less if the total time limit is being approached). The job time limit may be set to some other value (smaller or greater than 60) by using the TIME statement.

When the amount of computer time used by a job becomes equal to the job time limit, the OS^3 system prints the message "TIME CUT" on logical unit 61 (printed output for a batch job; teletype for an on-line job). In the case of a batch job, the job is terminated after this message. In the case of an on-line job, the system enters OS^3 control mode. The user can then type a TIME statement to increase his job time limit, followed by the statement GO to resume execution of his program. Or, if he suspects that his program is in a loop, he may do something else, such as using the STATUS statement, or calling DUMP or RADAR, etc. (It would be necessary to increase the time limit if DUMP or RADAR were used.)

File space

File space is handled by OS^3 in units called "file blocks". A file block holds about 500 words, or 2000 characters of information. (The actual quantity depends on the lengths of records in the file.)

Associated with each account/user number pair, are two file space limits. The "scratch file block limit" determines the maximum number of scratch and output file blocks which may be used. The "saved file block limit" determines the maximum number of file blocks which may be saved under the account/user number pair. These limits are selected by the user and may be changed by contacting the Computer Center.

Each job also has a scratch file block limit, which is normally 100, but may be set to some other value by using the MFBLKS statement. This limit may not exceed the scratch file block limit associated with the account/user numbers.

Note that the job scratch file limit includes not only scratch files, but also outputs from the job (other than teletype outputs to an on-line user). If a user program attempts to write a record on a scratch file or on an output device, that would cause the total file blocks used to exceed the limit, the OS^3 system prints the message "INSUFFICIENT FILE SPACE" on unit 61 (line printer or teletype). In the case of a batch job, the job is terminated. In the case of an on-line job, the system enters OS^3 control mode. The user can type an MFBLKS statement to increase his job file limit, and then GO, or he may take other action (such as releasing some files he no longer needs, etc.)

Limit on concurrent use

If the saved file block limit for an account/user number pair is non-zero, it is prohibited for two users to be using this same account/user number pair at the same time. The reason is that one of the users could save a file, and the other could delete it, and the result would be chaos in the OS^3 file system. Hence, if one job is active under a given account/user number pair, no other job (whether batch or on-line) can be initiated under the same pair of numbers. An attempt to do so will result in an error message.

The only exception to this rule is in the case of an account/user number pair for which the saved file block limit is zero. In this case, no files can be saved, and the file system cannot be destroyed by multiple users. Hence, when the saved file block limit is zero, it is permissible for more than one job to be active at the same time under the account/user number pair in question.

Section 3

OS^3 CONTROL STATEMENTS

The control statements recognized by the OS^3 system are listed below alphabetically. In the case of an on-line user, control statements are processed only when the user is in control mode, which is indicated when the pounds sign (#) is printed out. For a batch job, a control statement is recognized when the card has a 7,8 punch in Column 1.

DELETE,(name)

The file whose name is given is deleted from the file directory. (name) must be equipped to some logical unit, and the file must not be file protected. This statement does not destroy the information in the file; however, if the file is not subsequently saved, it will be released at the end of the job, or Logoff.

EQUIP,(lun)=(element)

Logical unit (lun) is defined to be a file or an input/output device. (lun) must be an integer in the range 0 to 99. The (element) may be one of three kinds:

xx where xx is the logical unit number of a previously equipped unit. This unit may now be referred to by either of the numbers.

type where type is one of the following:

FILE to equip a scratch file or a file that is to be saved.

PR or LP to equip a line printer as an output device.

PUN to equip a card punch as an output device.

PLOT to equip a plotter as an output device. (In the case of output data, the data is printed, punched, or plotted at end of job, or logoff, or when the unit is "unequipped".)

NULL to equip a "device" that absorbs any output given to it, and discards the information.

(name) where (name) is the name of a public file or a private file that has been saved under the account number and user numbers of the current job.

(Note: Only one logical unit may be equipped in each EQUIP statement.)

FP,(lun)

File protects the specified logical unit, which must be equipped as a file. This prevents writing on the file, or releasing it, or doing anything which would destroy the contents of the file.

GO

This statement is primarily for on-line use. When a user program has been interrupted, either deliberately, or as a result of an error, execution of the user program may be resumed by this statement.

JOB,(account number),(user number),(identification)

This control statement initiates a batch job. It must be the first card in the deck. The numbers are used for accounting purposes and to permit access to private files which have been saved under these numbers. The identification field is optional, but should be used to identify the deck, so it can be returned to the owner.

If the account number and user numbers are valid, the statement is accepted. The user's memory and register file are initialized to zero. Logical units 54, 55, and 56 are equipped as scratch files. Logical unit 60 is equipped as the card reader, 61 as the line printer, and 62 as the card punch.

The end of a batch job is signified when the next valid JOB card is read. User memory and unsaved files are removed, and output files are scheduled for output as soon as possible.

LOGIN,(account number),(user number),(identification)

Initiates an on-line job. The numbers are used for accounting purposes. They are also part of the identification for saved private files, to prevent unauthorized access to such files. The identification field is optional, and may contain any desired information.

If the account and user numbers are valid, the statement is accepted, and the user's register file (40_8 to 77_8) and memory are initialized to zero. Also, logical units 54, 55, and 56 are equipped as scratch files, and 60 and 61 are equipped as the user's teletype.

LOGON,(account number),(user number),(identification)

Same as the LOGIN statement.

LOGOFF

Terminates an on-line job. Removes all user memory and all files that were not saved. Any output files (print, punch, or plot) are scheduled for output as soon as the proper device is available. Closes out the accounting for the job. The teletype is now available for another user.

LOGOUT

Same as LOGOFF.

MFBLKS=(number)

Sets the maximum file blocks allowed for the job to the minimum of the number given and the number of file blocks allowed under the account numbers for the job. If the MFBLKS statement does not appear, 100 file blocks are allowed. (A file block holds about 500 words, or 2000 characters of information.)

MI

Manual Interrupt. The interrupt status is stored in the upper 8 bits of location 12_8, the current user program counter is stored in the lower 16 bits of this location, and the next instruction is read at location 13_8. Primarily for on-line use.

RFP,(lun)

Removes file protection from a file which has been protected by FP. It is illegal to remove file protection from a public file.

RMP

Removes memory protection, which may have been set on certain areas of memory by calling a library routine.

SAVE,(lun)=(name)

The file referenced by the specified logical unit number is placed in the file directory under the name given, and the account and user numbers for this job. It will be saved semi-permanently and may be retrieved later by the EQUIP statement. If (lun) was file protected, the saved file is also protected.

START,(address)

Causes an unconditional jump to the address given (in octal), and execution of a user program will begin at this location.

STATUS

The current status of the user's program is printed (on unit 61), giving the current values of the location counter, last jump address, registers A, Q, EU, EL, B1, B2, B3, and the Interrupt Status.

TIME

Causes the total amount of computer time used by the job since its initiation to be printed on unit 61.

TIME=(number)

Sets the maximum time allowed for the job to the minimum of the number given (seconds) and the number of seconds allowed under the account and user numbers. If no statement of this type appears, the time limit is 60 seconds. (This is actual computer time, and will usually be enough for a half-hour or longer session at a console.)

UNEQUIP,(lun)

Causes the logical unit referenced by (lun) to be "unequipped". If the unit is a file which has not been saved, the information

is removed from the system. If the unit is a saved file, it is saved, but cannot be referenced unless equipped again. If the unit is an output device (printer, punch, or plotter), the output data is scheduled for output as soon as the appropriate device is available. The logical unit number that has been unequipped can be used for some other purpose by using it in an EQUIP statement. It is illegal to unequip units in the range 50 to 61, inclusive.

If a control statement is not recognized as one of those listed above, it is examined to see if it is a library call statement. See the section on library calls. If it is not a library call statement either, the Utility routine is loaded into the user's upper memory and the statement is given to it. If Utility recognizes the statement as a valid utility call, the specified action is performed (see the section on the Utility routine). If Utility does not like the statement, an error message is printed (on unit 61).

Section 4

OS^3 LIBRARY ROUTINES

There are a number of library routines or systems available to users of OS^3. These routines are called by library call statements, which are recognized by the OS^3 control statement processor. The list below contains a brief description of the call statements. For further information on each library routine, see other documents or other sections of this manual. The general form of a library call is:

(name),(parameters)

The (name) is recognized by OS^3 and the specified library routine is copied into the user's memory, and given control. If the library routine uses parameters, it reads them as a string of characters in the ASCII code, using CTI instructions. A RETURN code (215_8) terminates the parameter string.

In the descriptions below, (lun) denotes a logical unit number, and (lunlist) denotes a sequence of one or more (lun)'s separated by commas.

COMPASS,I=(lun),P=(lun),X=(lun),L=(lun),R

The parameters may appear in any order, or may be omitted. The (lun) may be omitted, in which case a standard unit is assumed. The COMPASS assembler is copied into memory and processes symbolic source programs from unit I (60 if not specified). It generates binary relocatable output on units X (56 if unit not specified) and P (62 if not specified), provided that these parameters are present. If the L parameter is present, a listing of the program is prepared on unit L (61 if unit not specified), and if R is present, a cross-reference list of symbols is prepared on the same unit as L. If no L parameter appears, but there are diagnostics, a message will be printed on unit 61.

COMPASS uses unit 55 as a scratch file, and may also use unit 54.

COSY,I=(lun),L=(lun)

Calls the COSY (Compressed Symbolic) processor. If the L parameter is present, COSY lists the revision cards on the specified unit (61 if not specified). If it is absent, no log is printed. The I parameter, if present, specifies the unit from which COSY revisions and control cards will be read. If this parameter is absent, unit 60 will be used.

EDIT

Calls the on-line editing routine. There are no parameters on the call statement. EDIT can be used to prepare and modify files, which may then be used as input to other routines, such as COMPASS, FORTRAN, etc.

FORTRAN,I=(lun),X=(lun),L=(lun),P=(lun),A=(lun)

The parameters may appear in any order, or be omitted. The (lun) may be omitted, in which case a standard unit is assumed. The FORTRAN compiler is copied into memory and reads a source program from unit I (60 if not specified). It generates binary relocatable output on units X (56 if unit not specified) and P (62 if not specified), provided that these parameters are present. A listing of the program will be prepared on unit L (61 if not specified), if this parameter is present. If the A parameter is present, an assembly-language listing of the compiled program will be prepared on unit A (61 if not specified).

FORTRAN uses 54 and 55 as scratch files.

LIBEDIT,(lunlist),LIB=(lun)

Calls LIBEDIT, which reads binary relocatable programs from the units in the list. Then it generates a library file and writes it on the specified LIB unit (54 if not specified). This library file contains a directory, a file mark, then the binary programs and another file mark. This file is in the correct form for use by the loader. LIBEDIT uses 55 as a scratch file.

LOAD,(lunlist),LIB=(lun)

Calls the loader to load and link binary relocatable programs, reading them from the units specified. Each unit is read in turn until a file mark is encountered. (If unit 56 occurs, it is rewound before and after reading from it.) After reading programs from all the specified units, the loader reads from unit 60, where it may find more binary programs. Program loading is terminated by the reading of a BCD card from unit 60. (If there are no units specified, the loader immediately begins reading from unit 60.) When a BCD card is read from unit 60, the loader checks to see if there are any external symbols which have not yet been linked to entry points. If so, the specified library unit (63 if no LIB unit is specified), is examined, and programs are loaded from it as needed to satisfy the undefined symbols. When loading is finished, the BCD record read from unit 60 is examined. If it contains the word MAP (anywhere in the record), a memory map is generated and printed on unit 61. If it contains MAP,(lun) the map is generated and printed on (lun). In either case, another BCD record is read. If the first BCD record does not contain MAP, it must have the word RUN on it, or if it does, the second BCD record must have the word RUN. Otherwise, the loader will print an error message on unit 61.

If there are errors during loading the loader prints error messages on unit 61, and the program will not be run. If no errors occurred, and there were either one or two subprograms that had transfer symbols, the program is run. The loader transfers control to the transfer symbol (the second one if there were two).

OSCAR

There are no parameters. This calls the OSCAR routine, which is an on-line,""conversational", computing service.

RADAR

There are no parameters. Calls the RADAR routine, which is an on-line debugging aid, including a simple assembler and dis-assembler.

Section 5

OS^3 UTILITY ROUTINE

The Utility routine is loaded into the user's upper memory and given control when the OS^3 system encounters a control statement that it does not recognize. The Utility routine examines the statement, and if it is one of those listed below, carries out the action specified. If not, an error message is printed.

The notation (lunlist) in the descriptions below denotes a sequence of one or more logical unit numbers (luns), separated by commas or spaces.

AUTOLOAD,(lun)

If (lun) is omitted, logical unit 0 is assumed. Rewinds the unit if possible, then reads one record from it into memory locations starting at 0. The record can be as long as 32768 words (i.e., all of lower memory). Then the Utility routine jumps to location 0.

BKSPACE,(lunlist)

Each logical unit in the list is backspaced one record.

BKSP,(lunlist)

Same as BKSPACE.

BACKSPACE,(lunlist)

Same as BKSPACE.

CLEAR,(lunlist)

The status of each unit in the list is cleared. This sets to zero the bits which denote "file mark just read", "reverse read set", and "binary record processed". The bits denoting "file protected", "load point", and "end of data" are not affected.

COPY,IN=(lun),OUT=(lun),SHIFT=(number)

Copies BCD or binary information from a file or an input device to another file or an output device. Neither device is rewound by Utility before doing the copy. The three

parameters may appear in any order, or be omitted. The words can be spelled any way desired, so long as they start with the letters I, O, or S (a single letter is enough). If the IN parameter is omitted, unit 60 is assumed. If the OUT parameter is omitted, unit 61 is assumed. If the SHIFT parameter is omitted, 0 is assumed unless OUT is a line printer or teletype, in which case 1 is assumed. If IN is a file, information is copied from it until the end of data is encountered. If IN is an input device, information is copied from it until a control statement occurs. (BCD records, binary records, and file marks are all copied. The maximum record length allowed is about 60,000 words.) The information is copied to the OUT unit, after shifting it right SHIFT words. (If SHIFT is 1, the information in each record is shifted right one word, which is four characters.) The words inserted at the left are filled with blanks (60_8 code).

DUMP,(lun),(starting address),(ending address)

Generates a printable dump of lower memory from the starting address to the ending address (both expressed in octal). The contents of memory are printed in octal, 4 words per line if the output is going to a teletype, 8 words per line otherwise. The output is written on the specified logical unit. If the addresses are omitted, all of lower memory is dumped. If the logical unit is omitted too, the dump is printed on unit 61.

FWDSPACE,(lunlist)

Each logical unit in the list is spaced forward one record.

FWSP,(lunlist)

Same as FWDSPACE.

LABEL,(lun)/(message)

or

LABEL,(lun)'(message)

Outputs the given message on the specified logical unit. If (lun) is omitted, unit 61 is assumed. If the slash (/) precedes the message, a space is inserted in front of the

message, so that normal spacing control is provided for printing. However, if the slash is used, and the message is being output to the card punch, the first 6 characters (approximately) of the message are punched into the card in a displayed form, each letter or digit being represented by an appropriate group of holes. When the apostrophe (') is used, the message is simply copied directly to the specified logical unit, as a BCD record.

Since OS[3] does not print control cards, the LABEL statement should be used to label printed output. Punched output should also be labeled, if any is to be generated by the job.

RELEASE,(lunlist)

Each unit in the list is released. This may be done only to files, and causes all information in the file to be destroyed, and the storage space to be released.

REWIND,(lunlist)

Each unit in the list is rewound.

SBPFM,(lunlist)

Each logical unit in the list is spaced backward past a file mark, or to the load point if no file mark is encountered.

SEFB,(lunlist)

Same as SBPFM.

SFPFM,(lunlist)

Each logical unit in the list is spaced forward past a file mark, or to the end of data, if no file mark is encountered.

SEFF,(lunlist)

Same as SFPFM.

WFM,(lunlist)

Writes a file mark on each logical unit in the list. (A file mark on a printer unit will cause a page eject when the output is printed; a file mark on a punch unit

will cause the punching of an end-of-file card (7,8 punches in columns 1 and 2)).

WEOF,(lunlist)

Same as WFM.

Section 11

OS3 Error Messages

Non-fatal messages.

The following messages indicate that something is wrong with a control statement, and the message is printed on unit 61. However, most of these conditions do not cause a batch job to be terminated. In the case of an on-line job, the user is placed in control mode after one of these messages occurs.

DISALLOWED CONTROL STATEMENT FROM RUNNING PROGRAM

A user program has attempted to execute a control statement which is not allowed from a user program.

EQUIP ERROR

An equip statement is in incorrect format, or the unit number is already equipped, or one is trying to equate a unit to another unit that is not equipped.

EQUIP ERROR. NAME NOT FOUND.

An attempt has been made to equip a unit number to a saved file, and there is no file in the directory by the given name, accessible to the user.

FILE PROTECT VIOLATION.

An attempt has been made to delete a protected file, or to remove file protection from a public file, or to remove file protection from a file which is simultaneously being used by another user with the same account/user numbers, or to unequip a file protected file that is not saved.

ILLEGAL CONTROL STATEMENT.

The statement was recognized as a control statement, but the requested action cannot be carried out. Some examples: Attempting to delete a file that is not equipped; attempting to file protect a unit that is not a file; improper format of the control statement.

ILLEGAL JOB/USER NUMBER.

The account/user number pair on a JOB card is not in the Job file; or an attempt has been made to LOGIN under an account/user number pair that is not in the Job file. In the case of a batch job, the job is terminated.

ILLEGAL LOGON.

Any statement from a teletype which is not logged on, that does not start with LOGON or LOGIN, will cause this message.

INSUFFICIENT FILE SPACE.

An attempt has been made to save a file, and the total saved file blocks would exceed the limit allowed for the account/user numbers; or an attempt has been made to delete a saved file when the file space in the saved file, added to the current scratch file space, would exceed the scratch file limit. (See also the similar message in the "fatal" group.)

NAME ALREADY PRESENT.

An attempt has been made to save a file, and there already exists a saved file by the same name.

TIME XXXXX.XXX SECONDS. MFBLKS XXXX

This is not an error message. It is printed out when the control statement TIME is processed. (Not a TIME = (number) statement.) The message gives the total number of seconds of computer time (to the nearest thousandth of a second) that

TIME XXXXX.XXX SECONDS. MFBLKS XXXX (continued)

have been used so far in the current job, and the maximum number of scratch file blocks that have been in use at any time during the current job. If the TIME statement is used by an on-line user, the system remains in control mode; a GO statement would resume execution of an interrupted program. A TIME statement in a batch job terminates execution of any program that may be running, but does not terminate the job. (Following control statements are processed as usual.)

Fatal Messages.

The following messages indicate serious error conditions. Most of them are the result of an illegal action by a user program. If these conditions occur in a batch job, the job is terminated; the STATUS is printed out (see discussion at end), and the rest of the deck is skipped. In an on-line job, the user may take action to correct the condition, if possible, and resume what he was doing.

HLT yyyyy AT xxxxxx

A program attempted to execute a HLT yyyyy (Halt) instruction at location xxxxxx.

**ILLEGAL CONTROL STATEMENT

The statement was not recognized as a valid control statement. In a batch job, the statement is printed (approximately), and the job is terminated. In an on-line job, the user may re-type the statement, correcting the error.

ILLEGAL INSTRUCTION (yyyyyyyy) AT xxxxxx.

An attempt was made to execute the instruction yyyyyyyy at location xxxxxx. See discussion at end of this section on various types of illegal instructions.

ILLEGAL WRITE AT xxxxxx.

The instruction at location xxxxxx attempted to store into protected memory. Memory protection is set only on certain library routines (such as OSCAR, RADAR, FORTRAN, etc.), and an illegal write usually means a "bug" in a library program.

INSUFFICIENT FILE SPACE.

A user program attempted to write a record on a file or output device, which would have caused the scratch file block limit to be reached. If this happens to a batch job, the job is terminated; an on-line user may be able to correct the problem and go on. (By increasing the MFBLKS limit, for example).

(See also the similar message in the "non-fatal" group.)

LUN xx NOT DEFINED.

An attempt was made to read from, to write on, or to perform a control function (other than STATUS) on logical unit number xx, but this unit number has not been equipped. An on-line user can equip the undefined unit and type GO to resume execution.

MEMORY PARITY ERROR

A parity error occurred in the user's memory, or an irrecoverable parity error occurred during swapping between disk and the user's memory. This is, of course, a hardware failure.

SLS AT xxxxxx

An attempt was made to execute an SLS (Select Stop) instruction at location xxxxxx.

TAPE DRIVE FAILURE

A batch job, using a magnetic tape, ran into some difficulty with the tape unit.

TIME CUT

The job time limit has been reached. A batch job is terminated; an on-line user may set his job time limit to a larger value (with a TIME = xxx statement) and GO.

UCS AT xxxxxx

An attempt was made to execute a UCS (Unconditional Stop) instruction at location xxxxxx.

Status of a Job.

When a batch job is terminated by one of the fatal conditions listed above, or when an on-line user types the control statement STATUS, the current status of the user's program is printed out in the following form:

```
     P       xxxxxx
   LJA       xxxxxx
     A     xxxxxxxx
     Q     xxxxxxxx
    EU     xxxxxxxx
    EL     xxxxxxxx
    B1        xxxxx
    B2        xxxxx
    B3        xxxxx
    IS     xxxxxxxx
```

P is the program counter (16 bits), giving the location of the last instruction executed, or of the instruction to be executed next, depending on the situation. LJA is (usually) the location (16 bits) of the last jump instruction that jumped. A and Q denote the contents of the A and Q registers. EU and EL denote the contents of the upper and lower halves of the E register. B1, B2, and B3 give the contents of the index registers. IS is the internal status; the left-most 12 bits are the interrupt mask, bit 11 indicates a BCD fault, bit 10 denotes divide fault, bit 9 indicates arithmetic overflow, bit 8 indicates a floating point fault, and bit 0 indicates relocation with operand state is set.

The status information is often useful in determining what is wrong.

Illegal Instructions.

User programs under OS3 operate in "program state"; in this state, there are a number of machine instructions which are not allowed. All of these instructions cause a trap to the OS3 executive routine. Some of them are meaningful to OS3 and cause certain actions to be carried out for the user program, such as input/output, etc. The rest are simply illegal and cause an error message to be printed.

Programmers who use systems such as FORTRAN and OSCAR should not ordinarily encounter illegal instructions, except for input/output actions that are illegal. COMPASS programmers are likely to encounter almost anything; but then, they are usually better qualified to figure out what is wrong. We shall discuss the illegal instructions below, with particular emphasis on illegal input/output actions.

Instructions that could stop the computer are recognized by OS3, and one of the following messages is printed:

HLT yyyyy AT xxxxxx
SLS AT xxxxxx
UCS AT xxxxxx

In each case, xxxxxx is the location of the offending instruction. yyyyy is the address portion of the HLT instruction. Illegal input/output instructions cause the message:

ILLEGAL INSTRUCTION (yyyyyyyy) at xxxxxx. The instruction is printed in octal (yyyyyyyy) and its location is xxxxxx. If the instruction is of the form (740zzzzz), it is a READ instruction, and zzzzz is the logical unit number (in octal) on which the READ was attempted. If zzzzz is greater than 00143_8, then the logical unit number was greater than 99_{10} and this is illegal. If zzzzz is in the range 00000 to 00143, then it was illegal to read from the unit in question; either it

was a file at end of data or it was an output device.

An illegal instruction of the form (760zzzzz) is a WRITE instruction. Again, zzzzz is the logical unit number and must be in the range 00000 to 00143. If zzzzz is in the proper range, then the instruction was illegal, either because the unit was an input device, or because the length of record to be written was too long for an output device or because the unit was file protected. See Q in the STATUS for the number of words that were to be written.

An illegal instruction of the form (720zzzzz) is a CONTROL operation on a device. zzzzz is the logical unit number. If it is in the proper range, the contents of Q specifies the function which was to be carried out.

Q	Function
0	Check status
1	Clear status
2	Write file mark
3	Rewind
4	Space forward past file mark
5	Space backward past file mark
6	Clear reverse read
7	Set reverse read
10	Release file
11	Protect file
12	Space forward one record
13	Backspace one record