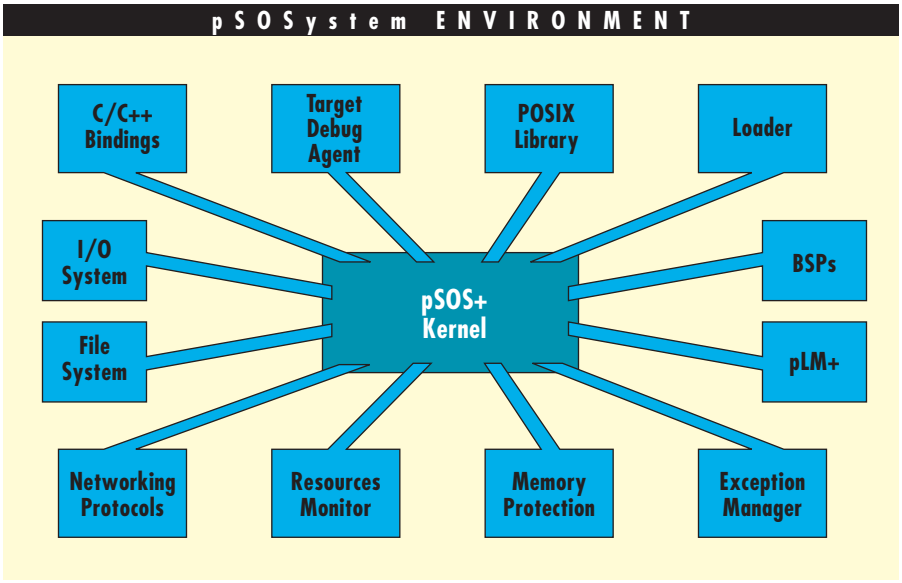
pSOSystem 3



The modular pSOSystem™ 3 real-time operating system (RTOS) is designed specifically to speed development of embedded applications ranging from simple, standalone devices to complex, networked systems. pSOSystem 3 is easy to use, and provides developers a reliable, proven multitasking environment that ensures optimum performance of applications built on either custom or commercial hardware. Since each pSOSystem component is completely self-contained, developers can scale OS features and memory to meet the precise requirements of a given application.

New features improve OS reliability

The pSOSystem 3 application program interface (API) is fully compatible with previous pSOSystem releases, and provides new operating system objects and OS services that not only preserve existing investments in pSOSystem, but also create a highly reliable RTOS. New features include:

- *Memory management and protection:* Protect the OS, applications, and individual tasks from misbehaving tasks.
- *Resource monitor:* Makes it possible for an application to detect failures before they occur.

Features

- Support for a wide variety of CPUs and drivers
- Fast, deterministic kernel
- Memory management and protection
- Resource monitoring
- Exception management
- Preemptive multitasking environment
- Reliable: proven in over 40 million customer devices worldwide
- Popular BSPs supplied in source code format
- File system support including ISO 9660 (CD-ROM), MS-DOS compatible file systems, NFS, and a high-performance embedded file system
- State-of-the-art pRISM+ 3 integrated development environment
- Tools for developing, analyzing, and testing embedded applications
- Simplifies navigation through legacy code and enables code reuse

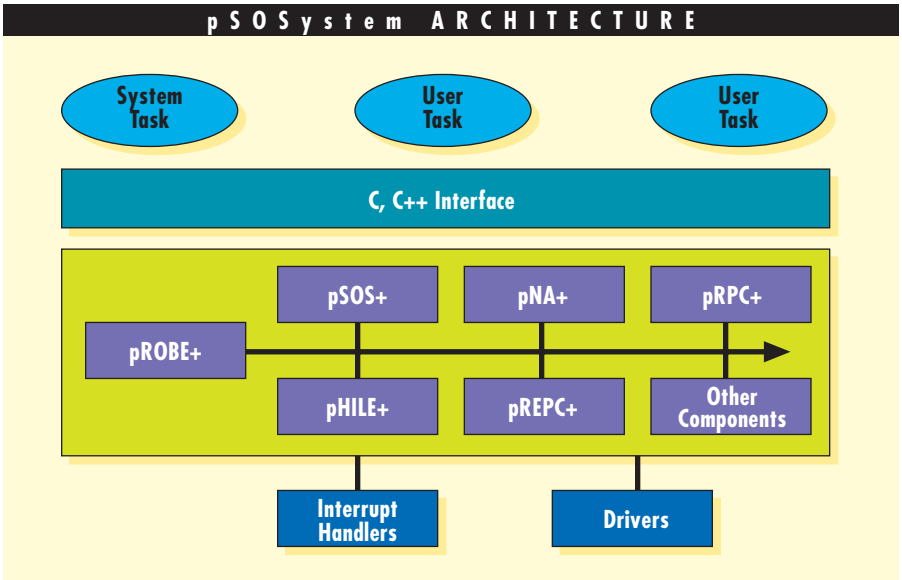
Supported targets

- Motorola/IBM PowerPC architecture family
  - Motorola/IBM PPC6xx, 7xx, 74xx
  - Motorola MPC8xx, 82xx
  - IBM PPC403
- MIPS architecture family
  - MIPS 16, 32, R500

Supported hosts

- Windows NT 4/2000
- UNIX-Solaris

WindRiver®



- *Exception manager:* Prevents task-generated exceptions from causing a system-wide reset.

pRISM+ 3 IDE

The pRISM+® 3 integrated development environment (IDE) includes all the tools and technology needed to bring up hardware, develop firmware, develop and debug applications, profile and optimize system performance, manage team-based development, and enable mixed-platform development for pSOSystem 3-based applications.

pSOSystem 3 architecture

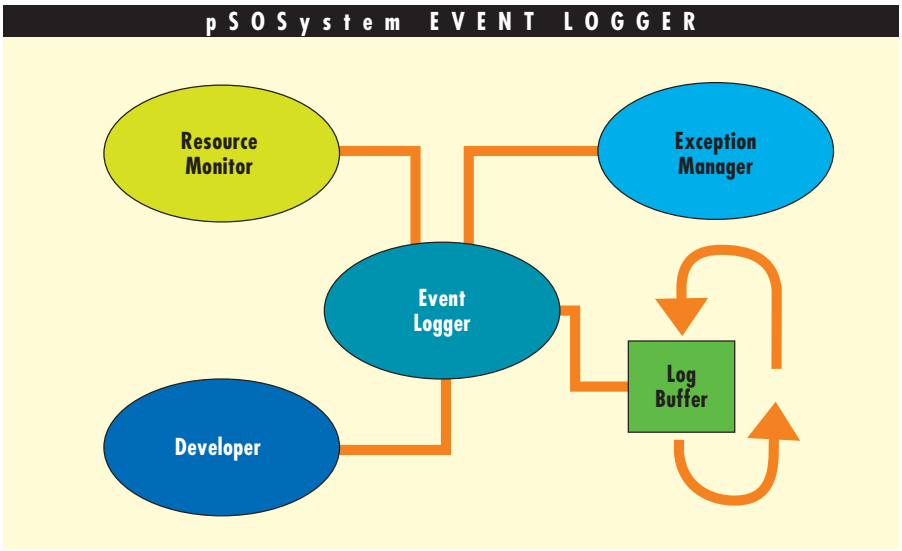
The heart of pSOSystem 3 is the pSOS+™ 3 multitasking kernel. Modular software components and libraries are standard building blocks that can be "plugged in" to the kernel as needed, and remain unchanged from application to application. A board support package (BSP) containing chip initialization and device drivers isolate pSOSystem components and applications from the underlying hardware. This protects investments in application development against processor and peripheral hardware obsolescence. New or upgraded hardware requires only simple modifications to device drivers in the BSP.

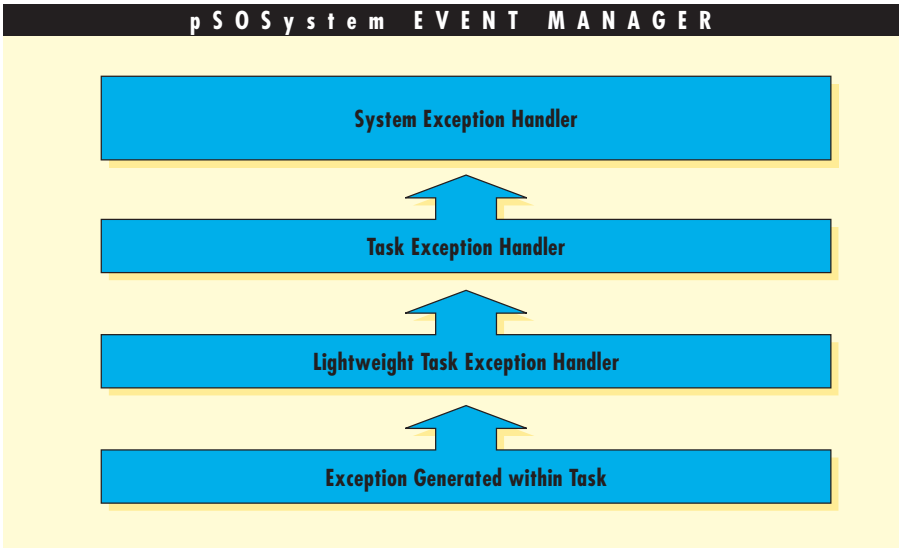
pSOS+ 3 real-time kernel technology

The pSOS+ 3 kernel is small, fast, reliable, and deterministic. It acts as supervisory software by performing services on demand: scheduling, managing, and allocating resources and coordinating multiple asynchronous activities. It employs a priority-based task scheduler that supports time-based, preemptive scheduling specified on a per-task basis. pSOS+ services include task management, semaphores, events, timers, fixed and variable length queues, and asynchronous signals.

Key pSOS+3 features include:

- *Memory management and protection:*
  - Management: pSOS+ supports multiple regions of memory in the OS, including noncontiguous regions. Individual tasks can "own" their own complete memory region if required. Both fixed and variable-sized memory regions can be dynamically allocated. pSOS+ also supports the memory management units (MMUs) of some CPUs, a useful function that protects a task's data and code spaces from accidentally being corrupted by other tasks.
  - Protection: pSOS+3 enhancements include features that protect the operating system and applications from misbehaving tasks. Memory overrun conditions are trapped by the OS, thus maintaining memory integrity during system operation. Code sections can be protected as execute-only so that there is no accidental corruption of an application. Stack overflow and underflow detection guards pages of virtual memory. Segments of memory, such as critical arrays, can be similarly guarded. Address verification is conducted before the OS attempts to access passed addresses. After an application is developed, a CPU's MMU can be switched off to optimize the product's performance.
- *Resource monitor:* Tracks critical system resources and makes it possible for an application to detect failures before they occur. For example, if a task is using too much resource time, the application can reduce the task's load on the system. If a task's stack usage reaches a high watermark, the application can avoid dealing with a memory protection issue that will most likely cause that task to be deleted or restarted.
- *Exception manager:* Exceptions may occur in the hardware, in a failing operating system call, or in a failed library call. The exception manager handles task-based exceptions at task level and system-based exceptions at system level. This prevents a task-generated exception from resulting in a system-level response such as a reset. Similarly, a task need not check system return codes, because system-based exceptions are handled in the OS, resulting in smaller, faster applications.
- *User/supervisor modes:* If a CPU supports multiple running modes, developers can run tasks in either user or supervisor mode, which gives developers the control





they need to optimize processor performance. User mode protects critical registers from being accidentally overwritten, while supervisor mode typically allows unrestricted access to critical registers or functions.

- *Event logger:* Enhances system manageability and maintenance by logging system and application-specific time-stamped event information, and providing a uniform event notification mechanism.
- *Device independence:* Facilitates operation on custom hardware. All hardware-specific operating system elements, such as processor initialization and device drivers, exist outside the pSOS+ kernel. This results in interrupt and exception handling that is fast and highly deterministic, while remaining under developer control and maintaining a stable kernel image that vastly improves an application's reliability.
- *Enhanced POSIX support:* POSIX extensions for lightweight *pthreads* and POSIX-style signals (e.g., timers, semaphores, and messages) enable faster reuse of software and easier porting of UNIX- and Linux-based applications to the pSOSystem environment.
- *Multiple "object wait" features:* Tasks can simultaneously wait on timers, events, queues, and semaphores. This enables event-driven operation, which eliminates unnecessary polling. As a result, a microprocessor can run applications more efficiently.
- *Mutex semaphores and conditional variables:* These provide an optimum set of system services for implementing mutual exclusion among tasks and device drivers. They include both the priority inheritance protocol and the more elegant priority ceiling, or protection, protocol.
- *QBIND (quick bindings) mode*: This prevents the trap from making system calls, resulting in as much as a 25 percent improvement in the speed of selected system calls.
- *Interrupt handling:* pSOS+ handles interrupts externally. As a result, designers have full flexibility in the way they handle interrupts and control key system elements, such as resetting hardware based on critical exceptions.

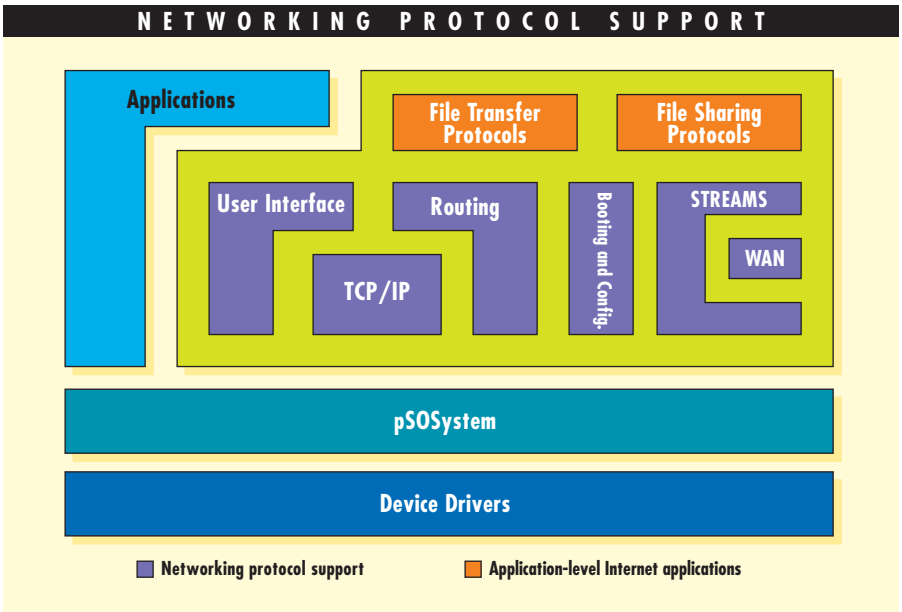
pSOSystem components

pROBE+ target/debug agent

A sophisticated kernel-aware component that functions both as a cross-development target agent and as a standalone target debugger. As the latter, designers can use pROBE+™ to bring up new hardware and to develop applications. pROBE+ also performs system profiling functions that aid in analyzing and optimizing application performance. Acting as a target agent for high-level language debuggers, pROBE+ supports system- and task-level debugging through serial and network connections, allowing dynamic switching between modes. pROBE+ also provides the key communications protocol to the host

development system. The agent runs externally, away from the kernel, so pROBE+ can debug start-up code, diagnostics code, and even device initialization code, all before the operating system has initialized. Developers can use pROBE+ to bring up custom or modified hardware using only simple serial drivers.

A number of source-level debuggers can connect to the target using BDM, JTAG, or pROBE+. pRISM+ for pSOSystem supports target connections through Wind River's SingleStep debuggers. pRISM+-supported debuggers have full pSOS+ object awareness, allowing detailed viewing of pSOS+ objects such as queues, semaphores, or memory regions.



Query library

Gives application-level code access to pSOS+ operating system internals through an API, and enables application developers to get advanced, detailed information on pSOS+ objects, such as which tasks are running, what stack was allocated to a task, and its current execution state. This feature is useful when creating applications that need diagnostic functions.

pMONT+

pMONT+™ gathers information at the operating system level about which tasks and interrupt service routines (ISRs) have run, how much CPU time each task has used, and what interrupts have fired for the object browser and ESp™ real-time analysis tools.

C/C++ support

pREPC+™ is a fully reentrant ANSI-compliant C library that contains more than 100 commonly used C functions. pSOSystem services integrate seamlessly with C++-based applications through the C++ class library. The fully reentrant *iostreams* library for pSOSystem contains additional C++ support. The *iostreams* library tightly integrates with pSOS+ and the C++ class library and supports all file and standard stream I/O.

pHILE+ file system manager

This component supports four separate file systems formats: a UNIX file system optimized for real-time operation; an MS-DOS-compatible file system that supports both floppies and hard disks; an ISO 9660 CD-ROM file system; and both client and server for NFS pHILE+,™ which permits preallocation of contiguous file segments and the growth of segments beyond their original boundaries. pSOSystem 3 enhancements include support for long file names in MS-DOS, NFS, and ISO 9660 (CD-ROM), as well as removable media, and improved scalability that allows designers to use only the file system support required by an application, thereby reducing the OS footprint and making the final product more cost effective.

pLM+ library manager

Provides shared library features by dynamically loading libraries into the target at run time. pLM+™ checks that version numbers are correct for the application, and performs the load and hook-ups to enable shared libraries to integrate seamlessly with an application. This offers a time- and cost-efficient way to provide upgraded features to embedded devices.

I/O supervisor

Provides a simple, flexible mechanism for interfacing peripheral devices to application code. With a standard interface for device drivers, developers can write portable application code that is not tied to a specific hardware device. Although device I/O can be performed without pSOS+ kernel intervention, device drivers integrated into pSOSystem's device-independent I/O supervisor maximize application portability and ensure protection from I/O device obsolescence.

Since device drivers are hardware-dependent, the pSOS+ kernel defines a standard set of six I/O services that a device driver may support,

and does not impose restrictions or make assumptions about the services provided by the driver.

SCSI support

Provides direct access support for initiator mode SCSI I and SCSI II. pSOSystem's device-independent SCSI library communicates with low-level device drivers to simplify system integration. Low-level device drivers for a number of popular SCSI controllers are also included in source form.

Interrupt handling

pSOSystem does not force interrupts through the kernel to an "interrupt handler task." Instead, the pSOS+ kernel handles interrupts externally. This makes the pSOS+ kernel extremely responsive, and gives the developer full flexibility to handle interrupts and to control key system elements such as resetting hardware based on critical exceptions.

Board-support packages

BSPs are available for a number of commercially available target platforms. Each BSP provides a software template that includes skeleton device-driver, low-level system-function code that each particular hardware device requires. The device-driver code itself is specific to individual peripheral devices, but is not specific to the board on which the devices reside. All BSPs contain initialization code, an interrupt-driven console driver, and a real-time clock driver. Boards with Ethernet and/or SCSI hardware include the appropriate drivers and boot ROM images of the BSP.

Integrated networking products

Networking is a key element in many embedded applications. The pSOSystem networking suite offers a broad set of target-based networking capabilities and features to embed TCP/IP, SNMP, Embedded Internet® utilities and multiprotocol routing.

Binary and source code packages designed and tuned to the extended capabilities and performance of the pSOS+ kernel are ready to run on a pSOSystem platform. Designed for a real-time environment, the Internet protocol (IP) stack provides concurrent high-speed connections with minimal resource requirements. Complete implementations of Internet protocols are provided, including support for subnet routing and masks, TCP, UDP, IP, ICMP, IGMP, and ARP. The stacks include support for message fragmentation, IP and UDP multicast, direct access to the IP layer, TCP/IP MIB-II information support, CIDR, out-of-order, routing, and gateways.

- *TCP/IP stacks:* pSOSystem includes the pNA+™ binary TCP/IP stack, plus a source code TCP/IP stack that is integrated with pSOSystem and its network interfaces and adds complete configurability and optional IPv6 and IPSec support. The source code stack is also easy to integrate into mixed hardware/software implementations of network traffic systems such as data communications switches. pSOSystem's TCP/IP protocol stacks support the optimization of network buffer configurations for reliable buffer management in memory-constrained applications. For example, pNA+ includes zero-copy technology that minimizes data copying in a multitasking environment.
- *Binary networking products:* Rich LAN and WAN protocol and network management capabilities, all of which are based on Internet RFC standards, are available for embedded networked devices.
- *Physical layer and drivers:* pSOSystem offers innovative features for managing physical layer device drivers. This allows higher-layer protocols such as console, PPP, (C)SLIP, and HDLC to be chip-independent, permitting developers to easily integrate their designs onto new hardware platforms and share the same chip or multiple chips. pSOSystem's network interface infrastructure allows sharing of network interface data between multiple protocols, even between compact and STREAMS-based implementations Modem scripting, secured PPP, and network address translation (NAT) are available for remote access or dial-up environments.
- *Distributed applications:* The pRPC+™ software component supports ONC/Sun remote procedure calls (RPC) and external data representation (XDR) specifications, and allows developers to construct distributed applications using the C procedure call paradigm. pSOS+ tasks and UNIX processes can invoke similar application calls. pRPC+ is used in conjunction with pHILE+, pSOSystem's file manager, to support NFS client and server protocols.
- *Embedded Internet features:* pSOSystem provides all of the "plug and play" features required to efficiently complete Embedded Internet designs, including file transfer and sharing, plus embedded user interfaces. Where applicable, these features are provided with both an API and a command-line interface. They include the following standard functions: FTP client and server, NFS server (the client is implemented in the pHILE+ file manager), TFTP client and server, telnet client and server, RARP, BOOTP client and server, DHCP, and DNS resolver.

  The pSOSystem shell allows a jump-start implementation of a classic command line-driven serial interface, and is integrated with telnet. The shell's configurability enables the fine-tuning of implemented commands, and is available with a variety of file system and network-

related commands to configure an embedded shell. An HTTP server that can run without a file system is available as an option. For time synchronization of a networked system or within multiprocessor applications, NTp™ is available.

- *STREAMS protocol framework and WAN:* In addition to Internet protocol functions, STREAMS is provided on the protocol framework level. This is a standard interface to the OS and to the hardware layer, with a framework for integrating third-party, legacy, and proprietary protocols.
- *Device and network management:* Simple network management protocol (SNMP) is an optional networking application that allows the development of high-performance, fully reentrant SNMP client applications for device management. A library of SNMP function calls and a MIB compiler permit the development of agent, proxy, master/subagent, and HTML applications.

Wind River
Worldwide Headquarters
500 Wind River Way
Alameda, CA 94501 USA
1-800-545-WIND toll-free
1-510-748-4100 phone
1-510-749-2010 fax
inquiries@windriver.com
NASDAQ: WIND

For additional contact information, please see our Web site at www.windriver.com.