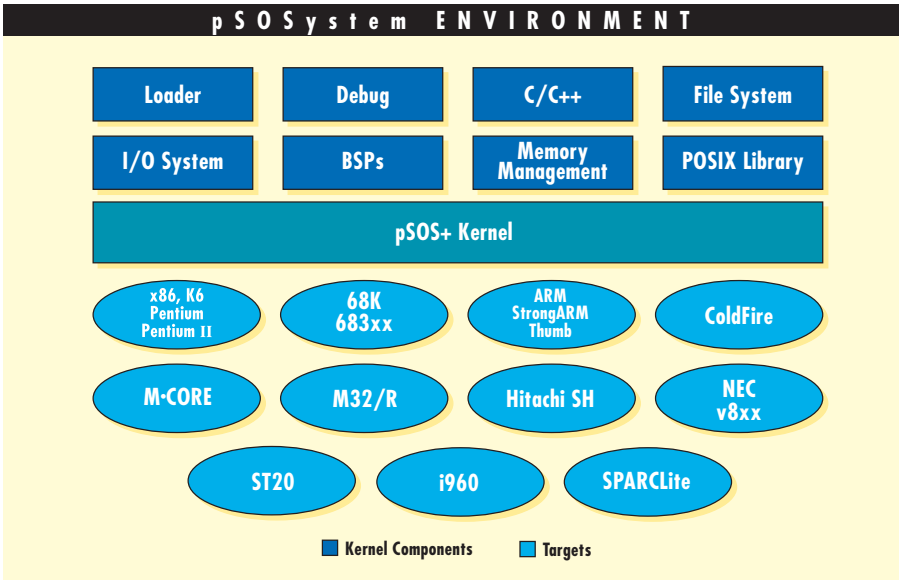
pSOSystem 2.5



pSOSystem™ is a modular, high-performance, real-time operating system intended specifically for embedded microprocessors. It offers a complete multitasking environment, providing performance, reliability, and ease-of-use on both custom and commercial hardware. Each component of pSOSystem is completely self-contained, allowing developers to scale OS features and memory to meet the precise requirements of a given application. Developers can easily scale pSOSystem-based designs from simple stand-alone devices to complex, networked, multiprocessor systems. pSOSystem's tool chain, the pRISM+® integrated development environment (IDE), includes a graphical toolbar for navigating between C/C++ compilers, source-level debuggers, source code browsers and editors, configuration and version control tools, and target analysis tools. pRISM+ is built on the CORBA (Common Object Request Broker Architecture) open object bus. CORBA supports third parties with a published application program interface (API), allowing easy integration with many different types of tools.

pSOSystem architecture

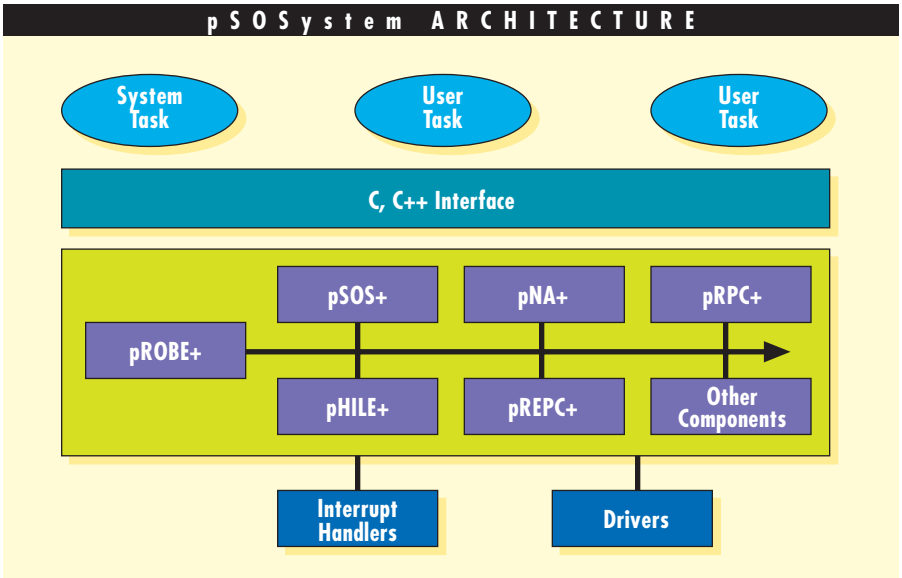
pSOSystem uses a modular architecture that contains the pSOS+™ real-time multitasking kernel along with a collection of

Features

- Support for a wide variety of CPUs and drivers
- Fast, deterministic kernel
- Preemptive tasking environment
- Reliable – proven in over 40 million customer devices worldwide
- Popular board support packages (BSPs) supplied in source-code format
- Multiprocessor support
- File system support including ISO9660 (CD-ROM), MS-DOS compatible file systems, NFS, and a high-performance embedded file system
- State-of-the-art integrated development environment
- Supports tools for developing, analyzing, and testing embedded applications
- Supports team development with configuration management and version control capabilities
- Simplifies navigation through legacy code and enables code reuse
- Available on Windows 95/98/NT and UNIX workstations

WindRiver®

companion software components and libraries. Software components are standard building blocks delivered as modules that remain unchanged from application to application. This component black-box technique eliminates maintenance by the user and assures high reliability. pSOSystem component technology is combined with a board support package (BSP) that contains chip initialization and device drivers that isolate pSOSystem components and applications from the underlying hardware. This protects application development investment against processor and peripheral hardware obsolescence. New or upgraded hardware requires only simple modifications to device drivers in the BSP.



pSOS+ and pSOS+m real-time kernel technology

Small, fast, reliable and deterministic, pSOS+ is the real-time, multi-tasking kernel that forms the base of the pSOSystem operating system, and 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 supporting time-based and preemptive scheduling specified on a per-task basis. pSOS+ services include task management, semaphores, events, timers, fixed and variable length queues, and asynchronous signals. pSOS+ supports multiple regions of memory, which can be non-contiguous. Individual tasks can "own" their own complete memory region if required. Both fixed and variable-sized memory regions can be dynamically allocated. pSOSystem supports the memory management units of some CPUs, a useful function that protects a task's data and code spaces from accidentally being corrupted by other tasks.

If a given CPU uses different running modes, pSOS+ allows developers to run tasks in either user mode or supervisor mode. Supervisor mode typically allows unrestricted access to critical registers or functions, while user mode protects critical registers from being accidentally overwritten. pSOS+ supports both modes of operation, providing developers maximum control for optimal processor performance.

The pSOS+ kernel is a distributed, multiprocessor implementation of pSOS+. The pSOS+m™ kernel is designed so that tasks comprising an application can reside on several processor nodes and still exchange data, communicate, and synchronize exactly as if they were running on a single processor. pSOS+m shares the pSOS+ API, allowing developers to easily migrate single processor applications to a multiprocessor environment. pSOS+m is fully heterogeneous, allowing designers to mix different processor families in the same application. Furthermore, pSOS+m allows processor nodes to be connected by any kind of connection such as shared memory, serial or parallel links, Ethernet or proprietary implementations. Both pSOS+ and pSOS+m are dynamic in their operation, creating, modifying and/or deleting all operating system objects at run-time. Tasks can load dynamically into a running target system and be added or removed from the schedule.

To facilitate operation on custom hardware, pSOS+ and pSOS+m are completely device independent. All hardware-specific operating system elements, such as processor initialization and device drivers, exist outside the pSOS+ kernel, which decouples the kernel from the hardware operation. This results in interrupt and exception handling that is fast, highly deterministic, and under developer control, while maintaining a stable kernel image that vastly improves application reliability.

pROBE+ — the pSOSystem target/debug agent

pSOSystem's pROBE+™ is a sophisticated kernel-aware component that functions both as a cross-development target agent and as a stand-alone 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. System-level debugging is most useful in debugging new hardware or in developing device drivers where it is necessary to stop all activity at a breakpoint. Task-level debugging is most useful for analyzing individual tasks while the rest of the system continues to run.

pROBE+ 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 been initialized. Developers can use pROBE+ to bring up custom or modified hardware using only simple serial drivers.

pROBE+ can operate in a terminal mode, without a source-level debugger connected. pROBE+ provides an ASCII interface to the debug agent, enabling users to view OS objects, disassemble memory, and change memory – all from a simple-to-use command-driven terminal interface.

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 SearchLight™ and SingleStep™ debuggers, CAD-UL's XDB symbolic debugger, and ARM Ltd.'s debugger. pRISM+-supported debuggers have full pSOS+ object awareness, allowing detailed viewing of pSOS+ objects such as queues, semaphores, or memory regions.

pREPC+ — C/C++ language support for pSOSystem

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 I/O and standard stream I/O, offering the embedded system designer *iostreams* functions such as "cout" and "cin."

pHILE+ — file system management

pSOSystem's pHILE+™ component is a file system manager that provides support for four separate file systems formats:

- A pHILE+ hierarchical UNIX-style file system format, optimized for real-time operation
- An MS-DOS compatible file system that supports both floppies and hard disks
- ISO 9660 CD-ROM file system
- Client and server for NFS pHILE+, which permits preallocation of contiguous file segments and the growth of segments beyond their original boundaries

I/O supervisor

pSOSystem's 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. While it is permissible to perform device I/O without pSOS+ kernel intervention, device drivers that are 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 driver implementation determines the exact services offered by a device driver. 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.

Device-independent serial interface

The pSOSystem device-independent serial interface (DISI) specification provides a standard API for serial communication over both synchronous and asynchronous connections. The synchronous API provides support for high-level data link control (HDLC) protocols such as X.25 and ISDN. All serial drivers supplied with pSOSystem software support the DISI specification.

SCSI support

Direct access support for initiator mode SCSI I and SCSI II comes with the standard pSOSystem product. 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

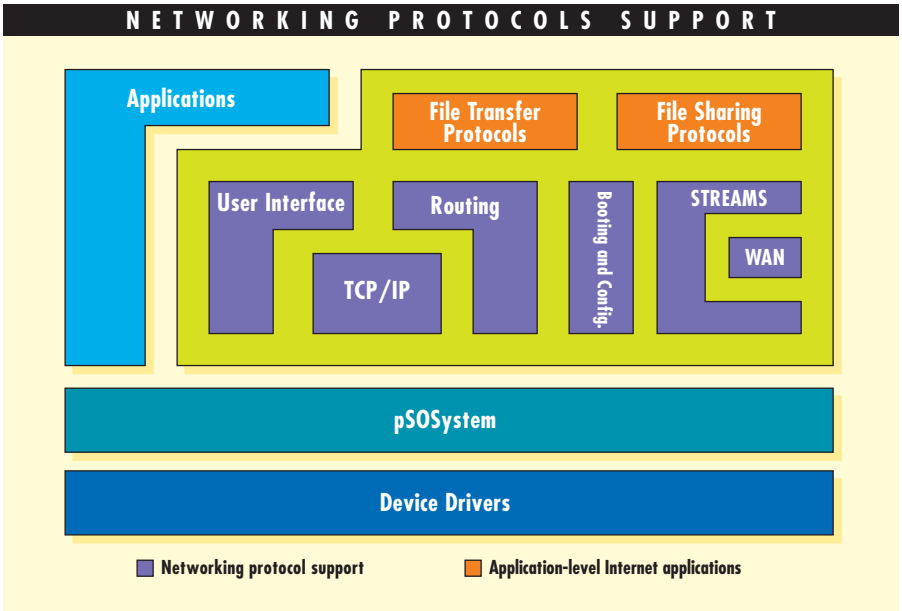
pSOS+ handles interrupts externally; that is, outside the kernel. The operating system does not force interrupts through the kernel to an "interrupt handler task." pSOS+ allows the developer to control how the OS handles interrupts and exceptions, giving full control of machine exception handling by jumping directly to the interrupt service routine (ISR) from the vector page without first going through the kernel. This makes the pSOS+ kernel extremely responsive. pSOS+ allows, but does not require, developers to place a wrapper around an ISR alerting the kernel that an interrupt is occurring. A wrapper may be useful if, for example, an ISR uses a pSOS+ object like a queue or a semaphore, and the object changes, at which time pSOS+ will reschedule tasks that were pending on the object. pSOS+ offers the designer full flexibility to handle interrupts and to control key system elements, such as resetting hardware based on critical exceptions.

Board support packages (BSPs)

To facilitate application development on customer hardware, pSOSystem provides BSPs that offer a mechanism for supporting standard and custom embedded hardware designs. Each BSP provides a software template that includes skeleton device driver code and code for low-level system functions 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. A number of board support packages are available in source code.

Embedded graphics support

The OpTIC™ graphics source library is an ideal choice for simpler applications requiring character generation, drawing primitives, as well as a small memory footprint. For more sophisticated windowing applications, Wind River works closely with a number of third-party graphics suppliers.



Integrated networking products

Networking products

Because networking is a key element in many of today's embedded applications, pSOSystem includes the pNA+™ binary TCP/IP stack. 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 sockets technology, which minimizes data copying in a multitasking environment. As network packets pass through the various TCP, UDP or IP layers, various data manipulations are performed (passing to higher/lower layers, fragmentation, or reassembly).

pSOSystem binary networking products

pSOSystem provides rich LAN and WAN protocol and network management capabilities for embedded networked devices, all of which are based on Internet RFC standards. Because of its binary design and preintegrated form with various popular platforms, pSOSystem offers easy start and extended support capabilities to embedded developers.

Physical layer and drivers

pSOSystem offers innovative features for managing physical layer device drivers, including a unique serial-line driver infrastructure for mixed asynchronous and synchronous serial protocols and applications (DISI). This allows higher layer protocols to be chip independent, permitting developers to easily integrate their designs onto new hardware platforms and share the same chip or multiple chips; for example, console, PPP, (C)SLIP and HDLC. pSOSystem's network interface infrastructure allows sharing of network interfaces data between multiple protocols, even between compact and STREAMS-based

pSOSystem COMPONENTS

pSOS+ real-time multitasking kernel
A highly reliable, field-proven, preemptive multitasking kernel that provides a responsive, efficient mechanism for coordinating the activities of a real-time system.

pSOS+m multiprocessor multitasking kernel
Extends the pSOS+ feature set to operate seamlessly across multiple, tightly coupled or distributed processors. pSOS+m maintains the familiar pSOS+ API while adding communications and coordination functions. It can be used with all pSOSystem components and libraries.

pNA+ TCP/IP protocol stack
Includes a complete TCP/IP stack implementation based on the industry-standard BSD socket interface. Contains the following protocols: TCP, UDP, ARP, RARP, ICMP, and IGMP. pNA+ uses a standard socket interface that includes streams, datagram and raw sockets, expanding connectivity options developers can deploy with pSOSystem.

pRPC+™ remote procedure call library
Offers SUN-compatible (ONC) Remote Procedure Call (RPC) and XDR services that allow users to build distributed applications using the familiar RPC.

pHILE+ file system manager
Grants efficient access to mass storage devices, locally or across a network. pHILE+ includes support for CD-ROM devices, MS-DOS compatible file systems, and a high-speed proprietary file system.

pREPC+ ANSI C standard library
Provides reentrant versions of ANSI run-time functions such as printf() and malloc() that are fully integrated with the pSOS+ kernel and I/O services. Programming with familiar functions significantly speeds development time. Using reentrant functions allows true multitasking.

pROBE+
Enables advanced target- and host-based debugging using the pROBE+ target resident agent. pROBE+ is not dependent on pSOS+, allowing developers to obtain debug support during the BSP development process.

pMONT+™
Gathers information for the Object Browser and ESp® (embedded system profiler) real-time analysis tools. The pMONT+ target agent collects information on 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. It passes all this information to ESp and the Object Browser on the host. ESp and the Object Browser are part of pRISM+ for pSOSystem's suite of real-time analysis and optimization tools.

pSOS+ query library
Affords application-level code access to pSOS+ operating system internals through an API. The pSOS+ query library provides application developers advanced, detailed information about 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 requiring diagnostic functions.

pLM+™
Provides a shared library feature to pSOSystem. The pSOSystem library manager (pLM+) dynamically loads 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 the dynamically loaded shared library to integrate seamlessly with an application. This offers a time- and cost-efficient way to provide upgraded functions to embedded devices.

Other pSOSystem components
A complete description of all the pSOSystem components appears in the *pSOSystem Programmers Reference Manual.*

implementations. For remote access or dial-up environments, modem scripting, secured PPP and network address translation (NAT) are available.

Distributed applications

The pRPC+ software component supports the 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 applications

pSOSystem provides all of the plug-and-play applications required to efficiently complete designs, including file transfer and sharing, plus embedded user interfaces. Where applicable, they are provided with both an API and a command-line interface. Embedded Internet applications include the following standard functions: FTP client and server, NFS server (the client is implemented in 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. As an option, an HTTP server is available and can run without a file system. 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

SNMP (simple network management protocol) is an optional networking application that allows the development of high-performance, fully reentrant SNMP client applications for device management. The library of SNMP function calls and a MIB compiler permit the development of agent, proxy, master/subagent, and HTML applications.

TARGET PROCESSORS

- **Motorola 68K architecture family including CPU32/CPU32+**
- **Motorola ColdFire architecture family**
- **ARM architecture family including Thumb and StrongARM**
- **x86, K6, Pentium, Pentium II, Pentium III architecture families**
- **Intel i960 architecture family**
- **Super Hitachi (SH) architecture family**
- **Motorola M•CORE architecture family**
- **Mitsubishi M32/R architecture family**
- **NEC architecture family**

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.