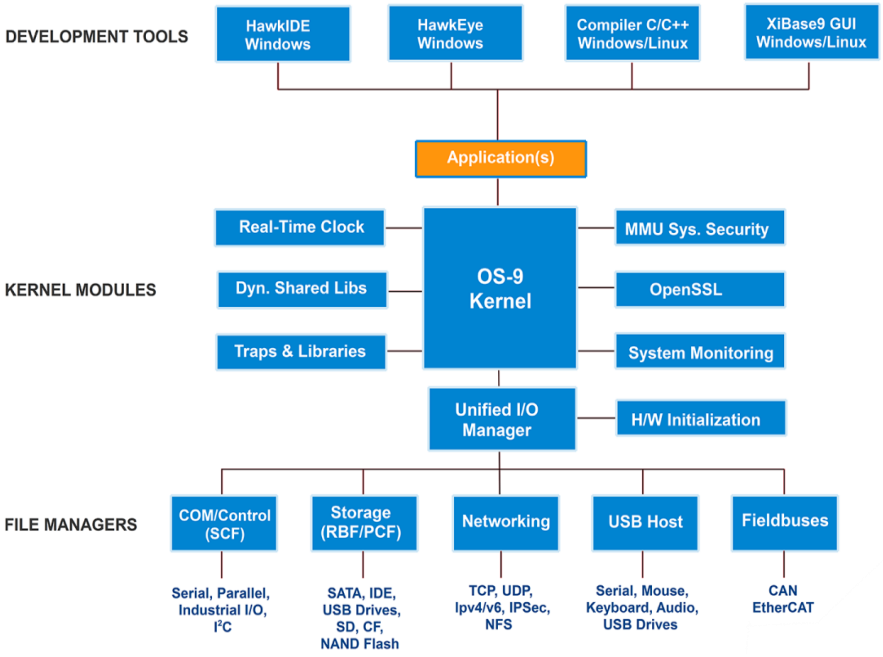
MICROWARE® OS-9 RTOS

Deterministic - Efficient - Scalable - Fast Booting

Embedded systems span a myriad of applications, ranging from simple microcontrollers to sophisticated medical imaging systems to complex industrial applications. At the heart of these diverse applications is an operating system (OS) - a software foundation that delivers a common set of services helping software developers deliver their product to market more quickly.

Enter Microware OS-9, the high-performance, high-availability real-time operating system platform from MicroSys. The Microware OS-9 RTOS has been deployed and proven in thousands of products worldwide and represented hundreds of embedded applications, including industrial automation and control and automotive and medical instrumentation.



Why Microware OS-9?

Microware OS-9 compact, high-performance multi-user, multi-tasking real-time kernel is a proven foundation for time-to-revenue success. OS-9 is a full-featured operating system framework, including the OS kernel, kernel services, and industry-standard APIs, middleware, and a complete IDE-based development framework.

Reduce Risk

- High reliability - the OS-9 secure process model, real-time operating system (RTOS) provides inherent memory management, resource authorization, and module CRCs, dramatically improving system reliability.
- High performance - OS-9 makes effective use of the CPU by providing integrated I/O and compiler technology tuned for the specific processor instruction sets.
- High availability - OS-9 has the ability to add, remove, and replace individual components in the system while on-line and in-use. This results in a high degree of system availability, even during maintenance. Proven over 30 years in mission critical devices around the world.
- Hard Real-Time Performance - Unlike Windows and Linux-based systems, Microware OS-9 was conceived from the ground up to meet the high-performance and reliability requirements of time-critical embedded applications.

Fast Booting

- Small footprint by modular architecture
- It fits services to application requirement
- No file systems needed

Offers an 'instant on' experience from reset to a running user interface.

Resource-Efficient by Design and Ease of Use

OS-9 delivers excellent performance in even the most constrained environments by its structured and modular architecture.

OS-9 is small and manageable. Applications are written as self-contained modules and not linked in as part of a single monolithic code base with the kernel. OS-9 includes a full interactive shell and many system utilities to directly monitor things like processes, memory usage, devices, system events, interrupts and more, all with built-in help.

Scalable Modular Architecture

The OS-9modular architecture enables dynamic configuration changes and enhancements in order to meet changing system requirements without rebooting. OS-9 applications are written as self-contained modules and are not linked in as a single monolithic code base with the kernel. Virtually any OS-9 component may be added, removed, or updated either at system startup or while the system is running. This means features and new functionality can be added easily, in real-time, and even after deployment to the field.

Reliability, Safety and Security

OS-9 was designed with reliability, safety and security in mind. Unlike monolithic architectures, the OS-9 advanced modular architecture offers an enhanced level of security, making it a preferred foundation for today's networked environment.

Extensive Services and Middleware

The OS-9 extensive I/O architecture supports a wide range of devices and networking options. The broad range of services, file managers and device drivers available from a single supplier speeds integration and application development, leaving customers with more time to innovate and differentiate their product.

Graphical User Interface XiBase9

Creating more compelling graphical user interfaces plays a crucial role in the success of a wide range of products and applications. The newest XiBase9 design tool set under OS-9 for graphic objects, allows the creation of complex and optically appealing graphic interfaces without programming effort involved. This new type of approach to generate graphic interfaces on embedded systems, offers the potential to reduce the development efforts up to 80%-90% compared to a traditional design process.

Native Development Framework

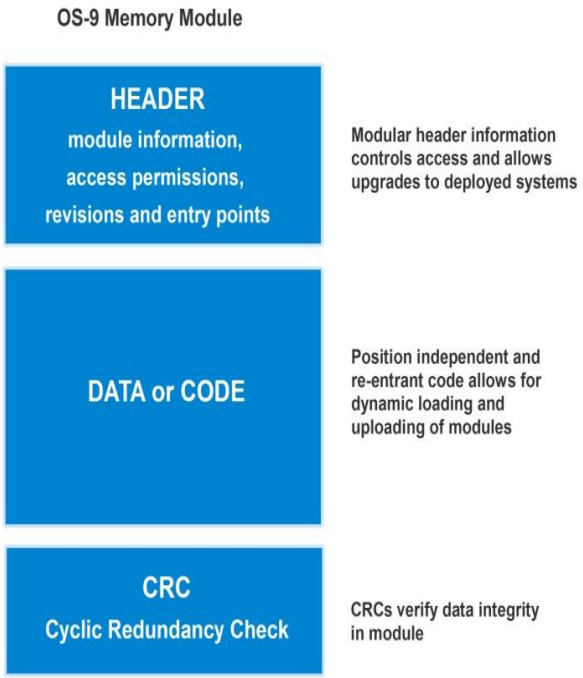
Designing embedded systems requires more than just an OS. The Microware Hawk development suite comprises of a comprehensive set of software development tools that span the embedded software workflow. The Hawk framework includes a highly optimizing C/C++ compiler, a fully customizable development environment, the CodeWright programmer's editor, graphical debugging tools and middleware libraries and solutions.

Shorten development time by an exception handling framework that is easily customized for automated error correction and recovery procedures for ultimate resilience.

Under the Hood:
The Microware OS-9 Architecture

OS-9 utilizes an advanced modular software structure that creates an optimum balance of speed and protection for the embedded systems and applications. OS-9 runs faster compared to microkernel operating systems and offers an increased level of security compared to monolithic architectures. All modules and components that make up an OS-9 system, including the kernel, file managers, drivers and application, are stored as logical OS-9 memory modules.

Each memory module is a self-contained program consisting of module header, module body, and a Cyclic Redundancy Check (CRC) value. These logical software components are easy to create and manage, while ensuring high availability of the OS-9 platform.



Real-Time, Multi-Tasking Kernel

OS-9 features a process-based, real-time kernel with POSIX thread support, complete with an extensive scheduler and inter-process communication facilities. Kernel features include:

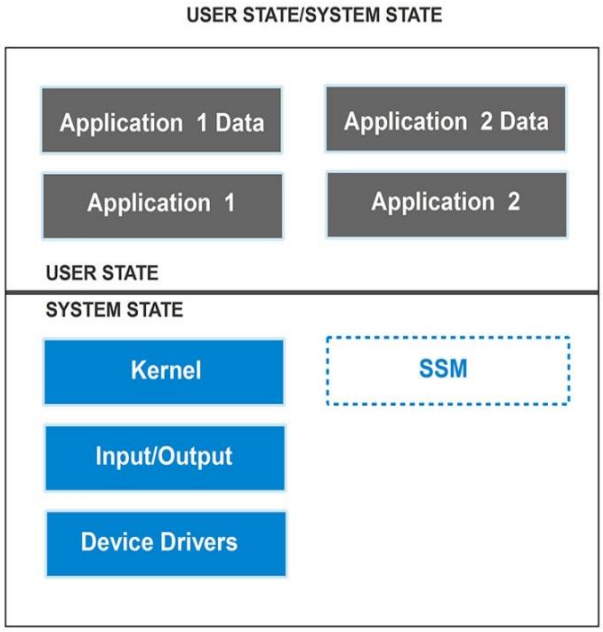
- Pre-emptive, priority-based, aging scheduler, assuring real-time responsiveness to interrupts and events
- Fault tolerance with exception handling and MMU support
- Process-based architecture with unlimited number of threads
- Inter-process communication services: Signals, Events, Semaphores, Pipes, Data Modules and Message Boxes.
- Tasks divided into priority-based classes
- Tasks can dynamically change priority
- Threads (POSIX) within processes
- Modular architecture based on dynamic linking
- Able to run out of Flash or ROM with no file system requirement

Protective Boundaries

Microware OS-9 supports three types of protective boundaries:

1. The user/system state boundary: User state code uses the processor's non-privileged instruction set, and the operating system's user-state system calls. These restrictions let the operating system contain the application within processor boundaries.

2. Process boundaries: Each process is granted access to resources by the operating system. OS-9 prevents processes from erroneously or maliciously disturbing other processes' resources without proper authorization. If the system is using a memory management unit, OS-9 uses it to enforce memory access protection rules. If there is no MMU, OS-9 still makes an effort to enforce protection rules.



3. User/group protection:
Each process, each module, and (for most file systems) each file has an associated owner. OS-9 uses ownership and permissions to administer access to resources.

The OS-9 real-time kernel allows multiple independent applications to execute simultaneously through task switching and inter-process communication facilities. All OS-9 programs run as processes containing at least one lightweight process (thread), but

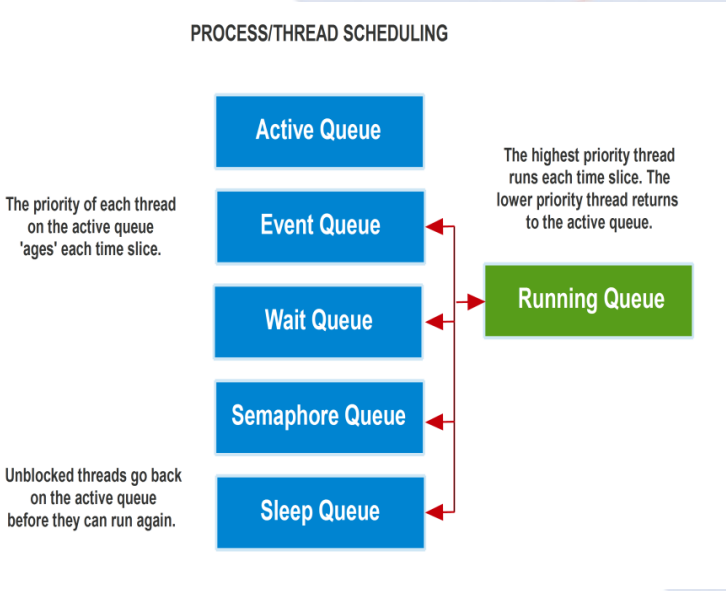
may also contain an effectively unlimited number of threads. Within a process, these lightweight processes share memory, I/O paths, and other resources in accordance with the POSIX threads specification and API.

Priority based Scheduling

OS-9 schedules threads using a fixed-priority preemptive scheduling algorithm with round-robin scheduling within each priority. The priority levels can be divided into a range that supports aging and a higher-priority range that uses strict priority scheduling. Each process can access any system resource by issuing the appropriate OS-9 service request. At every scheduling point OS-9 compares the priority of the thread at the head of the active queue to the priority of the current thread. It contextually switches to the thread on the active queue if its priority is higher than the current processes' priority. Aging artificially increases the effective priority of the threads in the active queue as time passes. At defined intervals, time slicing returns the current thread to the active queue behind other threads at the same priority.

Small Footprint

OS-9 and user applications can run completely out of ROM/Flash, leaving RAM dedicated for OS or program use. OS-9 modules are re-entrant. Ten copies of the same 500K program only uses 500K of program space and ten copies of data space (stack, process descriptors and other system structures associated with the process).



Power Management

- Critical design issue for any power sensitive product
- Kernel based sleep state - necessary for implementing power management
- Customizable module providing easy power policy specification for your device
- I/O systems designed to implement power-down or low-power operation when notified by the power manager

Processor Support

OS-9 supports most popular 32-bit processors, including the Power Architecture ARM/ARM Cortex, Intel Architecture, SuperH (SH-3, SH-4, SH-4a), the 68K family and MIPS3000, MIPS64.

Who is standing behind OS-9?

Since February 2013 Microware OS-9 is owned by a partnership of three companies, Freestation (Japan), MicroSys (Europe) and RTSI (USA).

Microware LP actively continues development on OS-9. Recent developments already provide support for ARM Cortex A8 and A9 cores with Freescale's latest i.MX5x and i.MX6 CPUs.

Microware OS-9 is available through:

- Freestation in Asia
- MicroSys in Europe
- RTSI in the United States