

# Prex Embedded Real-time OS



## Smart. Simple. Secure.

Prex delivers extremely portable and realtime microkernel for embedded systems.

Prex is an open source, real-time operating system for embedded systems. Prex is designed specifically for tiny, resource-constrained platforms. It is written in 100% ANSI C and its microkernel is highly portable. Prex supports POSIX interface and it provides file systems, an embedded libc, and CmdBox which includes tiny versions of Unix utilities.

## Key feature of Prex:

### Real-Time:

A preemptive, multithreaded kernel provides the basis for robust, power-efficient and responsive Prex.

### Memory Protection:

The system can be configured with MMU or without MMU. Its memory protection logic can prevent the entire system from any application faults.

### Portability:

The microkernel has a well defined hardware abstraction layer in it, and thus the developer can easily port the kernel for the different architecture or platform.

### License:

Prex is royalty-free software released under BSD license

### Scalability:

The basic OS functions like process, file system, application loading, and networking are provided by the user mode servers. So, the system designer can construct OS by choosing suitable servers for their target requisition.

### Security:

The microkernel provides the capability based security feature. The kernel or system servers can be protected by malicious applications.

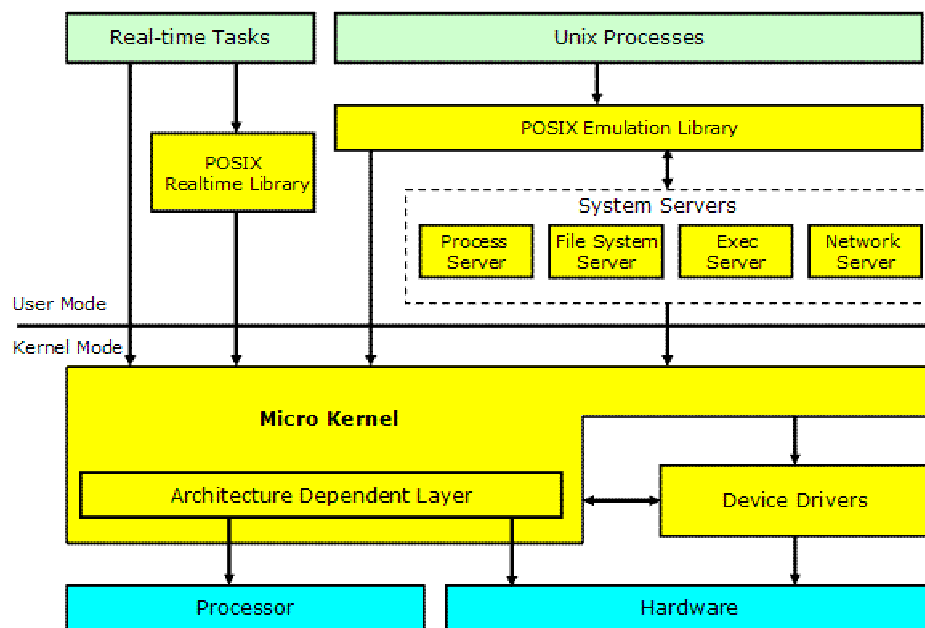
### POSIX Support:

The POSIX emulation libraries allow the system to use the existing Unix application properties.

### Small Footprint:

The size of a kernel module is only 20K bytes.

# Prex Embedded Real-time OS



## Technical specifications:

### Prex Microkernel

- Preemptive priority scheduling or FIFO scheduling
- 256 priority levels
- Fast context switching
- Memory protection
- Shared memory
- Synchronous IPC
- Fault trapping
- POSIX signal
- Counting Semaphore
- Condition Variables
- Mutex with priority inheritance
- Sleep timers
- One shot or periodic timers
- POSIX 1.e based Task capability

### Device Driver

- Separated from kernel module
- Nested interrupt service routines
- Prioritized interrupt service threads
- Deferred procedure call
- Power management framework
- Dynamic voltage scaling
- TTY

### Process Primitive

- Process ID
- fork
- exec

### File System

- Multithreaded VFS framework
- Buffer cache
- FIFO / pipe
- RAM filesystem
- FAT filesystem
- Dev filesystem

### User Applications

- Dynamic loading
- Embedded libc
- CmdBox
- Tiny shell
- Many sample applications

### Supported Target

- i386
- ARM