TargetOS™

REAL-TIME OPERATING SYSTEM

HIGHLIGHTS

- **Royalty Free**
- **Multi-Tasking**
- **Provides Semaphores, Queues, Events**
- **Extensive Timer Support**
- **Deterministic and Responsive**
- **Flexible Scheduler**
- **Small Footprint**
- **CodeWarrior Kernel Awareness**
- **Includes Standard C Library**
- **Command Line Monitor**

Blunk Microsystems provides system software, device drivers, and board support packages to the embedded systems market, both off-the-shelf products and custom work done under contract.

BLUNK
Microsystems

MULTI-TASKING

TargetOS is a real-time, priority-based, preemptive kernel. The scheduler shares the CPU among various tasks. Tasks are lightweight threads having a common address space and a single set of file descriptors. Tasks can be dynamically created and deleted and are always in one of five states: running, ready, blocked, suspended, or delayed. Only one task at a time is the running task.

When preempted by a higher priority task, the running task is placed at the front of the ready list for its priority level. Thus when the higher priority task blocks, the preempted task again becomes the running task. When the running task blocks to wait for an event or resource, it is placed at the end of its ready list.

Task priorities are assigned at creation and can be dynamically changed. Priority changes take effect immediately with tasks pending in a priority queue dynamically re-ordered in the queue. Priority inheritance is implemented to prevent priority inversion. This temporarily boosts the priority of any task owning a mutex that is being requested by a higher priority task.

SUPPORTS TASK COMMUNICATION

An application partitioned into multiple tasks requires support for task communication and synchronization. TargetOS provides event groups, message queues, counting semaphores, and a task sleep/wake mechanism. Both tasks and interrupt service routines can post events, messages, or semaphore tokens.

- Semaphores and queues perform either FIFO or priority scheduling of pending tasks.
- Queues are either limited to a maximum number of backlogged messages or allowed to grow indefinitely.
- Mutexes are binary semaphores with the addition of ownership and priority inheritance and are used to prevent priority inversion when tasks with multiple priorities use the same resource.
- Multiple tasks can block on a single event and a single event can make multiple tasks ready. Tasks wait for events using a 32-bit mask to select the events they respond to.
- Tasks sleep either indefinitely or for a number of kernel ticks. Tasks become ready after either timing out or a wake call from another task or interrupt service routine.

DETERMINISTIC AND RESPONSIVE

Interrupt latency and the execution time of most system calls do not increase as the number of tasks, semaphores, queues, etc., increase. The two exceptions, the get-identifier and the delete-object calls, are not usually time-critical. No loops are performed in the kernel with interrupts masked.

Timer-related processing that is variant, such as sorting the timer queue, is performed using background tasks. This allows the CPU to return more quickly to application tasks when a service call requires timer processing.

FLEXIBLE SCHEDULER

Preemption can be permanently disabled at compile-time, in which case each running task keeps the CPU until making a call that blocks. Preemption can be temporarily disabled at run-time by calling `taskLock()`. Preemption

Blunk Microsystems TargetOS™

www.blunkmicro.com

CONTACT INFORMATION

- Visit our web site:
  www.blunkmicro.com
- Customer Support:
  (408) 323-9833
- Technical Support:
  (408) 323-1758
- Fax:
  (408) 323-1757
- E-mail:
  sales@blunkmicro.com
- Address:
  6576 Leyland Park Drive
  San Jose, CA 95120
  USA

OPTIONAL COMPONENTS

TargetTCP
RFC compliant TCP/IP protocol stack providing the standard Berkeley Sockets API.

TargetFFS
POSIX compatible flash file system. Implements wear-leveling to prolong life of flash media. File system integrity is guaranteed across unexpected resets.

TargetLAPB
ISO/IEC 7776 protocol stack. Supports exchanging data on point-to-point networks. Provides automatic flow control and data reliability.

remains disabled until the running task either calls `taskUnlock()` or makes a call that blocks.

Round-robin scheduling among tasks of equal priority is supported. `OsTicksPerSlice` determines how long a round-robin task keeps the CPU if tasks of equal priority are ready. After the specified number of ticks elapse, the running task is placed at the end of its ready list and the first task in that list becomes the running task.

Extensive Timer Support
Timers can be used in a wide variety of calls. Semaphore tokens, messages, and events can be posted after a delay or posted repeatedly at intervals. User functions can be called after a delay or called repeatedly at intervals. Timers can be stopped, tested for expiration, and/or deleted at any time.

Easy to Use
Kernel objects (tasks, semaphores, timers, etc.) are created dynamically upon application request. There is no configuration file to edit and maintain. The only limit on the number of kernel objects is the amount of available memory. To help catch errors during development, a compile-time option promotes service call parameter errors to fatal errors that break into the command line monitor after printing an error message to `stderr`.

CodeWarrior™ Kernel Awareness
The kernel-awareness DLL lists every task and can change the debugger's source code display to show the execution point of any selected task. The state of tasks, semaphores, etc. can be displayed as well as a trace of recent kernel events: system calls, context switches, and interrupt service routines.

Interrupt Service Routine Manager
TargetOS allows interrupt service routines to be written entirely in C. Assembly language "wrapper" routines save and restore the necessary CPU registers. Interrupt nesting is fully supported. The CPU vector table is initialized at startup with default values to allow detection of bus errors and stray interrupts.

Standard C Library Support
The TargetOS Standard C library supplies re-entrant stream I/O and memory allocation routines. UART and real-time clock drivers have documented interfaces to support the "stdio.h" stream I/O and "time.h" routines.

Command Line Monitor
The command line monitor displays the state of the kernel objects (tasks, semaphores, etc.) and can be started by entering <CTRL-C> on the TargetOS console. Applications and drivers can extend the monitor with custom commands.

Turnkey Solution
Provided with full source code, default compiler settings, linker command files, user's manuals, benchmarking application, and one year of technical support. Blunk Microsystems offers board support packages and drivers for a range of commercial CPU boards and provides competitive bids on custom board support packages and drivers. ■

Licensing Terms

TargetOS™ is royalty free. Purchasers are granted a non-exclusive license to use the provided source code at a single site. Licensees have the right to disseminate or resell the software in executable format only. The source code and derived object code may not be redistributed or resold.