Operating Systems And Applications For Embedded Systems

eCos

Plan

eCos Core Components

- Hardware Abstraction Layer (HAL)—providing a software layer that gives general access to the hardware.
- Kernel—including interrupt and exception handling, thread and synchronization support, a choice of scheduler implementations, timers, counters, and alarms.
- ISO C and math libraries—standard compatibility with function calls.
- Device drivers—including standard serial, Ethernet, Flash ROM, and others.
- GNU debugger (GDB) support—provides target software for communicating with a GDB host enabling application debugging.

Processor and Evaluation Platform Support

- ARM
- Fujitsu FR-V
- Hitachi H8/300
- Intel x86
- Matsushita AM3x
- MIPS
- NEC V8xx
- PowerPC
- Samsung CalmRISC16/32
- SPARC
- SPARClite
- SuperH

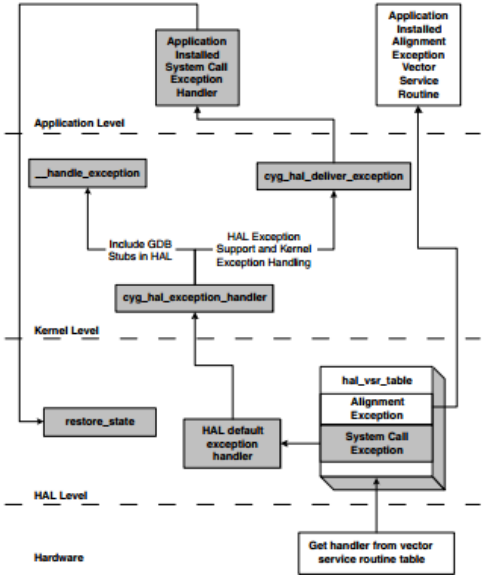
Component Repository Directory Structure Descriptions I

- compat: Contains packages for the POSIX (IEEE 1003.1) and µITRON 3.0 compatibility. cygmon Package contents for Cygmon standalone debug monitor.a devs Includes all device driver hardware-specific components such as serial, Ethernet, and PCMCIA.
- error: Contains common error and status code packages. This allows commonality among packages for error and status reporting.
- fs: Includes the ROM and RAM file system packages.
- hal: Incorporates all HAL target hardware packages.
- infra: Contains the eCos infrastructure such as common types, macros, tracing, assertions, and startup options.
- io: Packages for all generic hardware-independent Input/Output (I/O) system support, such as Ethernet, flash, and serial, which is the basis for system device drivers. isoinfra Contains package that provides support for ISO C libraries (such as stdlib and stdio) and POSIX implementations.

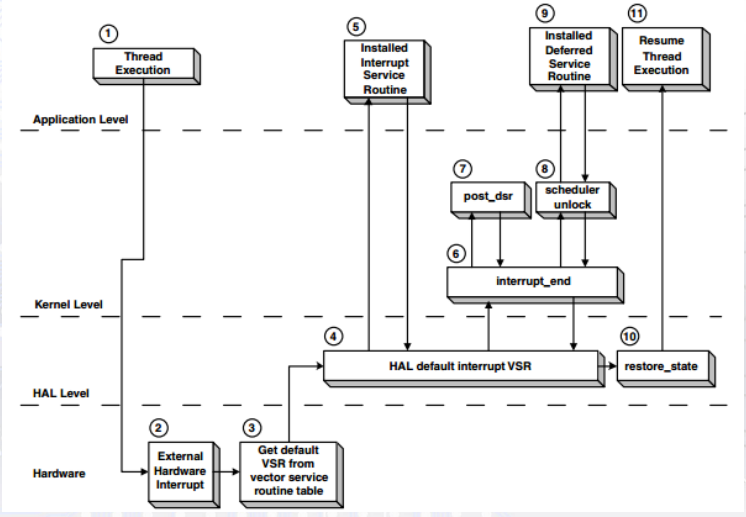
Component Repository Directory Structure Descriptions II

- kernel: Includes the package that provides the core functionality (such as the scheduler, semaphores, and threads) of the eCos kernel.
- language: Incorporates the packages for the ISO C and math libraries, which allows the application to use well-known standard C library functions and the floating-point mathematical library.
- net: Packages for basic networking support including TCP, UDP and IP, and the SNMP protocol and agent support libraries based on the UCD-SNMP project.
- redboot: Contains package for the RedBoot standalone debug ROM monitor. services Includes packages for dynamic memory allocation and support for compression and decompression library.

Exceptions flow



Interrupts flow



Interreupts I

1. The first item in Figure shows the execution of a thread.
2. The next event is an external hardware interrupt.
3. Now, the processor looks into the VSR table, hal_vsr_table, to determine the location of the interrupt vector service routine to execute. During HAL startup, hal_mon_init installs the default interrupt VSR into the VSR table for the external interrupt.
4. Next, the default interrupt VSR begins executing. The first task of the default interrupt VSR is to save the current processor state. As mentioned before, the current processor's state can be saved either on a thread's stack or on the separate interrupt stack, depend ing on the HAL interrupt configuration options selected. After the processor's state information has been stored, the default interrupt VSR incre ments the cyg_scheduler_sched_lock kernel variable to ensure that scheduling does not take place. Next, the default vector service routine needs to find out what ISR to call. ISRs are installed by the application. The HAL uses three tables, implemented as arrays, to maintain the ISR information needed. The size of these tables is architecture specific. The ISR tables are:

Interreupts II

5. hal_interrupt_handlers—contains the addresses of the interrupt service routines installed by the application.
6. hal_interrupt_data—contains the data to be passed into the ISR.
7. hal_interrupt_objects—contains information that is used at the kernel level and hidden from the application layer. The HAL default interrupt VSR uses an architecture-specific function, hal_intc_decode, to perform the lookup into the hal_interrupt_handlers table. This function finds the index into the table based on the interrupt vector number and/or through examining the hardware, such as an interrupt controller. The value is used for indexing into the data (which is passed into the ISR) and objects (used at the kernel level) tables as well.
8. Next, the default VSR calls the ISR installed by the application. The ISR, which executes at the application level, performs any necessary functions for the particular interrupt. The ISR notifies the kernel that the DSR should be posted for execution by returning CYG_ISR_CALL_DSR. The ISR also returns CYG_ISR_HANDLED to terminate any chained interrupt processing.

Synchronization Mechanisms

1. Mutexes
2. Semaphores
3. Condition variables
4. Flags
5. Message boxes
6. Spinlocks (for SMP systems)

Device driver packages

1. Ethernet
2. Flash
3. Compaq IPAQ Platform-Specific Keyboard
4. Compaq IPAQ Platform-Specific Touch Screen
5. Personal Computer Memory Card International Association (PCMCIA)
6. Serial
7. Universal Serial Bus (USB)
8. Watchdog
9. Wallclock

POSIX

- POSIX Scheduling Configuration (CYGPKG_POSIX_SCHED)
- POSIX Pthread Configuration (CYGPKG_POSIX_PTHREAD)
- POSIX Timers (CYGPKG_POSIX_TIMERS)
- POSIX Semaphores (CYGPKG_POSIX_SEMAPHORES)
- POSIX Message Queues (CYGPKG_POSIX_MQUEUES)
- POSIX Signals Configuration (CYGPKG_POSIX_SIGNALS)
- POSIX Utsname Configuration (CYGPKG_POSIX_UTSNAME)

µTRON

- Required (R)—Functions in this level are mandatory for µITRON 3.0 implementations.
- Standard (S)—Includes basic functions for achieving a real-time, multitasking operating system.
- Extended (E)—Includes additional and extended functions, such as object creation and deletion, memory pools, and timer handler functions.
- CPU Dependent (C)—Incorporates CPU or hardware configuration implementation dependent functions.

File Systems

- ROM
- RAM
- JFFS2

OpenBSD

- IPv4—Internet Protocol version 4
- ARP—Address Resolution Protocol
- RARP—Reverse Address Resolution Protocol
- ICMP—Internet Control Message Protocol
- UDP—User Datagram Protocol
- TCP—Transmission Control Protocol
- DHCP—Dynamic Host Configuration Protocol
- BOOTP—Bootstrap Protocol
- TFTP—Trivial File Transfer Protocol

FreeBSD

- IPv4—Internet Protocol version 4
- IPv6—Internet Protocol version 6
- ARP—Address Resolution Protocol
- RARP—Reverse Address Resolution Protocol
- ICMP—Internet Control Message Protocol
- IGMP—Internet Group Management Protocol
- UDP—User Datagram Protocol
- TCP—Transmission Control Protocol
- DHCP—Dynamic Host Configuration Protocol
- BOOTP—Bootstrap Protocol
- TFTP—Trivial File Transfer Protocol
- Multicast addressing

References

Anthony Massa.
Embedded Software Development with eCos.
Prentice Hall Professional Technical Reference, 2002.

The End