C5 Micro-Kernel: Real-Time Services for Embedded and Linux Systems
Outline

- Background
- Jaluna-1 Presentation
- Jaluna-2 Presentation
  - Overview
  - Use Cases
  - Architecture
  - Features
Background

- **Chorus**: started as a research project at INRIA
  - Distributed systems, proprietary API,
- **Chorus Systems**: founded 1986
  - Micro-kernel supporting distributed Unix sub-systems
  - Evolved to embedded real-time systems based on FreeBSD personality
- **Acquired by Sun Microsystems in 1997**
  - Telecom market
- **Jaluna founded August 2002**
  - Jaluna-1: Real-Time Embedded based on FreeBSD
  - Jaluna-2: Real-Time micro-kernel And Linux
Jaluna-1 – Carrier-Grade RTOS

Host System
(Linux or Solaris)

Target Platform

Tool Agents
Applications
C5 APIs/Services
RT-POSIX API/Services
C5 Microkernel
Jaluna-1 Benefits

- Open Source
- Real-time performance for critical applications
- Host-target development and deployment environment
- Standard real-time POSIX Interface
- Future generation networks ready
- Configurable (small/medium/large systems)
- Board support package
- Carrier-grade quality
- First class documentation
- Java Applications (J2ME)
- Migration kits for proprietary RTOS
Jaluna-1 Features

- Development environment
- Real-time operating system
- Board support package
- High availability
- Target platforms
Development Environment

- Solaris 8 and Linux Red Hat 7.3 Hosts
- Tools
  - GNU C and C++ cross development tools
    - gcc 3.2, binutils 2.11, gdb 5.1 with Insight GUI
  - System configuration tools
- Utilities
  - boot monitor (TFTP), console, remote shell
  - benchmarking, monitoring, logging, profiling, system dumps
- System and application interfaces
  - C and C++ Libraries
  - C5 microkernel APIs
  - Complete RT POSIX APIs
Application Types

- Posix multi-threaded processes:
  - benefit from C5 memory management
  - benefit from C5 scheduling features
  - run in user space or supervisor space
    - Restricted Posix in Supervisor: no fork, no signal handlers

- C5 Actors:
  - multi-threaded applications
  - direct access to C5 API's
  - run in user space or supervisor space
Real-time Operating System

- \( C5 \) real-time microkernel
  - Modular scheduler, memory management, communication, synchronization, time services, interrupt management

- RT POSIX system layer (FreeBSD 4.1)
  - Standard (easy migration of UNIX applications)
  - Applications in user or supervisor space
Real-time Operating System

Modular micro-kernel and system:
- System built out of many modules,
- Configuration chosen at system build time,
- Tools available to configure and check dependencies,
- Examples:
  - Memory management:
    - select one kind of management out of three
  - Schedulers:
    - select one out of two
Real-time Operating System

Core executive
- supports multiple, independent user and supervisor applications

Schedulers
- priority based, preemptive FIFO
- priority based round-robin with fixed-time quantum

Memory management
- flat memory
- protected memory
- virtual memory
Real-time Operating System

- **Communications**
  - local or remote Inter Process Communication
    - asynchronous, synchronous, multicast
  - mailboxes
    - local, pre-allocated messages, from interrupt handler

- **Synchronization**
  - semaphores
  - mutexes
  - event flags

- **Time services**
  - one-shot or periodic timers
  - high resolution timer

- **Interrupt management**
Real-time Operating System

- File systems
  - SCSI and IDE drivers
  - UNIX file system
  - MS-DOS file system
  - NFS file sharing (client, server)

- Application management
  - dynamic shared libraries
  - dynamic loading of applications,
    locked in memory, unless swap is configured
  - execution in place
Real-time Operating System

Networking

- Ethernet, serial line, cPCI drivers
- TCP/UDP/IP Stacks
  - IPv4, IPv6 (basic services)
  - Support of multiple network interfaces
  - IP forwarding and multicast
- DHCP (client), DNS (client), tftp, telnet, PPP
- Sun RPC
  - Extended for one-way communication (asynchronous)

Administration

- built-in administration tools (mount, ifconfig, ...)
- command line interpreter (TCL 7.4)
Board Support Package

- **Device Driver Framework**
  - layered drivers structure
  - portability and re-usability of drivers
  - bus communication support
    - Compact PCI
    - hotswap

- **Boot**
  - from local disk, flash, network (DHCP)
  - support embedded user applications
  - fail-safe boot
High-Availability Features

- Device driver framework
  - Hardening
  - Fault injection
- Instrumentation framework
  - attributes [fixed, read only]
  - counters [++, read only]
  - gauges [++/--, read only]
    - watermarks
    - thresholds (crossing a threshold generates an event)
- System events
- Black box
- Watchdog
Target Platforms

- **PowerPC**
  - MPC750 (MCP750, MCPN750)
  - MPC74x0 (MCPN765)
  - MPC850 (MCP850 FADS)
  - MPC860 (MCP860 FADS)
  - MPC8260 (SBC8260)

- **IA32**
  - x86, Pentium II/III (CPN5360, PC compatible)

- **SPARC**
  - UltraSPARC Ili (CP1500, Netra t1)
Real-Time Performance

Pentium III 500 MHz, 128 MB, L2 cache 256 KB

- **Thread switching**
  - **supervisor**: 0.7 µsec
  - **intra user actor**: 1.8 µsec
  - **inter user actors**: 3.5 µsec

- **Semaphore shuffle (with pre-emption)**
  - **supervisor actor**: 1.3 µsec
  - **user actor**: 3.4 µsec

- **Message put (mailbox)**
  - **supervisor actor**: 1.5 µsec
  - **user actor**: 3.3 µsec

- **Kernel preemption**: 0-1 µsec
Real-Time Performance

- Pentium III 500 MHz, 128 MB, L2 cache 256 KB

Kernel Interrupt Latency: 0-1 µs
Entry in IT Handler: 0.6 µs
Semaphore Post: 0.1 µs
Return from IT Handler: 0.5 µs
Switch to higher priority Thread: 1.2 µs

Total Interrupt Response Time < 1.6 µs
Total Thread Response Time < 3.4 µs
Jaluna-1 Software Component Suite

- Migration of proprietary RTOS-based projects
- Packages
  - Jaluna-1 Developer Edition
    - Jaluna Public License
    - Open Source Project
  - Jaluna-1 Carrier-Grade Edition
    - Validation
    - Maintenance and support services
Outline

- Background
- Jaluna-1 Presentation
- Jaluna-2 Presentation
  - Overview
  - Use Cases
  - Architecture
  - Features
Jaluna-2

- Host-target standard Linux distribution extended with Jaluna components
- C5 microkernel (same as Jaluna-1)
- Enables critical added-value in a standard Linux environment
  - Real-Time, High Availability, Security
  - No modification to Linux kernel
- General Availability in June 2003
- Preview Release available [4 February 2003]
Jaluna-2 Components
Jaluna-1, Jaluna-2 & Linux

Deployment Flexibility and Scalability
Outline

- Background
- Jaluna-1 Presentation
- Jaluna-2 Presentation
  - Overview
  - Use Cases
  - Architecture
  - Features
Availability – Linux Monitor

- Prompt fault detection and system recovery
- Software “watchdog”
- Client detect Linux kernel failures
  - kernel panics
  - kernel loop
  - kernel debugger calls
- Server reboots Linux from memory image
Availability – Persistent Memory

- State recovery (checkpoint)
- Persistent Memory Client on Linux
  - accessible through file system
- Persistent Memory Server on C5
  - Local RAM disk
    - survive Linux reboot
  - Persistent RAM disk
    - survive C5 reboot
Real-Time Networking

- Real-Time UDP/IP stack
  - Dedicated to critical traffic (QoS)
  - Guaranteed latency and bandwidth
  - Fine-grain locking
Outline

- Background
- Jaluna-1 Presentation
- Jaluna-2 Presentation
  - Overview
  - Use Cases
  - Architecture
  - Features
Jaluna-2 Architecture
Jaluna-2 Execution Model

- Critical services run on $C5$
- General purpose applications run on Linux
- $C5$ and Linux are coupled with efficient communication services (~ virtual bus)
  - One-way communication channels
  - Asynchronous
  - Reliable
  - Message based
Resource Allocation Policy

- Separate resources to enforce mutual independence and protection
- Assume critical resources requirements are well-known and mainly static
- $C5$ microkernel starts first, reserves dedicated resources
- All remaining resources are available to Linux
Memory Allocation

- Physical memory is statically split in two parts:
  - memory exclusively managed by \( C5 \) microkernel
  - remaining memory exclusively managed by Linux
- Both systems use their native physical memory management
- Both systems use their native virtual memory management
I/O Resources

- I/O and DMA are available from both execution contexts
- Critical devices are managed by C5 drivers
- General purpose devices are managed by Linux drivers
- Shared devices are managed by peer drivers
Outline

- Background
- Jaluna-1 Presentation
- Jaluna-2 Presentation
  - Overview
  - Use Cases
  - Architecture
  - Features
Host-Target Integrated Development Environment

- Cross-development environment
  - GNU C/C++ cross development tools (gcc 3.2.2, glibc 2.2.5)
  - Linux packages cross-installation (RPM based)
  - System configuration tools
  - Remote debugging environment (GNU gdb 5.2.1, Insight GUI)
  - GNU build system
    - autoconf, automake, libtool

- Supported Development Hosts
  - Linux RedHat 8.0
  - Solaris 8.0
Linux General Purpose Environment

(1)

- Tailored to real-time and embedded systems
  - generated, configured and installed on host

- Targets
  - PPC
  - x86

- Linux Operating System
  - kernel 2.4.20
  - Real-time patches
  - RPMS based source delivery
  - packages based on Red Hat source distribution
  - BASIC, NET, DEV package sets
Linux General Purpose Environment (2)

- Embedded Distribution
  - File System, NFS or RAM disk boot
  - simplified and dynamic administration
    - RARP, DHCP, NIS
  - simplified init scripts
  - /dev based on devfs
    - sharable read-only root file system

- Utilities for dynamic Loading $C5$ components

- Multiplexed system console
Linux General Purpose Environment (3)

- High Availability Extensions
  - Fast file system check upon recovery (ext3)
  - Linux fault detection and fast reboot
  - Fast and persistent RAM disk across Linux reboot
  - Heartbeat for node failure detection and recovery
  - Disk mirroring
  - Check pointing
C5 Microkernel Environment

- C5 microkernel services identical to Jaluna-1:
  - User / Supervisor actors
  - Real-time schedulers (FIFO, Round-Robin)
  - Communications: IPC, mailboxes, real-time message queues
  - Synchronization: semaphores, mutexes, events
  - Time Services
  - Device Driver Framework (DDI/DKI)

- Limited Real-time POSIX API:
  - posix threads, posix timers,
  - no files, no process, no signals

- Real-time networking