Realtime and Linux

Ingo Molnar, Red Hat
Paul E. McKenney, IBM LTC

2005 Linux Kernel Summit
Ottawa, Canada
July 18-19, 2005

Overview

- Realtime Desiderata
- Linux Realtime Approaches
- Realtime-Related Patches & Components
- Other Things RT Users Might Ask For
- Discussion

Realtime Desiderata

- Quality of Service (Beyond “Hard”/“Soft”)
  - Services Supported
    - Probability of meeting deadline absent HW failure
    - Deadlines supported
  - Performance/Scalability for RT & non-RT Code
- Amount of Global Knowledge Required
- API Provided (POSIX? Ad Hoc?)
- OS/Application Relative Complexity
- Fault Isolation
- HW/SW Configurations Supported

- “But Will People Use It?”

Linux Realtime Approaches (Violently Abbreviated)

Project	Quality of Service	Inspection	API	Complexity	Fault Isolation	HW/SW Configs
Vanilla Linux Kernel	10s of ms all services	All	POSIX + RT extensions	N/A	None	All
PREEMPT	100s of us Schd, Int	All spinlock critsect, preempt- & int-disable	POSIX + RT extensions	N/A	None	All
Nested OS	~10 us RTOS svs	RTOS + int-disable	RTOS	Dual environment	Good	All
Dual-OS / Dual-Core	<1 us RTOS svcs	All RTOS	RTOS	Dual environment	Excellent	Specialized
PREEMPT_RT	10s of us Schd, Int	All preempt- & int-disable (most ints in process ctxt)	POSIX + RT extensions	“Modest” patch	None	All (except some drivers)
Migration Between OSes	? us RTOS svcs	All RTOS + int-disable	RTOS (can be POSIX)	Dual env. (Fusion)	OK	All?
Migration Within OS	? us RTOS svcs	Scheduler + RT syscalls	POSIX + RT extensions	Small patch	None	All?

Examples of Linux Approaches

- Nested OS:
  - RTLinux, L4Linux, I-pipe (latency from RTLinux)
- Dual-OS/Dual-Core:
  - Huge numbers of real products, e.g., cell phones
- Migration Between OSes:
  - RTAI-Fusion
- Migration Within OS:
  - ARTiS (Asymmetric Real-Time Scheduling)

Related Patches & Components

- High-Resolution Timers (HRT)
  - Avoids the “three-millisecond shuffle”
  - Additional code provides fine-grained timers
- Variable idle Sleep Time (VST)
  - Suppress unneeded timer ticks, CONFIG_VST
  - Also helps virtualization/consolidation
- fusyn implementation
  - Priority inheritance for user-level mutexes
    - Such as pthread_mutex
- isolcpus
- Interrupt-shielding patches & config options

Other Things RT Users Might Ask For

- Deterministic I/O
  - Disk I/O – or, more likely, flash memory
  - Network protocols
    - Datagram protocols (UDP) relatively straightforward
    - “Reliable” protocols (TCP, SCTP) more difficult
- Other priority inheritance
  - Across memory allocation
    - Boost priority of someone who is about to free...
  - Reader-writer locks with concurrent readers
    - Writer-to-reader boosting quite complex...
  - Across RCU
    - Boost priority of RCU readers when OOM

Discussion

- Observations:
  - Many still consider realtime response from a general-purpose OS to be impossible.
    - Might be why we only have 7 different approaches.
  - Incremental philosophy works well
    - Each approach is incremental
    - Approaches not necessarily mutually exclusive
  - Much overlap between realtime and high end
    - CONFIG_VST
    - SMP testing on UP machines with PREEMPT_RT
    - Paul expects increasing need for realtime response from moderate SMP systems