Real-Time Systems

Lecture Topic – POSIX Real-Time Extensions

Dr. Sam Siewert

Electrical, Computer and Energy Engineering

Embedded Systems Engineering Program

Software Support for Real-Time

Cyclic Executives (Main loop + ISR)

- Multi-frequency executives
- May not scale well to large numbers of services and millions of lines of code

RTOS

- Possibly POSIX compliant (e.g. QNX)
- No user and kernel space
- Simple, but requires BSP and driver work

POSIX OS

- higher overhead
- many software and systems engineering advantages

Table 3: POSIX in commercial operating systems

OS	POSIX 1003.1a (Base POSIX)	POSIX 1003.1b (Real-time extensions)	POSIX 1003.1c (threads)
Solaris	Full support	Full support	Full support
LynxOS	Conformant	Full support	3.0.1 based on draft and missing thread attributes; 3.1 based on final standard
VxWorks	Partial support; support for functions that do not require a process model	Partial support; support for functions that do not require a process model	Support through a third party product
IRIX	Conformant	Full support	Full support
Linux	Full support	Partial support; no support for timers or message queues	Full support

Obenland, Kevin M. "The use of posix in real-time systems, assessing its effectiveness and performance." *The MITRE Corporation* (2000).

Summary of Extensions

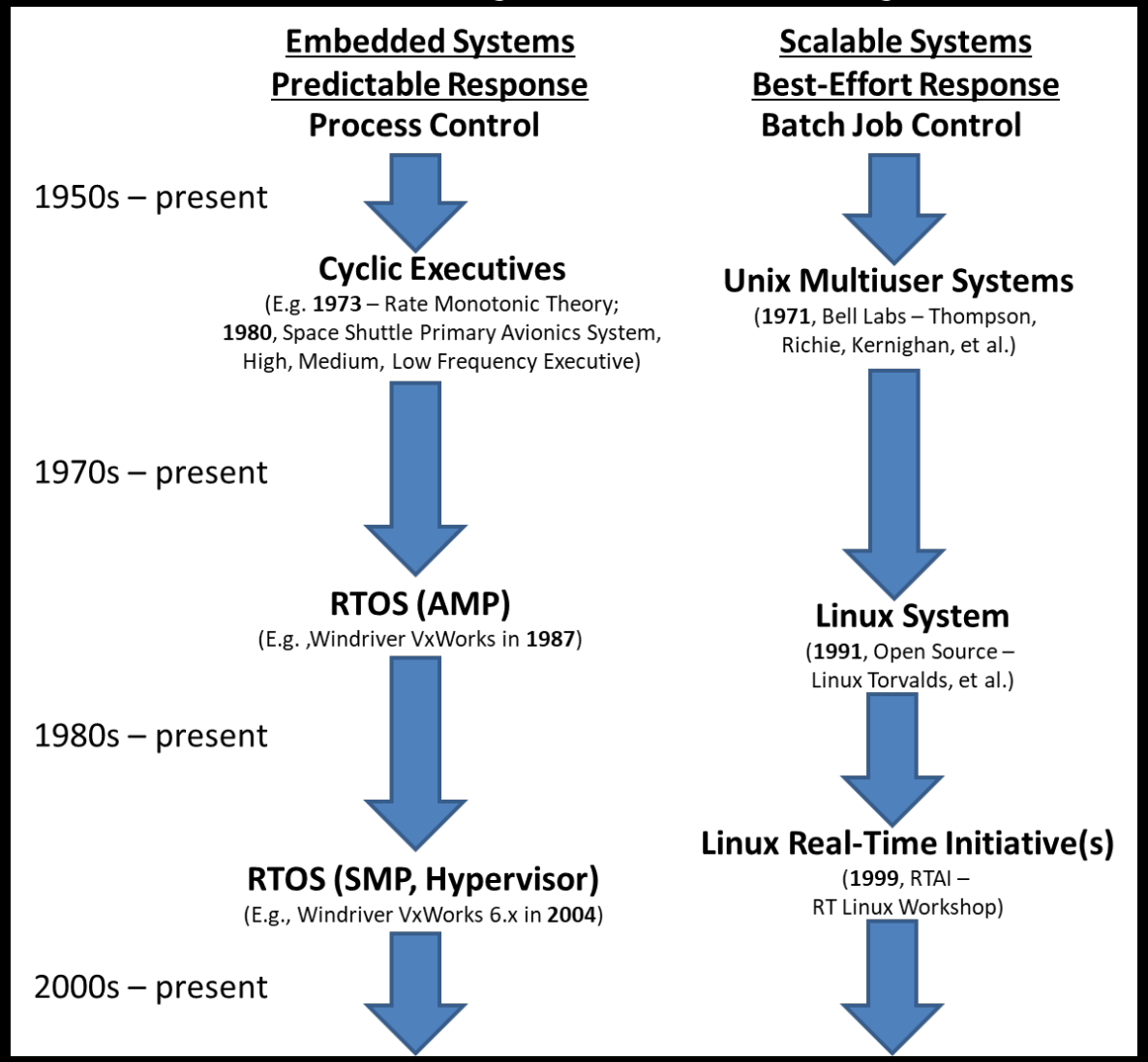
The Real-time amendments to IEEE Std 1003.1-1990 are as follows:

- IEEE Std 1003.1b-1993 Realtime Extension
- IEEE Std 1003.1c-1995 Threads
- IEEE Std 1003.1d-1999 Additional Realtime Extensions
- IEEE Std 1003.1j-2000 Advanced Realtime Extensions
- IEEE Std 1003.1q-2000 Tracing

Important Software Mechanisms

- Priority preemptive, run-to-completion scheduling (SCHED_FIFO)
- Dynamic priority scheduling with EDF (SCHED_DEADLINE)
- Real-time signals that queue
- RT Clocks and interval timers
- Semaphores with RT features (PIP, PCP, PCEP)
- Message Queues with RT features
- Shared memory (memory locking)
- Sync and Async I/O
- Logging and Tracing

History of RT Systems and General OS



Current Trends for RTOS and OS+RT

1) RTOS used for HRT, Mission Critical
2) OS + RT for SRT and Scalable
3) RTOS are adopting more General OS features
   - Drivers (App interface, Device interface)
   - Filesystems
   - Databases
   - GUIs
   - Network stacks (TCP/IP)
   - VM and SMP configurations
   - Multi-core
   - Security features
4) OS RT Extensions now Standard
   - POSIX 1003.1
   - Emergent Apps like Driver Assistants, Vehicle Autopilots
   - Large scale RTES for military, transportation, telecom
5) IoT – Nano and Micro Kernels
   - Zephyr RTOS - https://www.zephyrproject.org
   - FreeRTOS - https://www.freertos.org