Real-Time Operating Systems/Linux

David L. Levine
Washington University, St. Louis
levine@cs.wustl.edu

26 April 2000

http://www.cs.wustl.edu/~levine/courses/cs422/rtos.ps.gz
Overview

- Motivation and context
- The role of the OS
- Limitations of conventions OS’s
- Real-time OS’s (RTOS’s)
- Performance comparisons of conventional and RT OS’s
Motivation for OS and Middleware

- **OS Role**
  - OS provides resource management and abstractions
  - Middleware provides higher-level building blocks

- **Reuse**
  - Buy vs. build
  - The lower the layer, the more difficult and error-prone the code. (And the lowest layer is hardware dependent.)
What is the OS Role?

• Provides services used by applications (and middleware)
  – Reuse at a fundamental level
  – The services are difficult to write, esp. for correctness, efficiency, portability

• Resource management
  – Hardware resource (CPU, memory, peripheral device) control and insulation
  – Software resource (I/O handle, synchronization primitive, communication port) control

• Features to support distribution (concurrency)
  – synchronization
  – communication
OS Limitations

- Does the concurrency support scale?
  - Do 1000 processors give 1000x speedup?

- Does the OS allow fair sharing of resources?
  - Should all sharing be fair?

- Can the OS footprint be minimized?

- Does the OS provide standard interfaces?
RTOS Capabilities

- Prioritized resource access
- Scheduling based on priority (or timeline)
- Predictable operation
- Very small size (under 1 Kb to a few hundred Kb)
  - Compare to 1.5 Mb of Linux 2.2.12 (Red Hat 6.1)
- High-level language (C++, Java) support
The RTOS Market

- Leader is VxWorks by Wind River
- Number 2 was pSOS by Integrated Systems
- POSIX: LynxOS, QNX Neutrino
- Windows and Linux compatibility
  - RT versions really aren’t
  - RT compatible versions are coming, e.g., Blue Cat Linux
Performance Experiment

- One 20 Hz high-priority client
- 1..n 10 Hz low-priority clients
  - Increasing $n$ increases load
- Server factory implements thread-per-connection
  - Each connection links client with its servant

http://www.cs.wustl.edu/~levine/research/RT-OS.ps.gz
High-Priority Request Latency Results

- Synopsis of results
  - LynxOS provides consistently low and predictable latency
  - VxWorks does not scale on x86
  - Non-RTOS’s are not predictable
  - ORB (TAO) provides low latency and avoids priority inversion
    * i.e., high priority client always has lowest latency
Performance is Maintained with Distribution

Server and Client on Same CPU

Server and Client on Different CPUs

Copyright ©1997-2000  Dept. of Computer Science, Washington University
Do We Need to Use an RTOS?

- Depends on application, but in general, probably not
- However, RTOS’s are often used as lightweight OS’s
- We can use middleware for services, and use any suitable OS
Linux Support for Real-Time

- POSIX.1 IPC, threads, scheduling, synchronization, etc.
- Both Process (`sched_`) and thread (`pthread_`) scheduling

- Scheduling modes:
  - `SCHED_FIFO` (run-to-completion/block)
  - `SCHED_RR` (round-robin, with time quantum)
  - `SCHED_OTHER` (time sharing, dynamic based on nice setting and block time)

- Priority levels
  - 1 to 99 for `SCHED_FIFO` and `SCHED_RR`
  - 0 for `SCHED_OTHER`
Linux Threads

- User threads are implemented as processes
  - `clone` system call forks a process, but allows sharing of process ID, memory, file descriptors, file system info, and signal handlers.
  - Process fork time: $110 \mu\text{sec}$ on 450 MHz Pentium II
- Thread context switch does not invalidate page tables. (Kernel checks for whether the context switch is within the same process.)
- Thread context switch: 1) restore registers and TSS base, then 2) jump to thread’s IP
- Thread context switch time: $1.2 \mu\text{sec}$ on 450 MHz Pentium II
- Signal handling is slightly different between threads and processes.
- Kernel threads introduced in Linux 2.2, full support in 2.3/2.4
Linux 2.0 Kernel Source Structure

- **Features**
  - Separation of concerns/encapsulation
  - Configuration support
  - Modular (except for Alphas)
<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>acct.c</td>
<td>9079</td>
<td>2000-04-02</td>
</tr>
<tr>
<td>capability.c</td>
<td>6360</td>
<td>1999-12-08</td>
</tr>
<tr>
<td>dma.c</td>
<td>2861</td>
<td>2000-01-26</td>
</tr>
<tr>
<td>exec_domain.c</td>
<td>3474</td>
<td>2000-03-13</td>
</tr>
<tr>
<td>exit.c</td>
<td>13411</td>
<td>2000-04-02</td>
</tr>
<tr>
<td>fork.c</td>
<td>18190</td>
<td>2000-04-02</td>
</tr>
<tr>
<td>info.c</td>
<td>737</td>
<td>1999-08-23</td>
</tr>
<tr>
<td>itimer.c</td>
<td>4006</td>
<td>2000-02-10</td>
</tr>
<tr>
<td>kmod.c</td>
<td>5856</td>
<td>2000-04-02</td>
</tr>
<tr>
<td>ksysms.c</td>
<td>13800</td>
<td>2000-04-07</td>
</tr>
<tr>
<td>module.c</td>
<td>22498</td>
<td>2000-03-11</td>
</tr>
<tr>
<td>panic.c</td>
<td>2366</td>
<td>2000-04-11</td>
</tr>
<tr>
<td>pm.c</td>
<td>6554</td>
<td>2000-03-15</td>
</tr>
<tr>
<td>printk.c</td>
<td>11693</td>
<td>2000-03-10</td>
</tr>
<tr>
<td>ptrace.c</td>
<td>3588</td>
<td>2000-03-21</td>
</tr>
<tr>
<td>resource.c</td>
<td>7160</td>
<td>2000-02-29</td>
</tr>
<tr>
<td>sched.c</td>
<td>27178</td>
<td>2000-03-27</td>
</tr>
<tr>
<td>signal.c</td>
<td>25200</td>
<td>2000-03-23</td>
</tr>
<tr>
<td>softirq.c</td>
<td>5954</td>
<td>2000-03-05</td>
</tr>
<tr>
<td>sys.c</td>
<td>27206</td>
<td>2000-03-19</td>
</tr>
<tr>
<td>sysctl.c</td>
<td>31983</td>
<td>2000-03-23</td>
</tr>
<tr>
<td>time.c</td>
<td>12220</td>
<td>1999-10-17</td>
</tr>
<tr>
<td>timer.c</td>
<td>20898</td>
<td>2000-02-10</td>
</tr>
<tr>
<td>uid16.c</td>
<td>4073</td>
<td>2000-01-11</td>
</tr>
</tbody>
</table>
# Linux Kernel Runtime Structure

Visible through `/proc` filesystem

<table>
<thead>
<tr>
<th>Directory</th>
<th>Mount Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/</td>
<td></td>
</tr>
<tr>
<td>1476/</td>
<td></td>
</tr>
<tr>
<td>1478/</td>
<td></td>
</tr>
<tr>
<td>1479/</td>
<td></td>
</tr>
<tr>
<td>193/</td>
<td></td>
</tr>
<tr>
<td>2/</td>
<td></td>
</tr>
<tr>
<td>202/</td>
<td></td>
</tr>
<tr>
<td>213/</td>
<td></td>
</tr>
<tr>
<td>224/</td>
<td></td>
</tr>
<tr>
<td>2262/</td>
<td></td>
</tr>
<tr>
<td>235/</td>
<td></td>
</tr>
<tr>
<td>246/</td>
<td></td>
</tr>
<tr>
<td>2487/</td>
<td></td>
</tr>
</tbody>
</table>

- `cmdline` | `mounts` |
- `cpuinfo` | `net/` |
- `devices` | `pci` |
- `dma` | `scsi/` |
- `filesystems` | `self@` |
- `interrupts` | `stat` |
- `iports` | `sys/` |
- `kcore` | `uptime` |
- `kmsg` | `version` |
- `loadavg` |
- `locks` |
- `meminfo` |
- `misc` |
Linux 2.0 Process Info

```
/proc$ ls self
  cmdline environ fd/    mem    stat    status
  cwd@    exe@    maps@    root@    statm

/proc$ cat self/status
Name:    cat
State:   R (running)
Pid:     2404
PPid:    2398
Uid:     1046 1046 1046 1046
Gid:     1213 1213 1213 1213
VmSize:  2056 kB  VmStk:    8 kB  SigBlk:    00000000
VmLck:   0 kB  VmExe:    16 kB  SigIgn:    00000000
VmRSS:   560 kB  VmLib:    856 kB  SigCgt:    00000000
VmData:  56 kB  SigPnd:   00000000
```
## Linux 2.0 System Processes

<table>
<thead>
<tr>
<th>PID</th>
<th>TTY</th>
<th>STAT</th>
<th>TIME</th>
<th>COMMAND</th>
<th>PID</th>
<th>TTY</th>
<th>STAT</th>
<th>TIME</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?</td>
<td>S</td>
<td>0:08</td>
<td>init</td>
<td>246</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>inetd</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>SW</td>
<td>0:00</td>
<td>(kflushd)</td>
<td>257</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>lpd</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>SWN</td>
<td>0:00</td>
<td>(kswapd)</td>
<td>269</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>rpc.mountd</td>
</tr>
<tr>
<td>4</td>
<td>?</td>
<td>SW</td>
<td>0:00</td>
<td>(nfsiod)</td>
<td>278</td>
<td>?</td>
<td>S</td>
<td>0:04</td>
<td>rpc.nfsd</td>
</tr>
<tr>
<td>5</td>
<td>?</td>
<td>SW</td>
<td>0:00</td>
<td>(nfsiod)</td>
<td>292</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>gpm -t ps/2</td>
</tr>
<tr>
<td>6</td>
<td>?</td>
<td>SW</td>
<td>0:00</td>
<td>(nfsiod)</td>
<td>303</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>/usr/local/sbin/sshd</td>
</tr>
<tr>
<td>7</td>
<td>?</td>
<td>SW</td>
<td>0:00</td>
<td>(nfsiod)</td>
<td>310</td>
<td>1</td>
<td>S</td>
<td>0:00</td>
<td>/bin/login -- levine</td>
</tr>
<tr>
<td>67</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>kerneld</td>
<td>311</td>
<td>2</td>
<td>S</td>
<td>0:00</td>
<td>/sbin/minigetty tty2</td>
</tr>
<tr>
<td>193</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>syslogd</td>
<td>312</td>
<td>3</td>
<td>S</td>
<td>0:00</td>
<td>/sbin/minigetty tty3</td>
</tr>
<tr>
<td>202</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>klogd</td>
<td>313</td>
<td>4</td>
<td>S</td>
<td>0:00</td>
<td>/sbin/minigetty tty4</td>
</tr>
<tr>
<td>213</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>atd</td>
<td>314</td>
<td>5</td>
<td>S</td>
<td>0:00</td>
<td>/sbin/minigetty tty5</td>
</tr>
<tr>
<td>224</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>crond</td>
<td>315</td>
<td>6</td>
<td>S</td>
<td>0:00</td>
<td>/sbin/minigetty tty6</td>
</tr>
<tr>
<td>235</td>
<td>?</td>
<td>S</td>
<td>0:00</td>
<td>portmap</td>
<td>317</td>
<td>?</td>
<td>S</td>
<td>0:05</td>
<td>update (bdflush)</td>
</tr>
</tbody>
</table>
## Linux System Info

```bash
/proc$ cat /proc/version
Linux version 2.0.34 (root@linux00.amt.tay1.dec.com)
```

```bash
/proc$ cat cpuinfo

cpu : Alpha
cpu model : EV5
cpu variation : 0
cpu revision : 0
cpu serial number : Linux_is_Great!
system type : EB164
system variation : 0
system revision : 0
system serial number : MILO-0000
cycle frequency [Hz] : 0
timer frequency [Hz] : 1024.00
page size [bytes] : 8192
phys. address bits : 40
max. addr. space # : 127
BogoMIPS : 497.02
kernel unaligned acc : 0 (pc=0,va=0)
user unaligned acc : 244999 (pc=12005056c,va=12005056c)
platform string : N/A
```
Linux System Info, continued

```
/proc$ cat /proc/interrupts
  1:  60316   keyboard
  4:  27127 + serial
 12: 269083   PS/2 Mouse
 16: 390908  3c905 Boomerang 100baseTx
 19: 14555   aic7xxx
```
Linux System Status

```
/proc$ cat /proc/loadavg
0.00 0.00 0.00 2/56 2425

/proc$ cat /proc/meminfo
    total:  used:  free:  shared:  buffers:  cached:
Mem:  262324224 111017984 151306240 63127552 18161664 55885824
Swap: 534634496 0 534634496
MemTotal:  256176 kB
MemFree:   147760 kB
MemShared:  61648 kB
Buffers:   17736 kB
Cached:    54576 kB
SwapTotal: 522104 kB
SwapFree:  522104 kB
```
RT Linux

- Victor Yodaiken’s RTLinux
  - Small real-time kernel runs Linux as (lowest priority) application
  - Real-time and Linux tasks can communicate via queues and shared memory (without using locks)

- Lynx’s BlueCat Linux
  - Derivative of RedHat 6.1 (2.2.12-20 kernel) that provides binary compatibility with LynxOS applications