Linux Kernel 2.6: New Features

Jerry Cooperstein

Axian Inc

coop@axian.com

@ MWVLUG 2/04/03

On-line

Download this from:

http://www.axian.com/learning.php

Linux Kernel 2.6

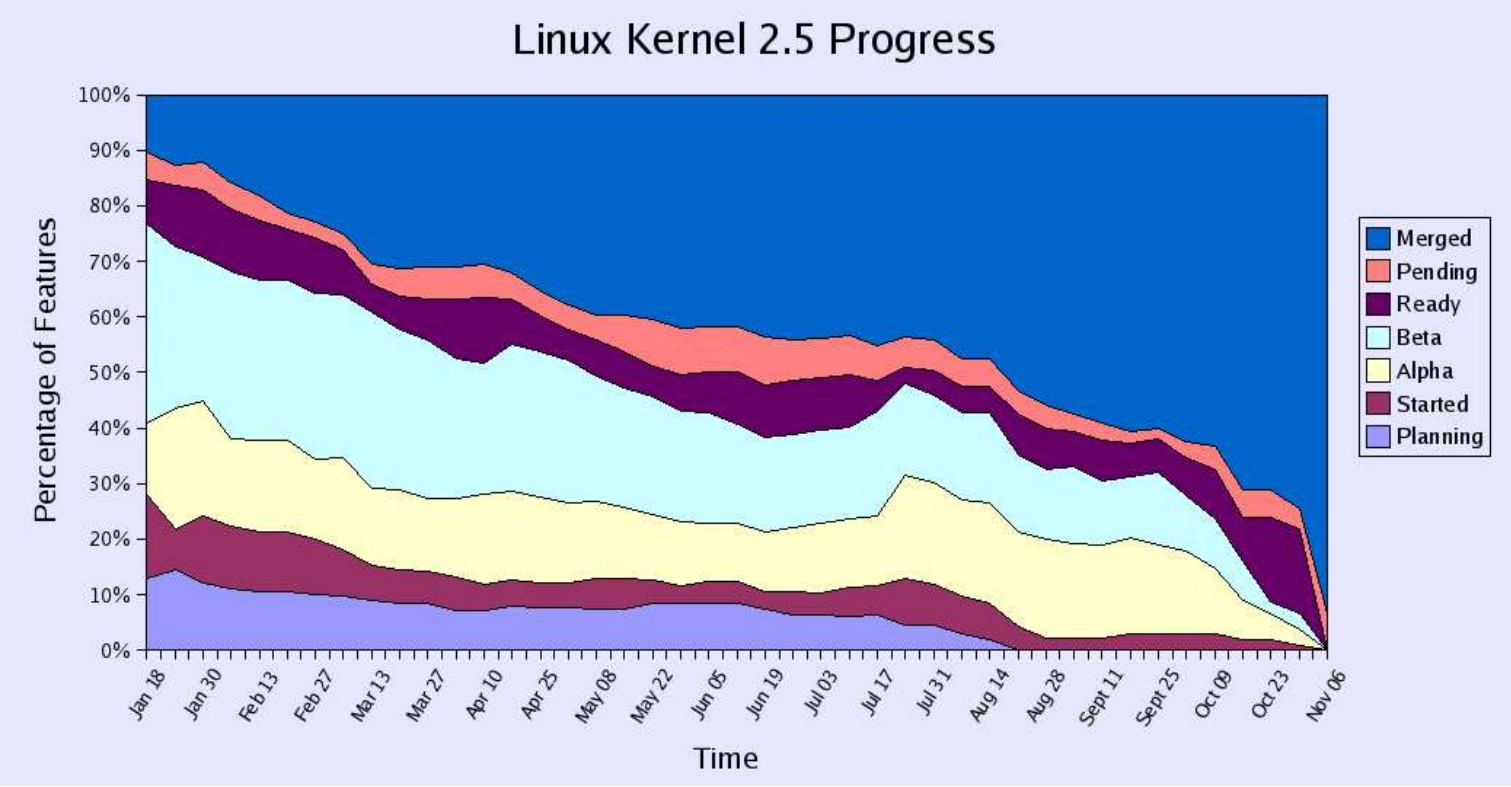
- Feature Freeze: Halloween 2002
- Better Performance, Especially on **SMP**
- Better Scalability
- Better I/O Subsystem, New Filesystems
- Many New Hardware Drivers
- New Platforms
- Many Features Tested as 2.4 Patches

Latest Status

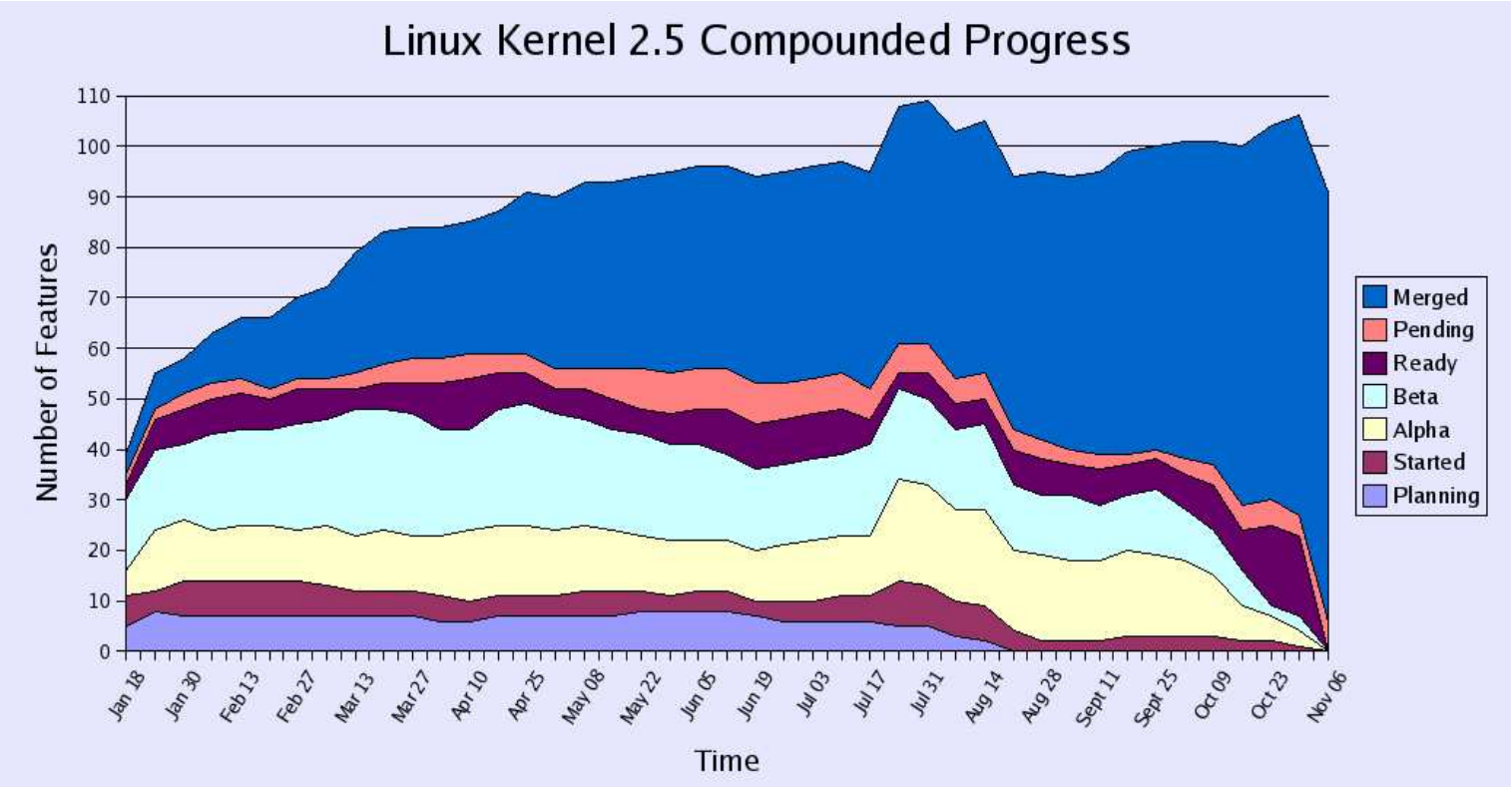
- Guillaume Boissiere maintains a status report, updated weekly at:

  http://kernelnewbies.org/status/latest.html

Boissiere's Status Report: Progress



Boissiere's Status Report: Compounded Progress



Linux Kernel History:
Release Dates, Lines of Code

- 0.01: 09/1991 7.5 K
- 1.0: 03/1994 158 K
- 1.2: 03/1995 277 K
- 2.0: 07/1996 649 K
- 2.2: 01/1999 1536 K
- 2.4: 01/2001 2888 K
- 2.6: ??/2003 ~4200 K

New Features: General

- Preemptable Kernel
- O(1) Scheduler
- New Kernel Device Structure (**`kdev_t`**)
- Improved Posix Threading Support (NGPT and NPTL)
- New Driver Model & Unified Device Structure*

*not discussed today

New Features: General

- Faster Internal Clock Frequency
- Paring Down the BKL (Big Kernel Lock)
- Better in Place Kernel Debugging
- Smarter IRQ Balancing*
- ACPI Improvements*
- Software Suspend to Disk and RAM*

*not discussed today

New Features: General

- Support for USB 2.0*
- ALSA (Advanced Linux Sound Architecture)*
- LSM (Linux Security Module)*
- Hardware Sensors Driver (lm-sensors)*

*not discussed today

New Features: Architectures

- AMD 64-bit (x86-64)
- PowerPC 64-bit (ppc64)
- User Mode Linux (UML)

New Features: Journalling Filesytems

- Ext3 (already in 2.4)
- ReiserFS (already in 2.4)
- JFS (IBM)
- XFS (SGI)

New Features: General

- CPU Clock and Voltage Scaling
- Setting Processor Affinity
- Improved **NUMA** Support*
- Reverse Mapping VM System (**rmap**)
- Large Page Support*
- High Resolution Posix Timers*
- New Serial Port Driver Rewrite and API*

*not discussed today

Other New Items

- New Module Implementation and Utilities
- New System Call Mechanism

New Features: I/O Layer

- Rewrite of Block I/O Layer (BIO)
- Asynchronous I/O*
- IDE Layer Update
- ACL Support (Access Control List)*
- New NTFS Driver*

*not discussed today

New Features: Networking

- NFS v4*
- Zero-Copy NFS*
- TCP Segmentation Offload*
- SCTP Support*
  (Stream Control Transmission Protocol)
- Bluetooth Support (not experimental)*
- NAPI (Network Interrupt Mitigation)*

*not discussed today

Removed Features

- Export of **`sys_call_table`**
- End of **Task Queues**

Preemptable Kernel

- Robert Love **www.kernel.org/pub/linux/kernel/people/rml** (original patch from MontaVista)
- Old: No Kernel Preemption
  - Execution in kernel mode interrupted only by explicit yields, sleeps, and IRQ's
- New: Kernel May Be Preempted
  - New process may be swapped in after servicing an interrupt

Preemptable Kernel

- **OLD:**
  - Kernel executing code for process A
  - Services interrupt
  - Returns to Process A
- **NEW:**
  - Kernel executing code for Process A
  - Services interrupt
  - Returns to process A, B, C, …..

Preemptable Kernel

- Modify spinlocks for Preemption:

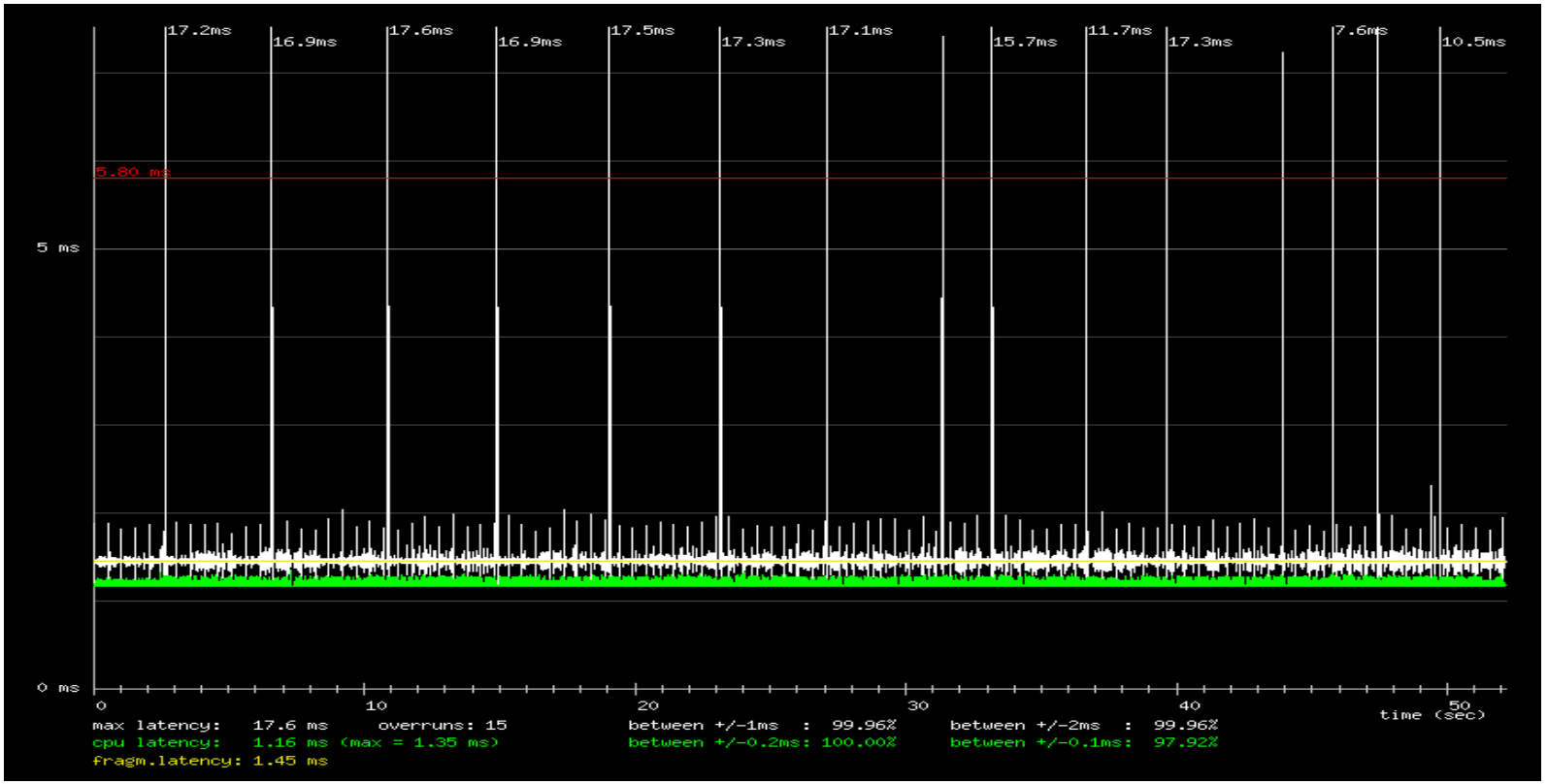
```
spin_lock(lock)          ->
            preempt_disable()
            _raw_spin_lock(lock)
      spin_unlock(lock)      ->
            _raw_spin_unlock(lock)
            preempt_enable()
```

- **`preempt_disable(), preempt_enable()`**
  - increment or decrement a counter, and introduce a memory **`barrier()`** call

Preemptable Kernel

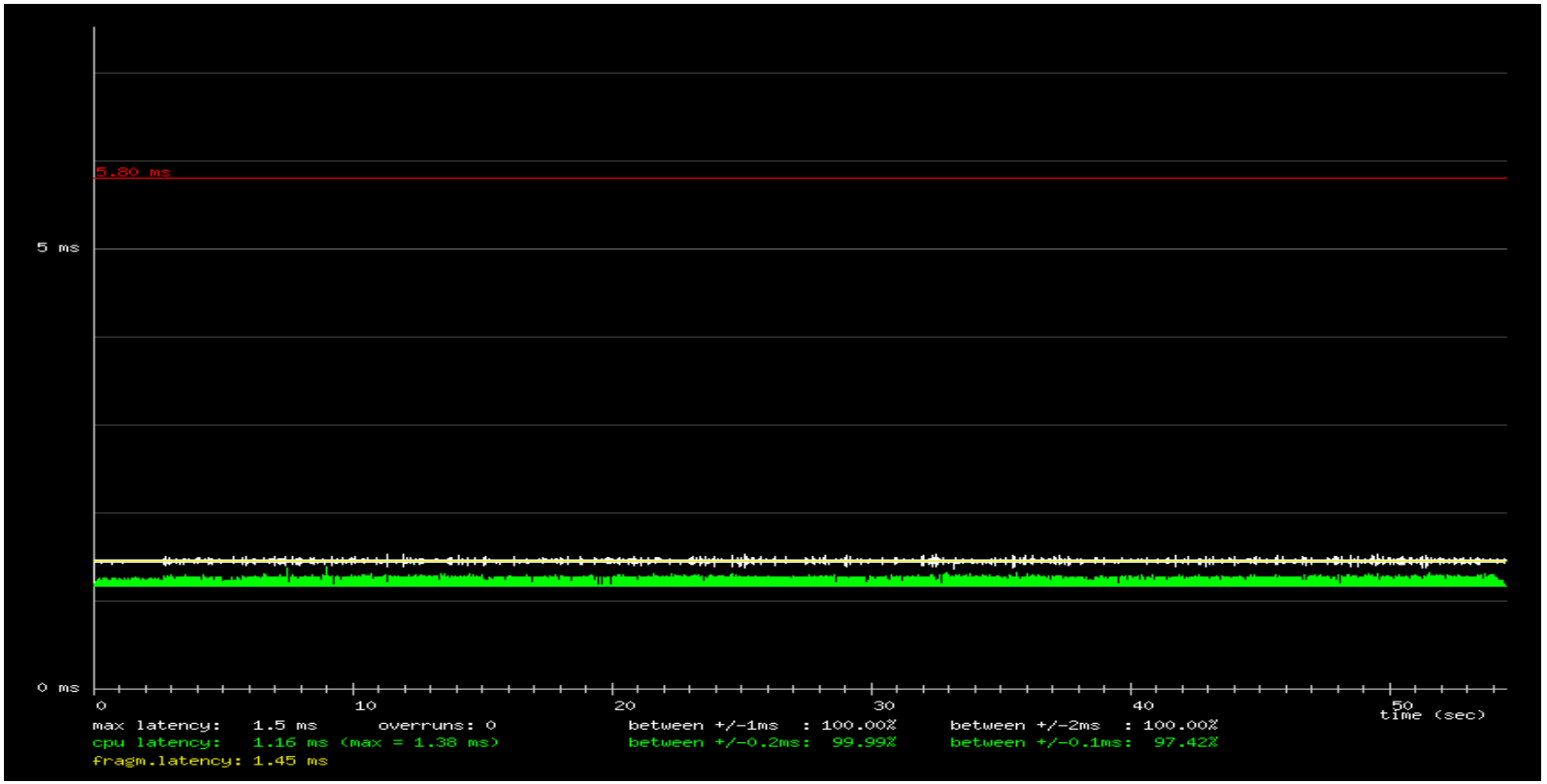
- Kernel must be fully re-entrant
- UP systems will show SMP problems (good for debugging)
- System latency greatly reduced

Latencies with Preemption Off



Linux Magazine, May 01, 2002

Latencies with Preemption On



Linux Magazine, May 01, 2002

O(1) Scheduler

- Ingo Molnar
  (www.kernel.org/pub/linux/kernel/people/mingo)
- Old: Scan all processes, CPU's: O(N)
- New: Maintain 2 queues per CPU - active and expired processes
- Sort by priority - no search: O(1)
- Real time processes share one queue

O(1) Scheduler

- Already included in Red Hat 7.3
- Much better scalability with SMP
- Some (Old) Results:

```
  01/21/2002

Here are some results from running VolanoMark on different
versions of O(1)-scheduler based on 2.4.17.
Volcanomark is a Java(TM) chatroom benchmark: multiple rooms, where for
   each room, every input from a client generates a write to every
   other client (think broadcast storm).
Partha Narayanan, partha@us.ibm.com
```

O(1) Scheduler

VolanoMark 2.1.2 Loopback test,
8-way 700MHZ Pentium III,
1GB Kernel,
IBM JVM 1.3. (build cx 130 -20010626)
Throughput in msg/sec

KERNEL	UP	4-way	8-way
2.4.17	11005	15894	11595
2.4.17 + D2 patch	10606	23300	29726
2.4.17 + G1 patch	10415	23038	31098
2.4.17 + H6 patch	10914	22270	32300
2.4.17 + H7 patch	11018	23427	31674
2.4.17 + J2 patch	13015	23071	33259

Partha Narayanan, partha@us.ibm.com

New Kernel Device Structure (`kdev_t`)

- 16-bit **`dev_t`** seen by **`knod(),stat()`**:
  - 8-bit major No. (driver)
  - 8-bit minor No. (instance, mode, device)
- **`kdev_t`** is new internal kernel structure
- For now just:
  `struct{ushort major, minor}`

New Kernel Device Structure (`kdev_t`)

- Eventually will have more information:
  - device, block and sector sizes
  - name, flags, methods jump table, etc.
- Eventually:
  - 20-bit major numbers
  - 12-bit minor numbers

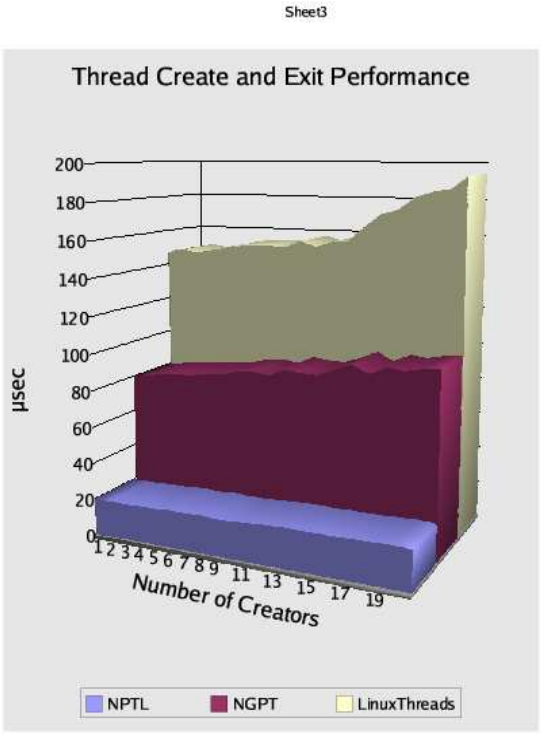
Improved Posix Threading Support

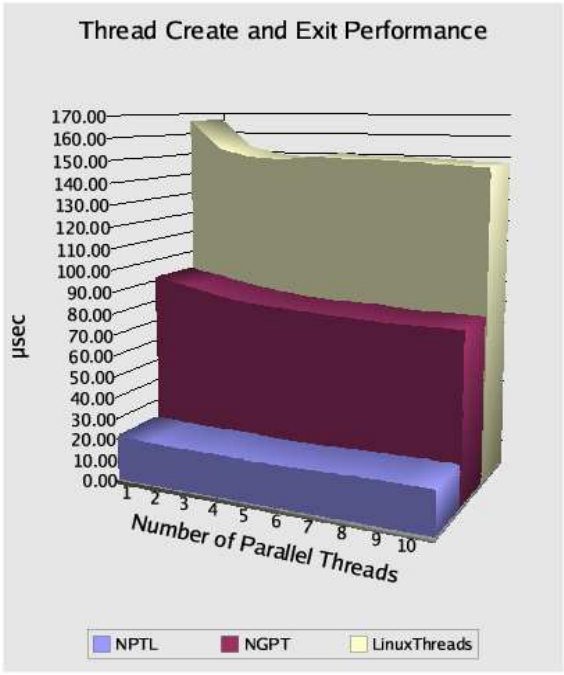
- Next Generation Posix Threads (NGPT)
- Drop in Replacement for LinuxThreads
- Better POSIX Compliance
- Better performance on SMP
- Mostly user space; needed kernel mods
- **http://www.124.ibm.com/developerworks/oss/pthreads**

Improved Posix Threading Support

- Native Posix Thread Library (NPTL)
- Ingo Molnar and Ulrich Drepper
- 1:1 pure kernel thread model (NGTP is M:N, M user threads per N kernel threads)
- Requires 2.536+K + gcc 3.2 + glibc 2.3
- **http://people.redhat.com/drepper/nptl-design.pdf**

Improved Posix Threading





- http://people.redhat.com/drepper/perf-s-100000-pro{par}.pdf

Faster Internal Clock Frequency

- Raise HZ to 1000
  - More frequent switching
  - Better interactive response

```
/* Internal kernel timer frequency */
 # define HZ   1000
/*.. some user interfaces are in ticks"*/
# define USER_HZ 100
/* like times() */
# define CLOCKS_PER_SEC  (USER_HZ)
```

Paring Down the BKL (Big Kernel Lock)

- Finer-grain control over locking
- Better scalability, SMP
- Tedious task
- Prone to intermittent errors
- Will take a long time to complete
- BKL will remain in limited role

BKL Removal (Example)

- **OLD:**

```
lock_kernel();
critical code;
unlock_kernel();
```

- **NEW:**

```
spin_lock(&lock_A);
critical code
 spin_unlock(&lock_A);
spin_lock(&lock_B);
critical code;
spin_unlock(&lock_B);
```

Better in Place Kernel Debugging

CONFIG_PREEMPT=y **CONFIG_DEBUG_SLAB=y**

CONFIG_DEBUG=y **CONFIG_DEBUG_SPINLOCK=y**

CONFIG_KALLSYMS=y

- Check if sleeping in atomic code
- No need to run **ksymoops**
- Poison memory on freeing
- Also possibly: LTT (Linux Trace Toolkit) and DProbes (Dynamic Probes)

AMD 64-bit (x86-64)

- “Hammer”
- Supports IA32 binaries
- 4 KB pages with 4 level Page Tables
- Linux has 3 levels: leads to 40 bits per user process (1 TB)
- **http://www.x86-64.org**

PowerPC 64-bit (ppc64)

- Used in IBM iSeries and pSeries
- Full 64-bit and 32-bit addressing.
- **http://www.penguinppc64.org**

User Mode Linux (UML)

- Virtual machine, not real hardware
- Can be used for:
  - Kernel development and debugging
  - User space debugging
  - Trying new distributions, kernels, filesystems
  - Commercial hosting (ASP)
- **http://user-mode-linux.sourceforge.net**

Journalling Filesystems

- Operations grouped into **transactions**
- **Transactions** completed **atomically**
- **Log file** records each transaction
- On system failure, power outage, etc., only the most recent transactions need checking
- Result: **fsck** runs very fast (seconds)

Journalling Filesystems: Built into the Kernel

- **EXT3** (in 2.4) Extension of **EXT2,** same on-disk layout, easiest migration path **http://e2fsprogs.sourceforge.net/ext2.html**
- **ReiserFS** (in 2.4) **http://www.namesys.com**
- **JFS (**IBM, AIX) **http://oss.software.ibm.com /developerworks/opensource/jfs**
- **XFS (**SGI, IRIX) **http://oss.sgi.com/projects/xfs**

Journalling Filesystems: Enhancements

- Large files allocated using **extents:**
  (file offset, starting block, length)
- Better handling of large directories
- Dynamic inode allocation
- 64-bit
- Limit internal fragmentation from files smaller than a block

Journalling Filesystems: Features

Feature	Ext3	Reiser	JFS	XFS
Largest Block Size (IA32)	4 KB	4 KB	4 KB	4 KB
Largest Filesystem	16384 GB	17592 GB	18000 PB	32 PB
Largest File Size	2048 GB	1 EB	9000 PB	4 PB
Growing Filesystem Size	Patch	Yes	Yes	Yes
Access Control Lists	Patch	No	Yes	WIP
Dynamic disk inode allocation	No	Yes	Yes	Yes
Data Logging	Yes	No	No	No
Log on external device	Yes	Yes	Yes	Yes

data from Steve Best (IBM), Linux Magazine, October 2002

CPU Clock and Voltage Scaling

- Change CPU clock speed on the fly
- Save battery power
- Many platforms including: Intel SpeedStep, Transmeta Crusoe, Intel Xeon, AMD PowerNow K6, ARM, AMD Elan, VIA Cyrix Longhaul
- Read and change from **`/proc/cpufreq`**
- **http://www.brodo.de/cpufreq**

Setting Processor Affinity

- **Bind** (or **pin**) a process to specific CPU
- Can set by writing a mask to **`/proc/[pid]/affinity`**
- Or use new system calls:

```
sched_setaffinity(pid,len,&mask);
sched_getaffinity(pid,len,&mask);
```

- **http://www.kernel.org/pub/linux/kernel /people/rml/cpu-affinity**

Reverse Mapping Virtual Memory System (**rmap**)

- One way mapping:
  - given a virtual address, find page table entry (**PTE**) pointing to page of physical RAM; if not present, generate page fault
  - No inverse operation: find **PTE**'s corresponding to a physical page. This makes freeing memory inefficient. All page tables must be scanned to make sure a page is not referenced.

Reverse Mapping Virtual Memory System (**rmap**)

- Reverse mapping:
  - Create a data structure for each physical page that lists **PTE**'s pointing to it, referenced through the `page` structure
  - Freeing pages much faster, more overhead
  - 2.4 kernel version yanked in 2.4.12; built in piece by piece in 2.5
- (Rik van Riel) **http://surreil.com/patches/**

Rewrite of Block I/O Layer (BIO)

- Complete rewrite
- More tunable at low and high levels
- Better performance possible
- Requires new API for block drivers

Rewrite of BIO Layer: Low Level Tuning

- Per-queue parameters instead of global
  - max request size, sector size, max sectors, etc., can take more optimal values
- High memory I/O support
  - If possible avoid **bounce buffers**
- I/O scheduler modularization
  - Can write a method for a queue, or choose from a list of generic ones

Rewrite of BIO Layer: High Level Tuning

- I/O Barriers
  - Can request strict ordering of requests
  - uses **`BIO_BARRIER`** flag
- Request priority, latency
  - Specify low, med, high priority for request
  - Place latency limits on requests

Rewrite of BIO Layer:

- **Bypass Mode** permits direct low level access without use of `ioctl's`
- Larger I/O requests can be sent without fragmenting and then recombining
- `io_request_lock` replaced by finer-grained per-queue lock
- 64-bit sector numbers

IDE Layer Update

- Early 2.5 kernels had a complete rewrite
- Led to stability problems, filesystem corruption
- Led to technical and political upheaval
- 2.4 kernel IDE layer restored, but many incremental improvements made
- Lesson: build new and old at same time

New Features: Module Implementation and Utilities

- Module loading, unloading, etc, has been moved (mostly) back into the kernel
- New, thinner, set of utilities (**insmod, rmmod, depmod, modprobe**)
  - Older versions still remain (e.g., **insmod.old**)
- Came **after** feature freeze
- Work of Rusty Russel: download from **ftp://ftp.kernel.org/pub/linux/kernel/people/rusty**

New Features: System Call Mechanism

- System calls on P4 are slower than for earlier CPU's
- Old mechanism:
  - use **`int 0x80`** instruction to generate exception
- New mechanism:
  - use **`sysenter`**, is much faster

New Features: System Call Mechanism

- Not all **x86** CPU's support **`sysenter`**
- Monkeys with some CPU registers
- Requires support in **C** library (**glibc**)
- Hard to use with more than 5 arguments
- Speedup:
  - Pentium 4: factor of 2
  - Pentium 3: factor of 1.2

Removed Features: Export of `sys_call_table`

- System calls are done by jumping to `sys_call_table[n]`
- The table is **exported** to modules
- Thus it is possible to substitute for standard system calls, or introduce new ones.

Removed Features: Export of `sys_call_table`

- Example:

Module loading:

```
save_syscall = sys_call_table[n];
sys_call_table[n] = my_syscall;
```

Module unloading:

```
sys_call_table[n] = save_syscall;
```

Removed Features: Export of `sys_call_table`

- Technical problems (solvable):
  - Unsafe, prone to race conditions
  - Non-portable, different on every platform
  - Security, possibly
- Licensing problem s (more basic)
  - Modules should not change heart of kernel
  - Can still put in “stubs” for new calls

Removed Features: End of **Task Queues**

- Task queues were used for deferred processing, *e.g.*, interrupt bottom halves
- Have been replaced by **tasklets**, which are better on SMP systems, cleaner
- Have been deprecated throughout 2.4, so not many instances were left.
- Quick fix was to use **`schedule_task()`**, run under **`keventd`** context; gone now

Upcoming Linux Programming Classes at Axian

- RHD143 (Linux Programming Essentials)
  - **April 7-11**
- RHD221 (Linux Device Drivers)
  - **April 14-18**
- RHD236 (Linux Kernel Internals)
  - **April 21-25**
- Linux Kernel Network Programming ***(New)***

Upcoming Linux Programming Classes at OGI

- Linux Kernel Internals
  - **March 17–23, June 16-20**
- Linux Device Drivers
  - **April 14–18, July 14-18**
- Linux Kernel Network Programming ***(New)***
  - **February 10–14, May 12–16, August 11-15**