

# Guías de estilo, políticas, herramientas y el kernel 2.6



**Juan Quintela**  
quintela@trasno.org

□ Introducción

---

□ ¿Qué es el kernel Linux?

- Herramientas principalmente utilizadas
- Primeros contactos/impresiones con el kernel
- Cambios principales en kernel 2.6
- Otros cambios
- Algunos temas de interés.
- Conclusiones
- Referencias

# Herramientas

---

- Compilador (gcc)
- Depurador (gdb y kdb)
- diff y patch
- CVS y Bitkeeper
- LXR
- Martin Dovera: gengraph (cdep, genfull, ..)
- bugzilla
- patchset
- VM Regress
- ksymoops
- rtc-debug (Puntos criticos)

# Primeros comienzos con el kernel

---

- Proyecto kerneljanitors
- Proyecto kernelnewbies
- Listas de correo (lkml, kmm, kcrypto..)
- Subir código
- Guías de estilo
- Grupos de confianza
- CodingStyle y Submitting Patches.

## Principales cambios en kernel 2.6.x

---

- Cambio del algoritmo de planificación  $O(1)$
- Preemptivo y baja latencia
- Soporte ALSA
- Amplio soporte ficheros
- Nuevas arquitecturas
- Linux Console
- ACPI
- Soporte Bluetooth
- Soporte HotPlugging (CPU, PCMCIA, FireWire, PCI, SCSI..)
- VM rmap
- Hilos nativos
- SCTP

## Otros cambios:

---

- Soporte NUMA
- Añadido ACLs/HTree a ext2/3
- UML
- Oprofile
- EVMS
- IPsec
- CryptoApi
- kconfig y gconfig
- Cambio en el cargador de módulos (LKM's)
- ide-tools

## Planificación:

---

- Procesos e hilos.
- Estados de procesos.
- Colas de espera.
- Relación de parentesco.
- Llamada al planificador.
- Cambio de contexto, estado y proceso.

## Planificador $O(1)$

---

- Mejora escalabilidad.
- No necesario recorrer runqueue.
- Uso de dos arrays.
- Mejora de prioridades
- Verificación de prioridades dinámicas.
- Dormir procesos interactivos (sleep\_avg: dinámica menos estática muy pequeña.)
- Soporte Hiperthreading.



# □ Preempt

---

- Junto baja latencia grandes cambios en respuestas.
- Latencia de planificación: despertar
- Latencia de interrupciones
- Soluciones:
  - Uso planificador frecuentemente
  - Expulsión y schedule en tiempo real.
  - Cambio en spinlocks y API para preempt (\*\_enable&\_disable&schedule)
  - rtc-debug: ej dcache

## □ Ejemplo de preempt:

---

```
□ if (TEST_RESCHED_COUNT(100)) {  
□ RESET_RESCHED_COUNT();  
□ if (conditional_schedule_needed()) {  
□ spin_unlock(&dcache_lock);  
□ unconditional_schedule();  
□ goto redo;  
□ }  
□ }
```

## Mejoras en reescritura de:

---

- Gestión de vm respecto a aa del 2.4
- Lectura/escritura en disco
- Framebuffer layer
- API en carga de módulos
- Documentation/sched-design.txt
- SCHED\_BATCH y mejora escalabilidad

## □ Mejora en PID accounting

---

- Cambio de complejidad cuadrática a lineal.
- NPTL (libc y aportaciones de Ingo Molnar) 1:1
- NGPT (M:N por IBM)
- Linuxthreads.

# □ LSM

---

- NSA, SELinux, SGI, Inmunix y Janus.
- Mejora de capabilities y uso anterior de credenciales.
- sys\_security()
- security\_operations
- register\_security y unregister\_security
- inode\_security\_ops
- Rama security/selinux
- selinux\_plug\_init y selinux\_plug\_exit
- execve\_secure, task\_security\_struct superblock\_security\_struct  
inode\_security\_struct, file\_security\_struct, ipc\_security\_struct,  
sock\_security\_struct, skb\_security\_struct
- Seguridad en netfilter.

## □ Otras mejoras:

---

- Sustitución de kbuild (antiguo CML1) por CML2.
- Nueva configuración (kconfig)
- Modificado xconfig para el uso de QT.
- En 2.5.65 se añade el uso de gconfig con soporte para GTK+ 2.x

## □ USB 2.0

---

- Soporte USB 2.0 en 2.6 a 480 Mb/s
- Mayores tasas de transferencia frente a 1.1.
- Conectividad con dispositivos USB 1.1 a 12 Mb/s
- Tecnología denominada OTG (On The GO)
- Cambio de usbdevfs a usbfs (usbtree)
- Cambios notables: tipos de transferencias y alta velocidad.

# ACPI

---

- Advanced Configuration and Power Interface
- Viene a substituir APM
- Especificacion actual 2.0
- PM tools.
- /proc/acpi



# □ NUMA (Non-Uniform Memory Access)

---

- Características de NUMA.
- Nuevo planificador.
- Nueva API.
- SGI MIPS64, Compaq/Alpha Wildfire, Nec Azusa, IBM x440 y NUMAQ.

Mejora notable en escalabilidad.

- NUMA vs SMP.
- NUMA junto SMP (IBM NUMAQ)
- Configuración en anillo SGI MIPS64.
- IBM PPC de arquitectura de varias CPU's en un solo chip.

## □ Estado actual de NUMA

---

- Añadido nuevo planificador O(1): Arcangeli, Franke, Focht y Libenzi.
- Migración Páginas, Procesos y relacionada con I/O.
- Nueva API en espacio de usuario.
- Conjunto de utilidades (añadidas a top, ps, pstree, sar, vmstat..)
- Asociar recursos a nodos y CPU's (memoria, tareas..)

## Pruebas o benchmarks:

---

- Contest
- LTT (Linux Trace Toolkit)
- LSM: micro y macrobenchmarks.
- Scripts que compilan continuamente el kernel
- Envío/recepción constantemente de bloques por el loopback
- Operaciones con números grandes
- Uso de memoria compartida entre procesos.

## Conclusiones:

---

- Uso de 2.6 como version estable.
- Un oops no hace daño a nadie, simplemente es un oops
- Si encuentras un bug, oops, REPORTALO.
- Bugs no importantes a pesar de notables mejoras.
- Grandes mejoras notables ya comentadas.