

Init/main.c

Diseño de Sistemas Operativos
Ingeniería en Informática

Jonay García Santana
Santiago M. Ojeda Ramírez

- Introducción
- Proceso de arranque
- Start_kernel
- Init

Índice

- ¿Cómo arranca el sistema?
- ¿Quién carga el kernel?

Introducción

El arranque está dividido en las siguientes etapas:

- 1.- La CPU se inicializa y ejecuta una instrucción en una localización fija.
- 2.- La instrucción salta a la BIOS.
- 3.- BIOS encuentra un disp. de arranque y accede a su MBR, que apunta al GRUB.
- 4.- BIOS carga y transfiere el control al GRUB.
- 5.- GRUB carga el kernel comprimido.
- 6.- El kernel se descomprime y transfiere el control al descomprimido.

Proceso de arranque

- Introducción
- Proceso de arranque
- Start_kernel
- Init

Índice

- Se inicializa a sí misma.
- Si existen varias, una se establece como primaria.
 - Se encarga de lo que queda de inicialización.
- El kernel, posteriormente, inicializa al resto de las CPUs.

Inicialización de la CPU

- La CPU ejecuta la instrucción 0xFFFFFFFF, que es un salto al comienzo de la BIOS.
- BIOS comienza eligiendo un dispositivo de arranque.
- Lee el primer sector de dicho dispositivo (primeros 512Kb), que se llama Master Boot Record (MBR)
- Carga dicho sector, que contiene la dirección donde se encuentra LILO/GRUB en memoria.

Carga de la BIOS

- ¿Por qué se necesita LILO/GRUB?
- Se carga y encuentra sus datos de configuración en el archivo grub.conf
- A continuación, GRUB carga el kernel en memoria y salta hacia esa dirección.

Gestor de Arranque (LILO/GRUB)

- Los kernel se encuentran guardados como archivos comprimidos.
 - Tienen pequeñas instrucciones básicas sin comprimir, que ejecutan pequeñas rutinas.
 - `Setup()` y `startup_32()` son las rutinas que se encargan de descomprimir el núcleo.

Descompresión del kernel

- Contiene código en ensamblador de 16 bits.
- Se ejecuta en modo real.
- Reconfigura los dispositivos según las necesidades del sistema.
 - Averigua cuanta memoria hay.
 - Configura el teclado
 - Establece IDT y GDT provisionales.
 - ...
 - Pasa a modo protegido y salta a `startup_32()`

Setup()

- Existen realmente dos rutinas startup.
 - Una se encarga de descomprimir el núcleo.
 - Uncompressing Linux...
 - OK, booting the kernel...
 - La segunda establece el entorno para el primer proceso del sistema.
 - Establece los segmentos con sus valores finales.
 - Activa la paginación.
 - Se cargan la IDT y GDT.
 - Salta a `start_kernel()`

Startup_32()

- Introducción
- Proceso de arranque
- Start_kernel
- Init

Índice

- Escrito en C.
- Punto de entrada de inicio al sistema.
- Realiza todo el proceso de arranque de Linux.
- Llama a todas las funciones necesarias.

Start_kernel

- Realiza un cierre global del núcleo. (para que solo una CPU realice la inicialización)
 - 566 `lock_kernel();`
- Muestra el “anuncio” de la versión del núcleo, compilador usado, hora, etc. (`linux_banner`)
 - 571 `printk(linux_banner);`
- Realiza config. específicas de la arquitectura (572)
 - Se ayuda de la información que dio `Setup.S`
 - 572 `setup_arch(&command_line);`

Start_kernel

- Inicia los datos requeridos por el planificador (scheduler)
 - 580 /*
 - 581 * Set up the scheduler prior starting any interrupts (such as the
 - 582 * timer interrupt). Full topology setup happens at smp_init()
 - 583 * time - but meanwhile we still have a functioning scheduler.
 - 584 */
 - 585 sched_init();
- Inicializa las excepciones (traps)
 - 604 trap_init();
- Inicializa las irqs (Int. Hardware)
 - 606 init_IRQ();

Start_kernel

- Inicializa fecha y hora del sistema
 - 608 `init_timers();`
- Inicializa el subsistema softirq (Int. Software)
 - 610 `softirq_init();`
- Inicializa la consola
 - 620 `/*`
 - 621 `* HACK ALERT! This is early. We're enabling the console before`
 - 622 `* we've done PCI setups etc, and console_init() must be aware of`
 - 623 `* this. But we do want output early, in case something goes`
 - 624 `wrong.`
 - 625 `*/`
 - 626 `console_init();`
 - 627 `if (panic_later)`
 - 627 `panic(panic_later, panic_param);`

Start_kernel

- Calcula el valor BogomIPS para esta CPU
 - 662 `calibrate_delay();`
- Llama a `rest_init`
 - 697 `rest_init();`
 - Desbloquea el núcleo.
 - Limpia los restos de inicializaciones anteriores.
 - Lanza un nuevo hilo con la función `init`.

Start_kernel

- Introducción
- Proceso de arranque
- Start_kernel
- Init

Índice

- Última función de inicialización del núcleo.
 - Prepara al sistema para lanzar el proceso init.
- Inicializa los dispositivos.
 - 865 do_basic_setup();

 - 768 static void __init do_basic_setup(void)
 - 769{
 - 770 rcu_init_sched(); /* needed by module_init stage. */
 - 771 init_workqueues();
 - 772 usermodehelper_init();
 - 773 driver_init();
 - 774 init_irq_proc();
 - 775 do_initcalls();
 - 776}

Función INIT

- Libera la memoria que se utilizó para las inicializaciones previas.
 - 797 `free_initmem();`
- Establece los canales de comunicación con la consola.
 - 803 `if (sys_open((const char __user *)`
`"/dev/console", O_RDWR, 0) < 0)`
 - 804 `printk(KERN_WARNING "Warning: unable`
`to open an initial console.\n");`
 - 805
 - 806 `(void) sys_dup(0);`
 - 807 `(void) sys_dup(0);`

Función INIT

- Lanza el proceso INIT

- 817 /*
- 818 * We try each of these until one succeeds.
- 819 *
- 820 * The Bourne shell can be used instead of init if we are
- 821 * trying to recover a really broken machine.
- 822 */
- 823 if (execute_command) {
- 824 run_init_process(execute_command);
- 825 printk(KERN_WARNING "Failed to execute %s. Attempting "
- 826 "defaults...\n", execute_command);
- 827 }
- 828 run_init_process("/sbin/init");
- 829 run_init_process("/etc/init");
- 830 run_init_process("/bin/init");
- 831 run_init_process("/bin/sh");
- 832
- 833 panic("No init found. Try passing init= option to kernel.");

Función INIT

- Primer proceso de usuario que ejecuta el núcleo. Tiene PID=1.
 - Se localiza en /bin/init
- Verifica y monta los sistemas de archivos.
- Inicia servicios de usuario.
- Se ejecuta con un parámetro llamado “runlevel” (de 0 a 6)
 - El fichero /etc/inittab determina el comportamiento para cada nivel de ejecución.
- Pone en marcha todo el resto de procesos necesarios para inicializar el sistema.
 - Proceso de login, servicio de ficheros, demonios de red, etc.

Proceso INIT

- Se localizan en /etc/rc.d
 - Contiene el fichero rc.sysinit.
 - Es el encargado de las primeras inicializaciones del sistema : Hardware, reloj del sistema, swap, discos, red, sistema grafico, etc.
 - Los directorios init.d, rc0.d, ..., rc6.d.
 - Init.d contiene un script por cada servicio que se pueda necesitar en el arranque.

Proceso INIT (Ficheros de configuración)

- El núcleo busca el proceso init en diferentes partes.
- El proceso init ejecuta `/etc/rc.d/rc.sysinit`
- Ejecuta los scripts del nivel de ejecución por defecto, indicado por `/etc/inittab`
- Ejecuta el proceso `/etc/rc.local`
- Una vez hecho esto, la máquina ya está lista para interactuar con el usuario.

Resumen INIT

- Referencias bibliográficas

- [MAX99] - Maxwell, Scott; Linux Core Kernel commentary. Ed. Scottsdale, Arizona: The Coriolis group, 1999.
- Linux Cross Reference.
 - <http://lxr.linux.no/>
- Documentación de años anteriores.

Fin

- Jonay García Santana
 - jjonaygs@gmail.com
- Santiago Ojeda Ramírez
 - sojedaramirez@gmail.com

Init/main.c

Diseño de Sistemas Operativos
Ingeniería en Informática

Jonay García Santana
Santiago M. Ojeda Ramírez