Linux Kernel

The kernel is the essential center of a computer operating system, the core that provides basic services for all other parts of the operating system. A kernel can be contrasted with a shell, the outermost part of an operating system that interacts with user commands.

Typically, a kernel (or any comparable center of an operating system) includes an interrupt handler that handles all requests or completed I/O operations that compete for the kernel's services, a scheduler that determines which programs share the kernel's processing time in what order, and a supervisor that actually gives use of the computer to each process when it is scheduled. A kernel may also include a manager of the operating system's

address spaces in memory or storage, sharing these among all components and other users of the kernel's services. A kernel's services are requested by other parts of the operating system or by application through a specified set of program interfaces sometimes known as system calls.

Because the code that makes up the kernel is needed continuously, it is usually loaded into computer storage in an area that is protected so that it will not be overlaid with other less frequently used parts of the operating system.

The kernel is not to be confused with the Basic Input/Output System (BIOS).

Some kernels have been developed independently for use in any operating system that wants to use it. A well-known example is

the Mach kernel, developed at Carnegie-Mellon University, and currently used in a version of the Linux operating system for Apple's PowerMac computers.

Linux versus Unix

- IEEE Portable Operating System based on Unix (POSIX)
- Specifies API, not internal design choices
- Linux kernel based on Maurice Bach's *The Design of the Unix Operating System* (1986)
- SVR4

Kernel Characteristics

Linux kernel is not as innovative or full-featured as many proporietary UNIX kernesl.

- Monolithic: Large, Complex.
- Most kernels compiled and linked statically. Linux can dynamically load and unload portions of kernel code called modules.
- Kernel threads: Execution context that can be independently scheduled. Kernel operates on common address space.

- Multithreaded Application Support: Lightweight processes, processes on a common address space, physical memory pages, opened files.

- Non-preemptive Kernel: Cannot arbitrarily interleave execution flows.

- Multiprocessor support.

- Filesystems.

- IO Streams.

Good Things

- Linux is free.
- Fully customizable.
- Runs on cheap hardware.
- Fully exploits features of the hardware for power and performance.
- Stable, low failure rate.

- Small and compact.
- Highly compatible with other OS.
- Well-supported.

Basic Unix OS Concepts

- OS: Basic set of programs.
- Kernen: most important.
- Loaded into RAM when System boots.
- Contains critical procedures for system operation.
- Versions: 2.4.20 (Stable release)

- Versions: 2.5.59 (Development release)
- OS must interact with hardware.
- OS must provide execution environment for applications.
- All low-level details of physical organization of the computer are hidden from User.
- All hardware requests through the kernel.
- Non-priviledged mode for Users.

- Priviledged mode for Kernel.

Unix Characteristics

- Multiuser: Concurrently and independently execute two or more applications belonging to two or more users.
  - Concurrently: Active at same time and contending for same system resources.
  - Independently: Don't worry about other applications.
- Unix is a multiuser system that enforces hardware protection on system resources.

- Each User gets private space.
- User IDs, logins, passwords.
- Groups to share resources.

- Processes: Fundamental abstraction.
- A process is an instance of a program in execution, or execution context of a running program.
- Scheduler decides who gets CPU.
- Nonpreemptive: voluntarily give up resources.
- Multiuser must be preemptive.
- Process/Kernel model:

- Each process thinks it is only one on machine.
- Has exculsive access to OS resources.
- System call to kernel resources. Mode from non-priviledge to priviledge. (User to Kernel).
- OS acts to satisfy request.
- Satisfied: return to User mode.

- Kernel Archtecture: Monolithic in that each kernel layer is integrated into the whole and runs in kernel mode on behalf of processes.

- Microkernel: very small set of functions in kernel. Other services on top for other functions.

- Modules achieve many advantages of microkernels without performance hit.

- Module is an object file whose code can be linked to the kernel at runtime.

Overview of Unix Kernels

- Kernel is process manager.
- Kernel Threads: priviledged processes.
- Reentrant Kernels: several processes in kernel mode concurrently.
- Each process has private address space. Private stack, data, and code areas.

Bootstrapping

- Starting up a computer.
- Kernel loaded into memory.
- Turned on, executes boot code stored in ROM.
- That code loads and starts the kernel.
- Kernel probes hardware and starts **init**.

- **init** is PID 1.

- Before login:
  - Filesystems checked and mounted.
  - System Daemons started.
  - Managed by **rc** scripts (run command).

Automatic and Manual Booting

- Automatic: System on its own.
- Manual: Automatic to a point, then, single-user mode.
- Most system processes are not running and no other uers can login.
- For problems.

Steps in Boot Process

- Load and initialize kernel.
  - **/vmlinuz** or **/boot/vmlinuz**
- Device detection and configuration.
- Creation of spontaneous system process.
  - **fork**
  - kernel processes

- Operator intervention.
  - **sulogin**
  - everything by hand
  - **fsck**
- Execution of system startup scripts.
- Multiuser operation.
  - **getty** process listening.
  - Spawn graphical system.

– **init** continues to enable system resources after boot.

Booting PC's

- Master Boot Record (MBR)
- First 512-bytes of disk.
- Contains program to load secondary boot loader.

Boot Loaders: LILO

- Installed either in MBR or boot linux partition.
- Configured and installed with **lilo**
- Based on contents of **/etc/lilo.conf**

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
lba32
default=linux

other=/dev/sda1
optional
label=nt

image=/boot/vmlinuz-2.4.9-13smp
```

```
label=linux.bak
initrd=/boot/initrd-2.4.9-13smp.img
read-only
root=/dev/sda8
append="hdd=ide-scsi"

image=/boot/vmlinuz-2.4.9-13
label=linux-up.bak
initrd=/boot/initrd-2.4.9-13.img
read-only
root=/dev/sda8
append="hdd=ide-scsi"

other=/dev/sda1
optional
```

```
label=dos

image=/boot/vmlinuz-2.4.9-21
label=linux-up.bak1
root=/dev/sda8
append="hdd=ide-scsi"
read-only
initrd=/boot/initrd-2.4.9-21.img

image=/boot/vmlinuz-2.4.9-21smp
label=linux.bak1
root=/dev/sda8
append="hdd=ide-scsi"
read-only
initrd=/boot/initrd-2.4.9-21smp.img
```

```
image=/boot/vmlinuz-2.4.9-31
label=linux-up
root=/dev/sda8
append="hdd=ide-scsi"
read-only
initrd=/boot/initrd-2.4.9-31.img

image=/boot/vmlinuz-2.4.9-31smp
label=linux
root=/dev/sda8
append="hdd=ide-scsi"
read-only
initrd=/boot/initrd-2.4.9-31smp.img
```

LILO Options

boot Name of device that contains boot sector. Specifying a device without number indicates that LILO should be installed in MBR.

map Location of map file created when LILO installed. Contains locations of all disk sectors needed for booting.

install Installs specified file as the new boot sector. Names file containing boot sector on MBR

prompt Displays boot prompt while waiting for user input.

timeout Timeout for keyboard input.

message File for display before boot prompt.

lba32 Removes 1024-cylinder limit.

image Nema of kernel image.

label Name for image at boot prompt.

root Device to mount as root.

append Appends options to parameter line passed to kernel.

initrd File to load into **/dev/initrd**. root is on device whose driver is a module (SCSI), **initrd** facility provides two-stage boot to first, set up temporary root filesystem on RAM disk containing modules you need to add, and then, load modules and mount real root filesystem. RAM disk is special device file **/dev/initrd**.

Boot Loaders: GRUB

- GRand Unified Boot loader.
- Reads configuration file on fly at boot time.
- Names physical devices differently.
- (hd0,0) first hard drive, first partition.
- **/boot/grub/grub.conf**

```
default=0
timeout=10
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
title Red Hat Linux (2.4.7-10)
root (hd0,0)
kernal /boot/vmlinuz-2.4.7-10 ro root=/dev/hda1
```