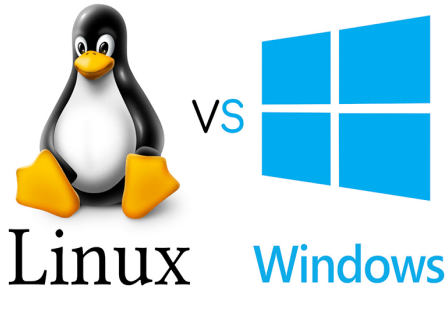
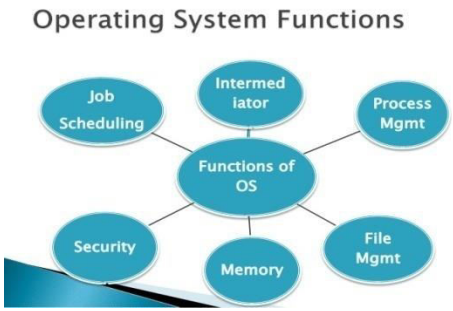
CIT210 OP. SYS. MANAG.

CSUN
COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

Week2Day1 : Unix/Linux Operating Systems, Linux Architecture, Linux Basic Components, kernel, boot loaders, GRUB, LiLO, network, init, daemons, shells, utilities, the X Window





OBSOLETE TYPES OF OPERATING SYSTEMS

- **Single-user, single-process operating systems:**
  - allow only one user at a time on the computer system
  - user can execute/run only one process at a time

  Examples: DOS, Windows 3.1

- **Single-user, multi-process operating systems:**
  - allow a single user to use the computer system
  - user can run multiple processes at the same time

  Example: OS/2

CURRENT OPERATING SYSTEMS

- **Multi-user, multi-process operating systems:**
  - allow multiple users to use the computer system simultaneously
  - Each user can run multiple processes at the same time

Examples: UNIX, Linux, Windows Server (2016/2019, Windows 7/10)

Before Linux

- In 80's, Microsoft's DOS was the dominated OS for PC
  - single-user, single-process system
- Apple MAC is better, but expensive
- UNIX is much better, but much much expensive. Only for minicomputer for commercial applications
- UNIX is a multi-user, multi-process operating system
- UNIX was designed to facilitate programming, text processing and communication

Before Linux

- People was looking for a UNIX based system, which is cheaper and can run on PC
- Both DOS, MAC and UNIX are proprietary, i.e., the source code of their kernel is protected
  - No modification is possible without paying high license fees

UNIX History

- The UNIX operating system was born in the late 1960s. It originally began as a one man project led by Ken Thompson of Bell Labs, and has since grown to become the most widely used operating system.
- In the time since UNIX was first developed, it has gone through many different generations and even mutations.
  - Some differ substantially from the original version, like Berkeley Software Distribution (BSD) or Linux.
  - Others, still contain major portions that are based on the original source code.
- An interesting and rather up-to-date timeline of these variations of UNIX can be found at

http://www.levenez.com/unix/history.html

History of UNIX

- First version written in assembly language
  - single user system, no network capability
- Thompson, Dennis Ritchie, Brian Kernighan
  - rewrote Unix in C: processor/architecture independent
- Unix evolution:
  - Bell Labs, USL, Novell, SCO
  - BSD, FreeBSD, Mach, OS X
  - AIX (IBM), Ultrix, Irix, Solaris (Sun), …
  - Linux: Linus Torvalds

History

- UNIX: 1969 Thompson & Ritchie AT&T Bell Labs.
- BSD: 1978 Berkeley Software Distribution.
- Commercial Vendors: Sun, HP, IBM, SGI, DEC.
- GNU: 1984 Richard Stallman, FSF.
- POSIX: 1986 IEEE Portable Operating System unIX.
- Minix: 1987 Andy Tannenbaum.
- SVR4: 1989 AT&T and Sun.
- Linux: 1991 Linus Torvalds Intel 386 (i386).
- Open Source: GPL.

What is Unix?

- **A portable, multi-tasking and multi-user operating system**
- Portable: runs on many different hardware architectures (Intel x86 and IA-64, Alpha, MIPS, HP PA-RISC, PowerPC, IBM S/390, SPARC, Motorola 680x0, etc.).
- Preemptive multi-tasking: several programs can run at the same time (time slices, interrupts, and task switching).
- Multi-user: many users can share the computer system at the same time.

What is Unix?

- Uses a simple, uniform file model which includes devices and access to other services in a flexible, hierarchical file system.
- Written in a high-level language (“C”) making it easy to read, understand, change and port.
- The command prompt is a simple user process, the Unix shell, which is also a convenient job programming language.
- Includes support for regular expressions which are convenient for complex searching.

GNU project

- Established in 1984 by Richard Stallman, who believes that software should be free from restrictions against copying or modification in order to make better and efficient computer programs
- GNU is a recursive acronym for “GNU's Not Unix”
- Aim at developing a complete Unix-like operating system which is free for copying and modification

Beginning of Linux

- In Sept 1991, Linus Torvalds, a second year student of Computer Science at the University of Helsinki, developed the preliminary kernel of Linux, known as Linux version 0.0.1
- It was put to the Internet and received enormous response from worldwide software developers
- By December came version 0.10. Still Linux was little more than in skeletal form.

Linux Kernel

- Version 0.01 (May 1991) had no networking, ran only on 80386- compatible Intel processors and on PC hardware, had extremely limited device-drive support, and supported only the Minix file system
  - Linux 1.0 (March 1994) included these new features:
  - Support for UNIX's standard TCP/IP networking protocols
  - BSD-compatible socket interface for networking programming
  - Device-driver support for running IP over an Ethernet
  - Enhanced file system
  - Support for a range of SCSI controllers for high-performance disk access
  - Extra hardware support
- Version 1.2 (March 1995) was the final PC-only Linux kernel
  - Kernels with odd version numbers are development kernels,those with even numbers are production kernels

Linux 2.0

- Released in June 1996, 2.0 added two major new capabilities:
  - Support for multiple architectures, including a fully 64-bit native Alpha port
  - Support for multiprocessor architectures
- Other new features included:
  - Improved memory-management code
  - Improved TCP/IP performance
  - Support for internal kernel threads, for handling dependencies between loadable modules, and for automatic loading of modules on demand
  - Standardized configuration interface
- Available for Motorola 68000-series processors, Sun Sparc systems, and for PC and PowerMac systems
- 2.4 and 2.6 increased SMP support, added journaling file system, preemptive kernel, 64-bit memory support
- 3.0 released in 2011, 20th anniversary of Linux, improved virtualization support, new page write-back facility, improved memory management, new Completely Fair Scheduler

Linux System

- Linux uses many tools developed as part of Berkeley's BSD operating system, MIT's X Window System, and the Free Software Foundation's GNU project
- The main system libraries were started by the GNU project, with improvements provided by the Linux community
- Linux networking-administration tools were derived from 4.3BSD code; recent BSD derivatives such as Free BSD have borrowed code from Linux in return
- The Linux system is maintained by a loose network of developers collaborating over the Internet, with a small number of public ftp sites acting as de facto standard repositories
- File System Hierarchy Standard document maintained by the Linux community to ensure compatibility across the various system components
  - Specifies overall layout of a standard Linux file system, determines under which directory names configuration files, libraries, system binaries, and run-time data files should be stored

Design Goals

- Linux is a multi-user, multitasking system with a full set of UNIX-compatible tools..
- Its file system adheres to traditional UNIX semantics, and it fully implements the standard UNIX networking model.
- Three goals
  - Modularity
  - Simplicity
  - Portability
- Numerous standard utilities
  - Eliminate need to write special code
  - Used in combination for specific tasks
- Numerous functions
- IEEE POSIX (Portable Operating System Interface) specifications conformity
  - Programs' portability

Linux Pros and Cons

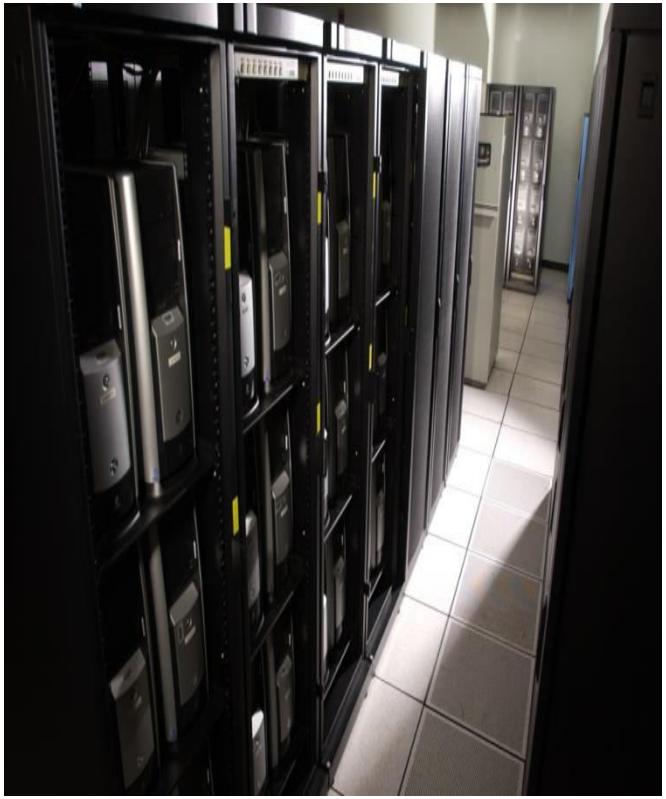
- Advantages over Windows
  - It's almost free to relatively inexpensive
  - Source code is included
  - Bugs are fixed quickly and help is readily available through the vast support in Internet
  - Linux is more stable than Windows
  - Linux is truly multi-user and multi-tasking
    - multiuser: OS that can simultaneously serve a number of users
    - multitasking: OS that can simultaneously execute a number of programs

Linux Pros and Cons (Cont)

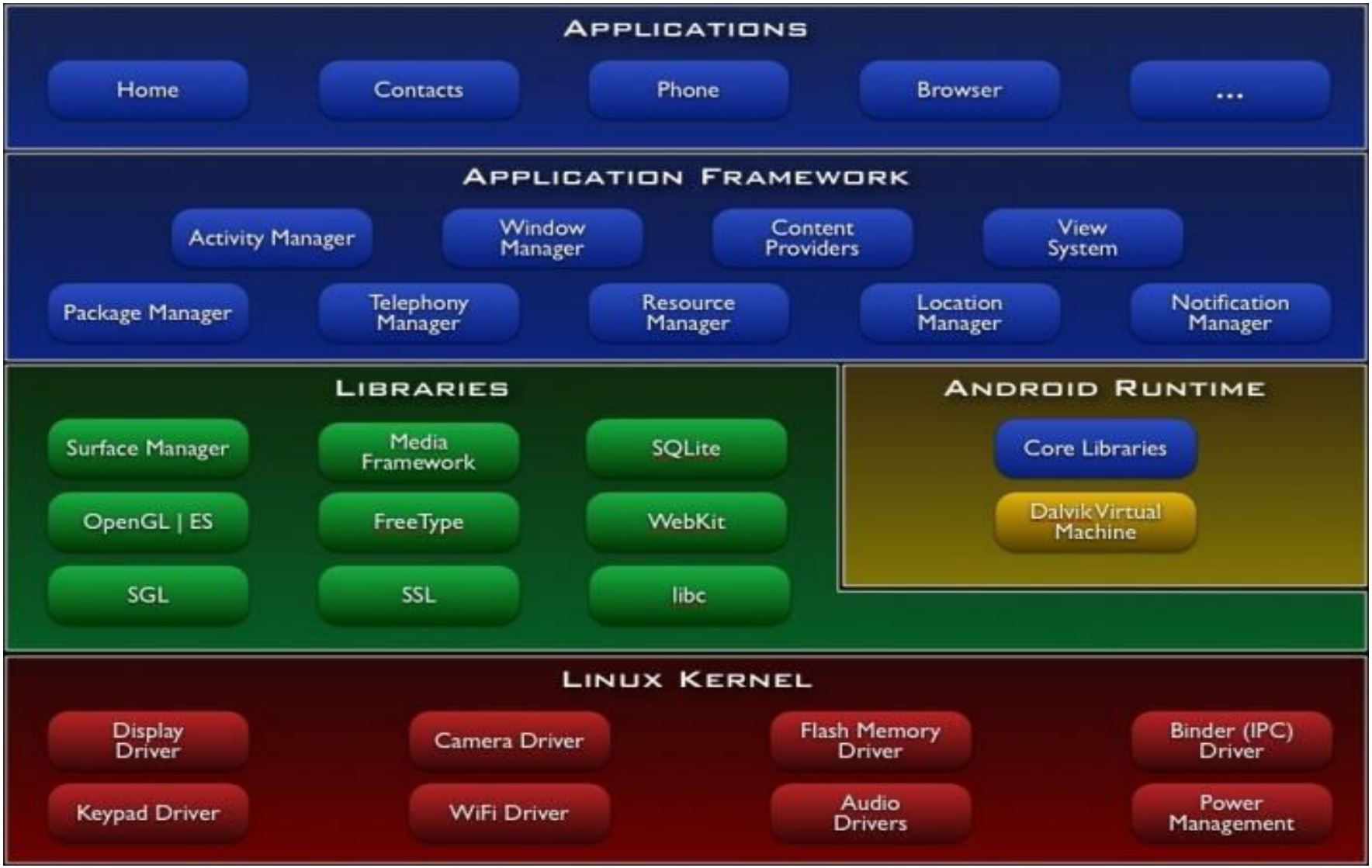
- Disadvantages compared with Windows
  - No one commercial company is responsible for Linux
  - Linux is relatively hard to install, learn and use
- Hence currently, Linux is mainly used in commercial applications, server implementation
- More than 75% current network servers are developed based on Linux or Unix systems
  - Due to the relatively high reliability

Distributions of Linux

- Red Hat Enterprise Linux
- Fedora
- SUSE Linux Enterprise
- openSUSE
- Debian GNU/Linux
- Ubuntu
- Mandriva Linux
- Slackware Linux
- Gentoo



Android Operating System



What is Kali Linux?

- Kali Linux is an advanced Penetration Testing and Security Auditing Linux distribution (distro).
- It was designed to replace the BackTrack Linux distro.
- A Linux distro is a operating system based off the Linux kernel.
- Linux is itself based off the UNIX kernel.
- UNIX > Linux > BackTrack > Kali.

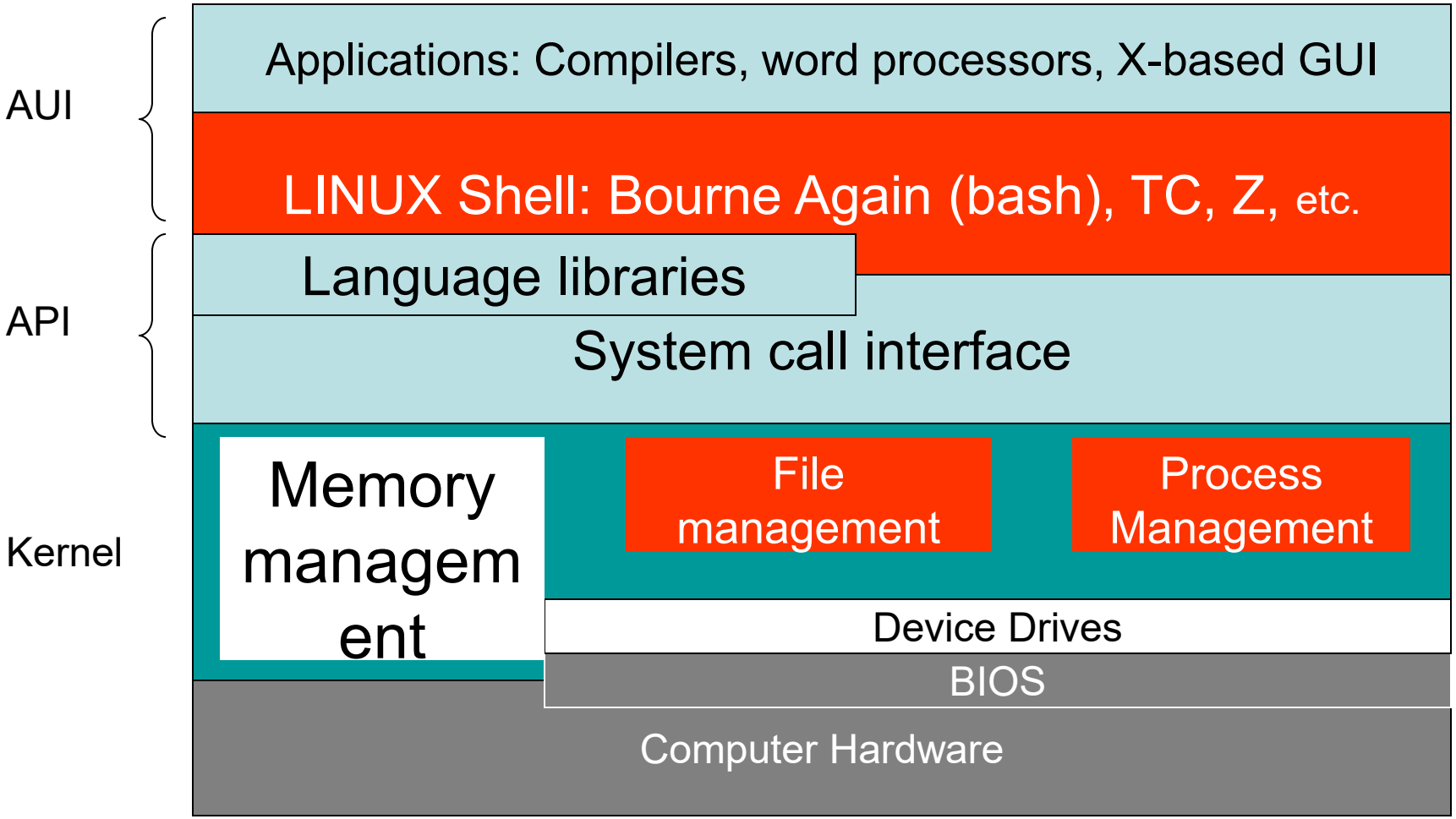
Why use Kali Linux?

- It is FREE!!!!!
- 300+ applications, from password crackers to digital forensics software.
- Vast wireless device support (ARM processors).
- Completely Customizable.
- Multilingual Support.
- Secure Development Environment.
- Open source Git tree.
- Filesystem Hierarchy Standard (FHS) Compliant.
- Gnu Privacy Guard (GPG) secure signed packages and repos.

Linux Operating System

- Basic Components
  - At least six categories of modules are associated with Linux:
    - kernel,
    - network,
    - init
    - daemons,
    - shells and utilities,
    - and the X Window.

Linux Operating System

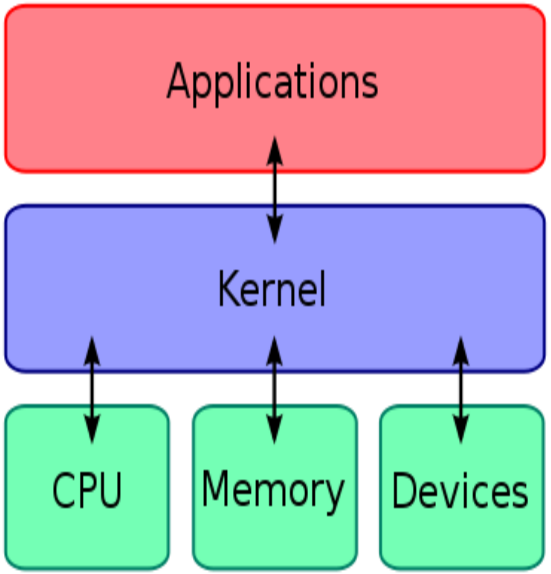


What's a Kernel?

- AKA: executive, system monitor.
- Controls and mediates access to hardware.
- Implements and supports fundamental abstractions:
  - Processes, files, devices etc.
- Schedules / allocates system resources:
  - Memory, CPU, disk, descriptors, etc.
- Enforces security and protection.
- Responds to user requests for service (system calls).
- Etc…etc…

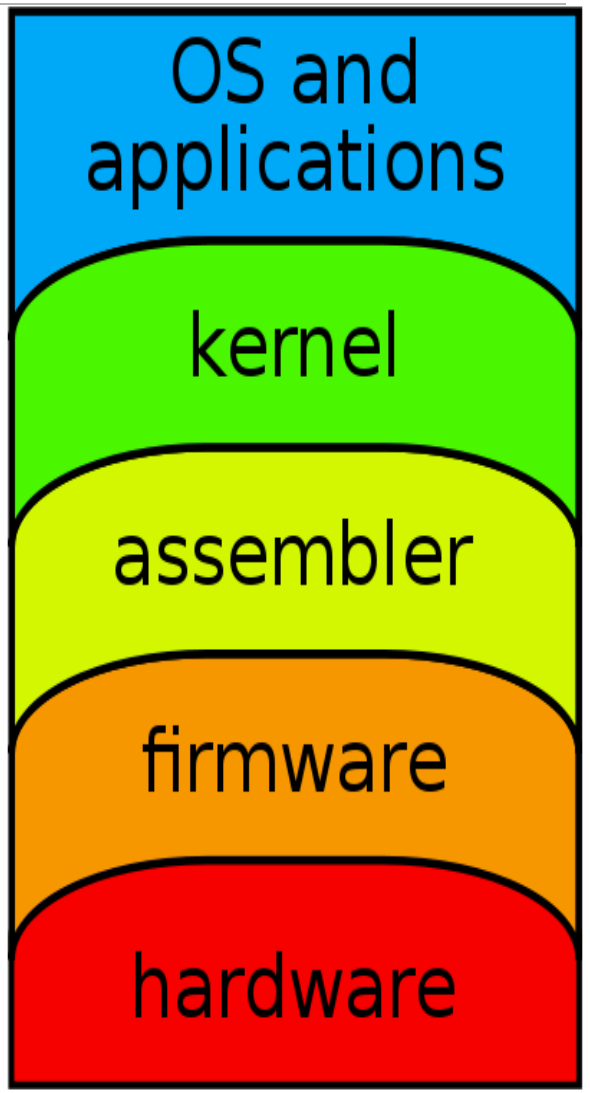
Linux Kernel

- The kernel is the central component of most computer operating systems
  - –it is a bridge between applications and the actual data processing done at the hardware level.
  - –It allows Linux and any software that you install to communicate with computer hardware.
  - –The kernel communicates with your hardware through dedicated device drivers.



Linux Kernel

- Manage the computer's resources
- Allow other programs to run and use these resources
- The resources consist of
  - The Central Processing Unit
  - The computer's memory
  - Any Input/Output (I/O) devices
- Kernels also usually provide methods for synchronization and communication between processes



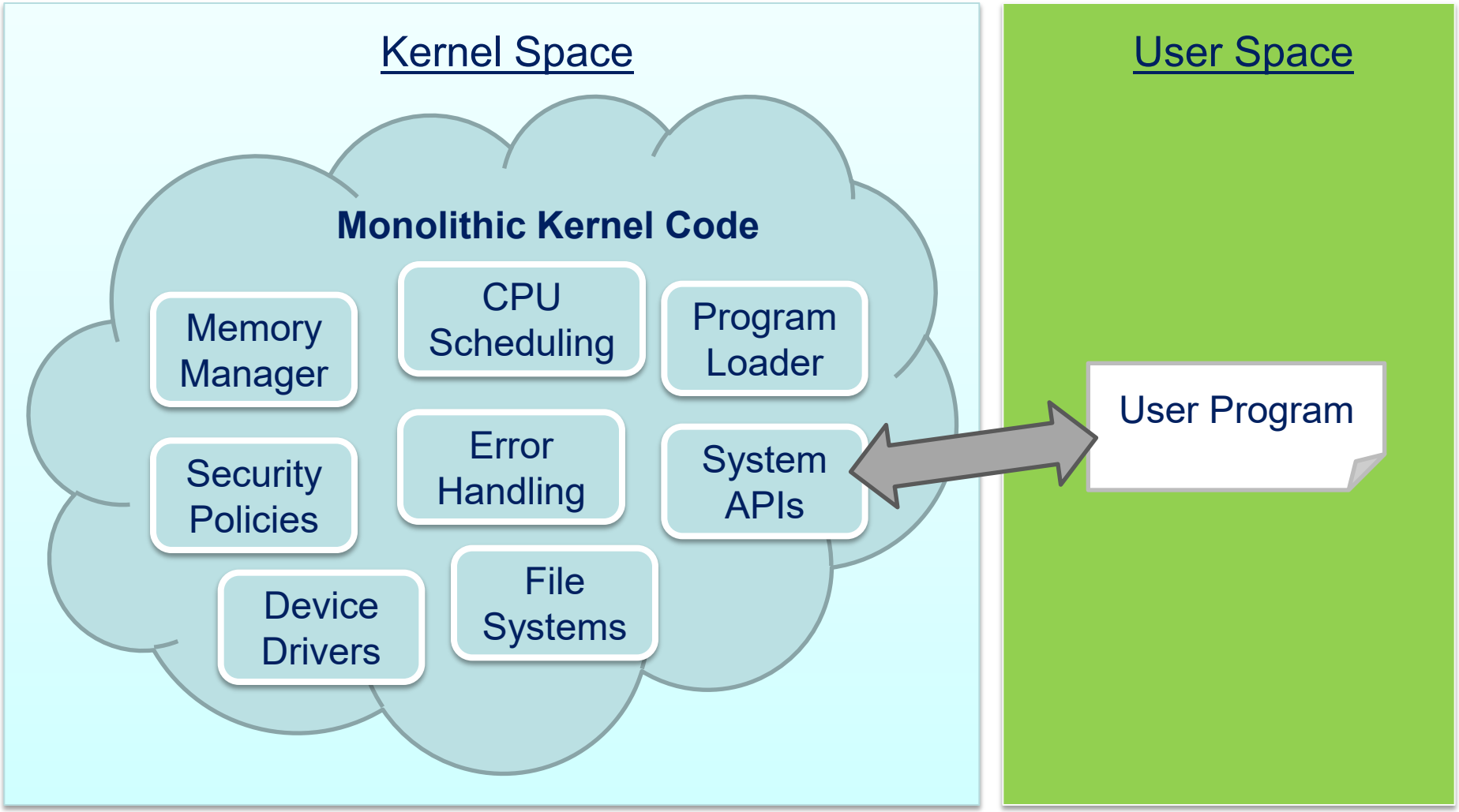
Kernel Design Goals

- Performance: efficiency, speed.
  - Utilize resources to capacity with low overhead.
- Stability: robustness, resilience.
  - Uptime, graceful degradation.
- Capability: features, flexibility, compatibility.
- Security, protection.
  - Protect users from each other & system from bad users.
- Portability.
- Extensibility.

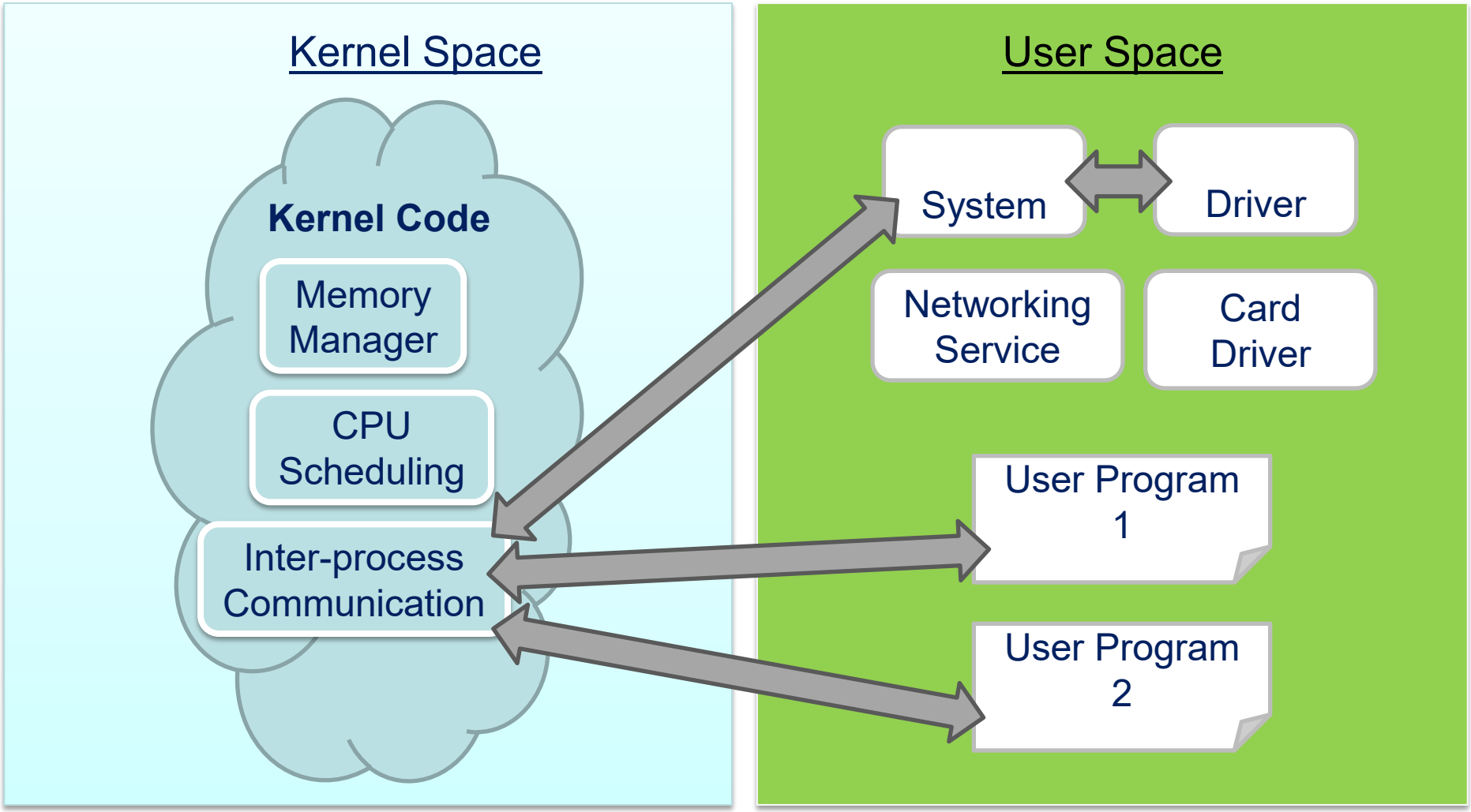
Architecting Kernels

- **Three basic approaches**
  1. **Monolithic kernels**
     - All functionality is compiled together
     - All code runs in privileged kernel-space
  2. **Microkernels**
     - Only **essential** functionality is compiled into the kernel
     - All other functionality runs in unprivileged user space
  3. **Hybrid kernels**
     - Most functionality is compiled into the kernel
     - Some functions are loaded dynamically
     - Typically, all functionality runs in kernel-space

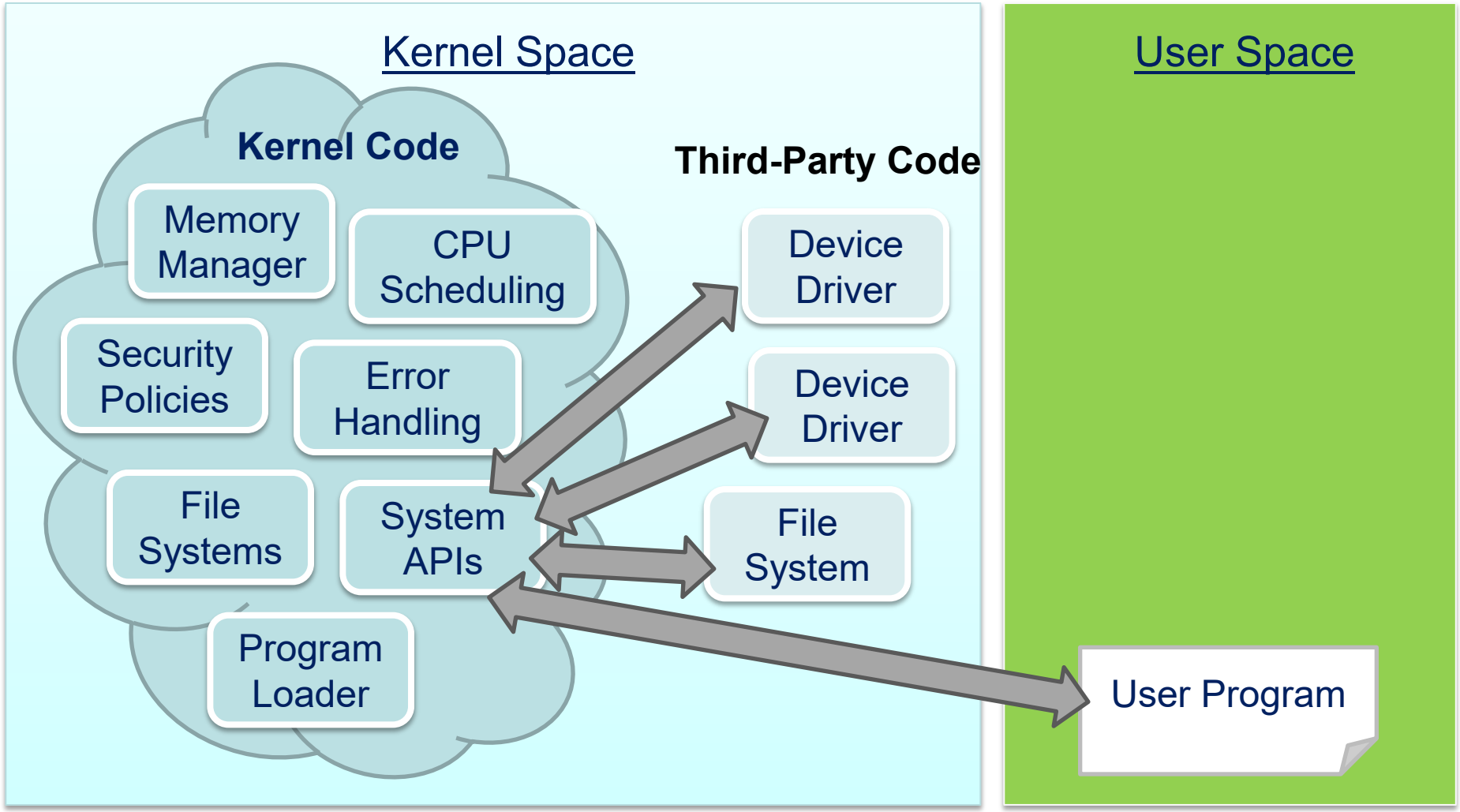
Monolithic Kernel



Microkernel



Hybrid Kernel



Pros/Cons of Monolithic Kernels

- **Advantages**
  - Single code base eases kernel development
  - Robust APIs for application developers
  - No need to find separate device drivers
  - Fast performance due to tight coupling
- **Disadvantages**
  - Large code base, hard to check for correctness
  - Bugs crash the entire kernel (and thus, the machine)

Pros/Cons of Microkernels

- **Advantages**
  - **Small code base, easy to check for correctness**
    - Excellent for high-security systems
  - **Extremely modular and configurable**
    - Choose only the pieces you need for embedded systems
    - Easy to add new functionality (e.g. a new file system)
  - **Services may crash, but the system will remain stable**
- **Disadvantages**
  - **Performance is slower: many context switches**
  - **No stable APIs, more difficult to write applications**

Linux Networking Components

- **Network**
  - Linux computers are most commonly organized in a client/server network.
  - Some computers act as workstations, or clients, for users;
  - others are servers which control resources shared by multiple users on different workstations.
  - In this type of network, clients ask servers for items they need, like files or applications.

Linux Networking Components

- **Socket**
  - A socket is just a logical endpoint for communication. They exist on the transport layer.
  - You can send and receive things on a socket, you can bind and listen to a socket.
  - A socket is specific to a protocol, machine, and port, and is addressed as such in the header of a packet.

Linux Networking Components

Socket

- A **network socket** is a software structure within a network node of a computer network that serves as an endpoint for sending and receiving data across the network.
- The structure and properties of a socket are defined by an application programming interface (API) for the networking architecture.
- Sockets are created only during the lifetime of a process of an application running in the node.
- Socket is externally identified to other hosts by its **socket address**, which is the triad of transport protocol, IP address, and port number.
- The term *socket* is also used for the software endpoint of node-internal inter-process communication (IPC), which often uses the same API as a network socket.

Linux (Init Process)

- **init**
  - Init is started by the kernel during the booting process
  - init (short for initialization) is the program on Unix and Unix-like (Linux)systems that spawns *(function that loads and executes a new child process)* all other processes.
  - It runs as a daemon and typically has PID 1
  - When you boot Linux on your computer, the kernel loads and starts init.
  - The init program then mounts your drives, and starts your terminal programs.
  - When you log in, the terminal program starts your command-line interface shell.
  - The role of init is to create processes from script stored in the file /etc/inittab which is a configuration file which is to be used by initialization system.

Linux (Init vs Systemd)

- **Init vs Systemd**
  - **Init and Systemd are both init daemons but it is better to use the latter since it is commonly used in recent Linux Distros.**
  - **Init uses service whereas Systemd uses systemctl to manage Linux services**
  - **Systemd is the new init framework**, beginning with Fedora and presently embraced in numerous circulations like RedHat, Suse, and Centos.
  - Systemd is a system and service manager for Linux operating systems.
  - **When run as first process on boot (as PID 1), it acts as init system** that brings up and maintains userspace services.
  - The systemctl command **manages both system and service configurations, enabling administrators to manage the OS and control the status of services**. Further, systemctl is useful for troubleshooting and basic performance tuning.

Linux (Runlevels)

- **Runlevels**
  - Red Hat Linux/Fedora runlevels
  - 0 Halt
  - 1 Single-User mode
  - 2 Multi-user mode console logins only (without networking)
  - 3 Multi-User mode, console logins only
  - 4 Not used/User-definable
  - 5 Multi-User mode, with display manager as well as console logins (X11)
  - 6 Reboot

Linux (Process or Daemons)

- **Daemons**
  - Daemon is a computer program that runs in the background, rather than under the direct control of a user;
  - They are usually initiated as background processes.
  - Typically daemons have names that end with the letter "d": for example, syslogd, the daemon that handles the system log, or sshd, which handles incoming SSH connections.
  - In Linux, several dozen daemons can run simultaneously, standing at the ready to start your network, serve web pages, print your files, or connect you to other Linux or Windows computers.

Linux (Process or Daemons)

- **Daemons**
  - In a Unix or linux environment, the parent process of a daemon is often (but not always) the init process (PID=1).
  - Processes usually become daemons by forking ***(when a process forks, it creates a copy of itself)*** a child process and then having their parent process immediately exit, thus causing init to adopt the child process.

Linux (Process or Daemons)

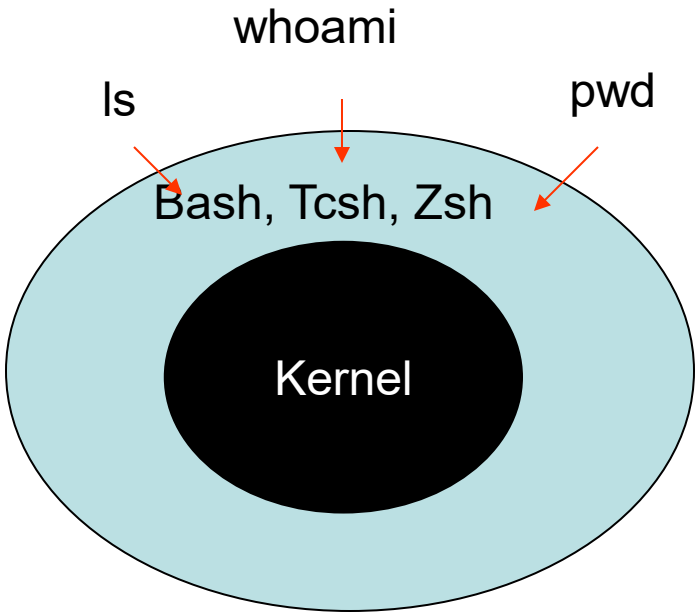
- **Daemons**
  - Typical daemons include:
    - Apache, the most popular web server on the Internet, also known as httpd.
    - Samba (also known as smbd), the network service that allows Linux to talk to Microsoft Windows computers.
    - A printer daemon that manages communication with your printers. The CUPS daemon is cupsd

Linux Shells and Utilities

- **Shells and Utilities**
  - Any Linux program or utility that talks to the kernel is a user-mode program, which consists of shells and utilities.
  - User-mode programs don't communicate directly with your hardware (that's a job for the kernel).
  - *Login* programs associate a user ID with a user's shell and other personalized settings, such as with the X Window and web browsers.
  - *Shell* programs act as Linux command interpreters. The most common Linux shell is known as bash, short for the Bourne Again Shell.
  - *Utilities* are small-scale commands used inside a shell.

Linux Shell

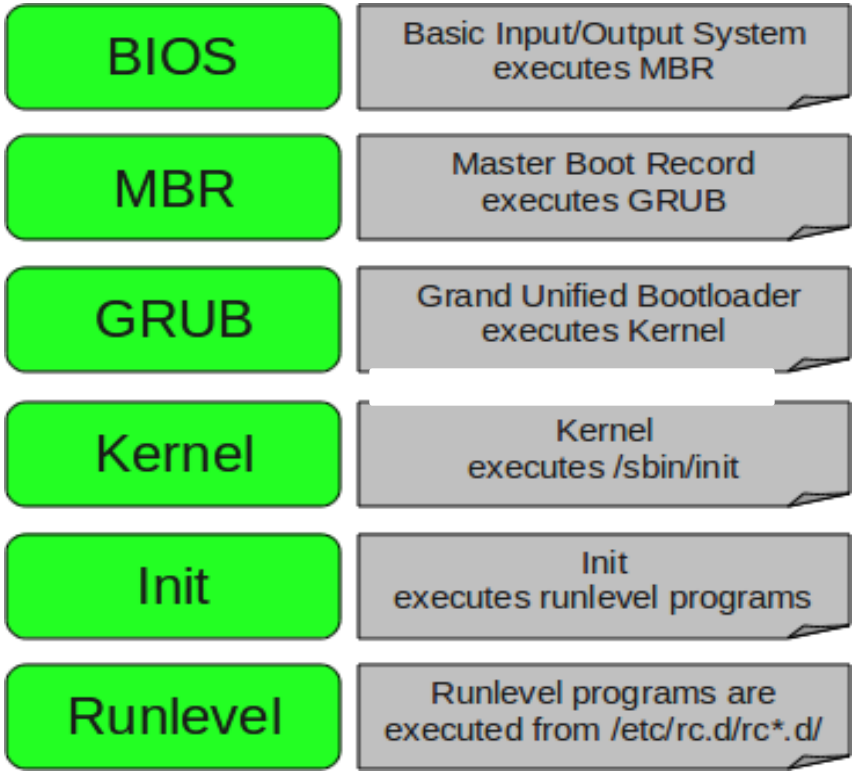
- Shell interprets the command and request service from kernel
- Similar to DOS but DOS has only one set of interface while Linux can select different shell
  - Bourne Again shell (Bash), TC shell (Tcsh), Z shell (Zsh)



- Different shell has similar but different functionality
- Bash is the default for Linux
- Graphical user interface of Linux is in fact an application program work on the shell

Linux Boot Process

- **Explain the Booting process in Linux**



Linux (Boot Process)

- 1.BIOS (UEFI) (Basic Input/Output System)
- 2.MBR(Master Boot Record)
- 3.LILO or GRUB
  - LILO:-LInux LOader
  - GRUB:-GRand Unified Bootloader
- 4.Kernel
- 5.init
- 6.Run Levels

Linux Boot Process

- A typical boot process consists of six phases
  - » Loading and Initialization of kernel
  - » Device detection and configuration
  - » Creation of spontaneous system processes
  - » Operator intervention
  - » Execution of system startup scripts
  - » Multiuser operation

Linux Boot Process

- The system BIOS checks the system and launches the first stage boot loader on the MBR of the Primary hard disk
- The first-stage boot loader loads itself into memory and launches the second-stage boot loader from the / boot partition.
- The kernel is loaded into memory, which in turn loads any necessary modules and mounts the root partition read-only.
- The kernel hands control of the boot process to the init program
- The /sbin/init program loads all services and user-space tools and mounts all partitions listed in /etc/fstab
- The user is presented with login prompt for the freshly booted linux system.

Linux Boot Loader

Linux boot loader broken into two stages

- First stage boot loader is small machine code binary on MBR

(It's job is to locate the second stage boot loader and load the first part into memory)

- ***Second stage boot loader is GRUB and LILO***

•Boot loaders—

- GRUB stands for Grand Unified Bootloader
- LILO stands for Linux Loader

Linux Boot Loader

- GRUB is the default boot loader
- LILO is available for those who require it for their hardware set up
- When the second-stage boot loader is in memory it presents the user with the Fedora Linux initial graphical screen showing different operating systems of kernel it has been configured to boot

Boot Loader Configuration

- GRUB
  - Installed by default, is a very powerful boot loader.
  - GRUB MBR consists of 446 bytes of primary bootloader code and 64 bytes of the partition table.
  - GRUB locates all the operating systems installed and gives a GUI to select the operating system need to be loaded.
  - Once user selects the operating system GRUB will pass control to the kernel of that operating system.
  - GRUB can load a variety of free operating systems, as well as proprietary operating systems with chain-loading (the mechanism for loading unsupported operating systems, such as DOS or Windows, by loading another boot loader).

Boot Loader Configuration

- **LILO**
  - LILO (LInux LOader) is a versatile boot loader for Linux.
  - LILO is a Linux boot loader which is too big to fit into single sector of 512-bytes.
  - It is divided into two parts :an installer and a runtime module.
  - The installer module places the runtime module on MBR. The runtime module has the info about all operating systems installed.
  - When the runtime module is executed it selects the operating system to load and transfers the control to kernel.
  - LILO does not understand filesystems and boot images to be loaded and treats them as raw disk offsets.
  - It does not depend on a specific file system, can boot Linux kernel images from hard disks, and can even boot other operating systems

init

i. The kernel, once it is loaded, finds init in sbin(/sbin/init) and executes it.

ii. Hence the first process which is started in linux is init process.

iii. This init process reads /etc/inittab file and sets the path, starts swapping, checks the file systems, and so on.

iv. It runs all the boot scripts(/etc/rc.d/*,/etc/rc.boot/*)

v. starts the system on specified run level in the file /etc/inittab

Programming Tools and Utilities Available under Linux

- **Text Editors**
  - Xemacs
  - Emacs
  - Pico
  - vi
- **Compilers**
  - C compiler - gcc
  - C++ compiler - g++
  - Java compiler & Java Virtual Machine - javac & java
- **Debuggers**
  - C / C++ debugger - gdb
- **Interpreters**
  - Perl - perl
  - Tcl/Tk - tcl & wish
- **Miscellaneous**
  - Web Browsers - Mozilla, Netscape, Firefox, and Lynx (lynx is text based)
  - Instant Messengers - Gaim
  - Email - Pine

Linux Graphical Desktops

- Graphical display
  - Optional
  - Most users choose to use GUI
- X window system
  - Foundation of graphical display
  - The X Window System (commonly X or X11) is a computer software system and network protocol that provides a graphical user interface (GUI) for networked computers

Understanding the X Window System

- **Project Athena**
  - Graphical environment for UNIX
  - Make UNIX easier to use
  - Eventually called X Window System
  - Released as public domain software in 1985
- **The XFree86 project**
  - Dedicated to creating version of X for Intel-based versions of UNIX

X-Windows

- Desktop environments such as
- Open Windows,
  - A desktop environment for Sun Microsystems workstations which handled X Window System protocols
  - Solaris uses CDE and GNOME 2.0.
- CDE
  - The Common Desktop Environment (CDE) is a desktop environment for Unix,
  - HP's OpenVMS uses CDE as its standard desktop environment

X-Windows

- GNOME,
  - GNOME is part of the GNU Project and can be used with various Unix-like operating systems,
  - Built on top of the Linux kernel
  - and as part of Java Desktop System in Solaris.
- KDE,
  - This allows KDE software based on Qt 4 to also be distributed to Microsoft Windows and Mac OS X.
- Xfce
  - is a free software desktop environment for Unix and other Unix-like platforms, such as Linux, Solaris and BSD. It aims to be fast and lightweight, while still being visually appealing and easy to use.

X-Windows

- X provides the basic framework, or primitives, for building such GUI environments:
  - drawing and moving windows on the screen and interacting with a mouse and/or keyboard.
- X does not mandate the user interface
  - individual client programs handle this.
  - visual styling of X-based environments varies greatly;
  - different programs may present radically different interfaces.
  - X is built as an additional application layer on top of the operating system kernel.

X-Windows

- X-Windows specifically designed to be used over network connections rather than on an integral or attached display device.
- X-Windows features network transparency: the machine where an application program (the *client* application) runs can differ from the user's local machine (the display *server*).
- X-Windows originated at MIT in 1984. The current protocol version, X11, appeared in September 1987.
- The X.Org Foundation leads the X project, with the current reference implementation, X.Org Server, available as free and open source software under the MIT License

X-Windows

- X uses a client-server model: an *X server* communicates with various *client* programs.
- The server accepts requests for graphical output (windows) and sends back user input (from keyboard, mouse, or touchscreen).
- The server may function as:
  - an application displaying to a window of another display system
  - a system program controlling the video output of a PC
  - a dedicated piece of hardware.

Components of the X Window System

- X server
- X client
- Window manager
- Graphical libraries
- Graphical application
  - Provides a comprehensive user interface

Components of the X Window System (continued)

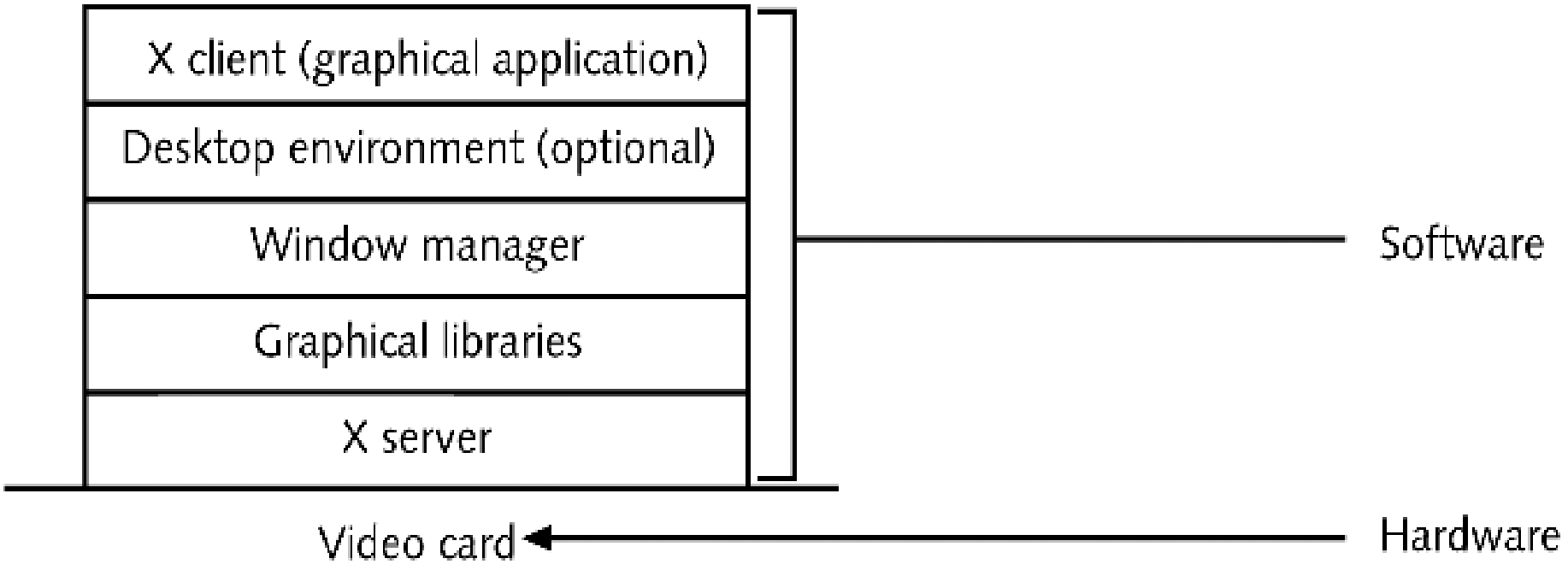
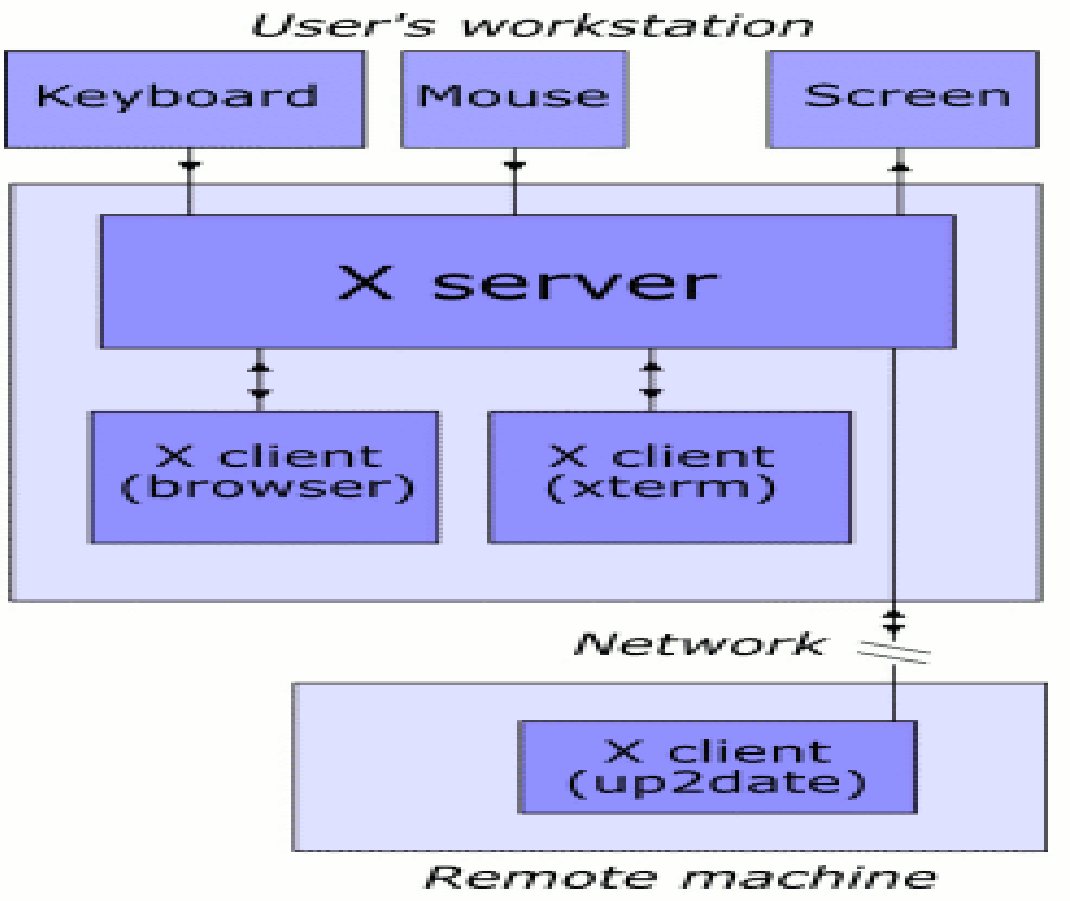


Figure 2-1 Components of the X Window System

X-Windows


User's workstation
Keyboard
Mouse
Screen
X server
X client
(browser)
X client
(xterm)
Network
X client
(up2date)
Remote machine

X-Windows

- The communication protocol between server and client operates network-transparently:
  - the client and server may run on the same machine or on different ones, possibly with different architectures and operating systems.
  - A client and server can even communicate securely over the Internet by tunneling the connection over an encrypted network session.
- An X client itself may emulate an X server by providing display services to other clients.

X-Windows

- The remote X client will then make a connection to the user's local X server, providing display and input to the user.
- Alternatively, the local machine may run a small program that connects to the remote machine and starts the client application.
- Practical examples of remote clients include:
  - administering a remote machine graphically
  - running a computationally intensive simulation on a remote Unix machine and displaying the results on a local Windows desktop machine
  - running graphical software on several machines at once, controlled by a single display, keyboard and mouse.

Components of the X Window System (continued)

- **Window managers available for Linux**
  - **Metacity**
    - is a compositing window manager used by default in the GNOME desktop environment Window Maker and Afterstep
  - **Compiz**
    - is one of the first compositing window managers for the X Window System that uses 3D graphics hardware to create fast compositing desktop effects for window management.
    - The effects, such as a minimization effect and a cube workspace are implemented as loadable plugins.
    - Compiz can substitute for the default Metacity in GNOME or KWin in KDE.


MPlayer
The GIMP
File Xtns Help
Paintbrush
tyros:eelco:~
tyros:eelco:~/Dev/nixpkgs/pkgs
tyros:eelco:~
top - 20:29:37 up 22:32, 3 users, load average: 0.51, 0.16, 0.08
Tasks: 124 total, 3 running, 121 sleeping, 0 stopped, 0 zombie
Cpu(s): 7.3%us, 1.3%sy, 0.0%ni, 90.7%id, 0.0%wa, 0.7%hi, 0.0%si, 0.0%st
Mem: 507428k total, 489308k used, 18120k free, 32412k buffers
Swap: 1020088k total, 41244k used, 978844k free, 128256k cached
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
5869 root 15 0 434m 90m 7172 R 6.0 18.2 17:03.23 X
27101 eelco 15 0 80536 7740 3104 R 1.3 1.5 0:18.22 compiz
26739 eelco 16 0 1604 652 552 S 0.7 3.4 0:09.05 gnome-terminal
1 root 15 0 30932 16m 8784 S 0.0 0.1 0:03.53 init
2 root RT 0 0 0 0 S 0.0 0.0 0:00.06 migration/0
3 root 34 19 0 0 0 S 0.0 0.0 0:00.42 ksoftirqd/0
4 root RT 0 0 0 0 S 0.0 0.0 0:00.00 watchdog/0
5 root 10 -5 0 0 0 S 0.0 0.0 0:00.72 events/0
6 root 18 -5 0 0 0 S 0.0 0.0 0:00.01 khelper
7 root 10 -5 0 0 0 S 0.0 0.0 0:00.00 kthread
35 root 10 -5 0 0 0 S 0.0 0.0 0:00.07 kblockd/0
36 root 10 0 0 0 0 S 0.0 0.0 0:00.04 kacpid
158 root 15 -5 0 0 0 S 0.0 0.0 0:00.03 kseriod
179 root 11 -5 0 0 0 S 0.0 0.0 0:02.78 pdflush
16 -5 0 0 0 S 0.0 0.0 0:05.63 kswapd0
0:00.00 aio/0
0:00.00 kpsmoused

Components of the X Window System (continued)

- **KWin**
  - is a window manager for the X Window System.
  - It is an integral part of the K Desktop Environment (KDE), although it can be used on its own or with other desktop environments.


Welkom bij CodeYard! - Konqueror
CodeYard
Home
Projecten
Bibliotheek
Community
Programmeren
Onderwijs
Documents : zsh
Documents : zsh

Components of the X Window System (continued)

- **Graphical library**
  - **Installed on Linux system like any other application**
  - **Provides tools for other applications**
  - **KDE**
    - Qt
  - **GNOME**
    - Gtk+
    - Default desktop in Red Hat Linux

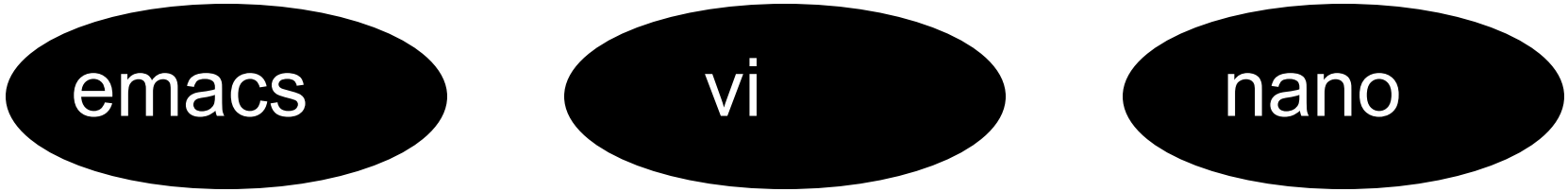
X-Windows

- GNOME,
  - GNOME is part of the GNU Project and can be used with various Unix-like operating systems,
  - Built on top of the Linux kernel
  - and as part of Java Desktop System in Solaris.

X-Windows

- **KDE,**
  - **KDE** is a powerful Open Source graphical desktop environment for Unix workstations. It combines ease of use, contemporary functionality, and outstanding graphical design with the technological superiority of the Unix operating system.
  - This allows KDE software based on Qt 4 to also be distributed to Microsoft Windows and Mac OS X.
- **Xfce**
  - is a free software desktop environment for Unix and other Unix-like platforms, such as Linux, Solaris and BSD. It aims to be fast and lightweight, while still being visually appealing and easy to use.

Editors



- **Dozens (hundreds) of others are available.**
- **Nedit**
- **Bluefish – an HTML editor**

LiveCDs: Easiest Path to Linux

- Live Linux CD are a specific group of distributions that distinguish themselves by

being complete and able to run from a CD or similar media.

  - CD
  - USB
  - DVD

Typically, can also be installed on a HD Or can be run under a virtual machine on Windows

LiveCD Advantages

- Lab set up is passing out the Live CDs.

Our first effort, it took seven minutes from CD distribution to having everyone booted up…

- CDs contain many applications.
  - Open Office
  - Graphics including screen capture
  - Many, many, networking and system tools.
  - No license tracking.
- Students can reproduce and distribute the O/S.
- Trouble shooting is “turn the computer off, turn the computer on”.

Immune from virus, worm, and spyware infection.

Disk Partitioning Setup

- ***Swap***
  - is used as the virtual memory space for an operating system and is automatically set at two times the amount of RAM on your computer.
- ***/boot***
  - will eventually contain the file and commands required for Linux to boot on your computer.
- */*
  - will be the top-level root directory for your filesystem.

Files Used by Bash

- You can easily create these files yourself using your favorite text editor.
- They are:
  - *.bash_profile* : read and the commands in it executed by Bash every time you log in to the system
  - *.bashrc* : read and executed by Bash every time you start a subshell
  - *.bash_logout* : read and executed by Bash every time a login shell exits

/etc/skel

- Provides a way to make sure that all new users on your system begin with the same system settings.
- The /etc/skel directory is used by the /usr/sbin/useradd program.
- It copies all the files, including the hidden files into the new user's home directory.
- When the home directory of a new user is created, it is initialized with files from this folder. Files like, .bash_logout, .bash_profile, .bashrc, .profile are placed in here.