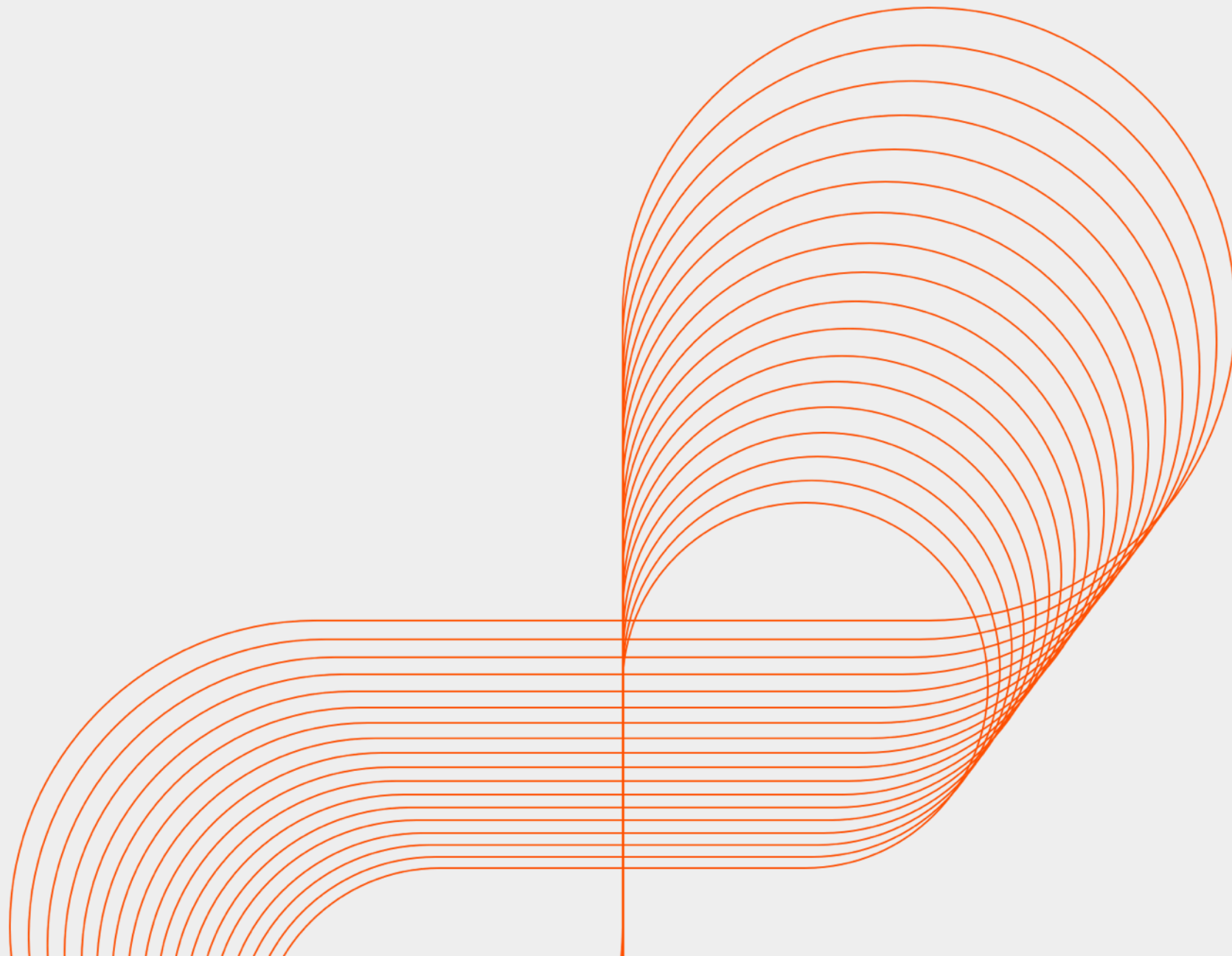
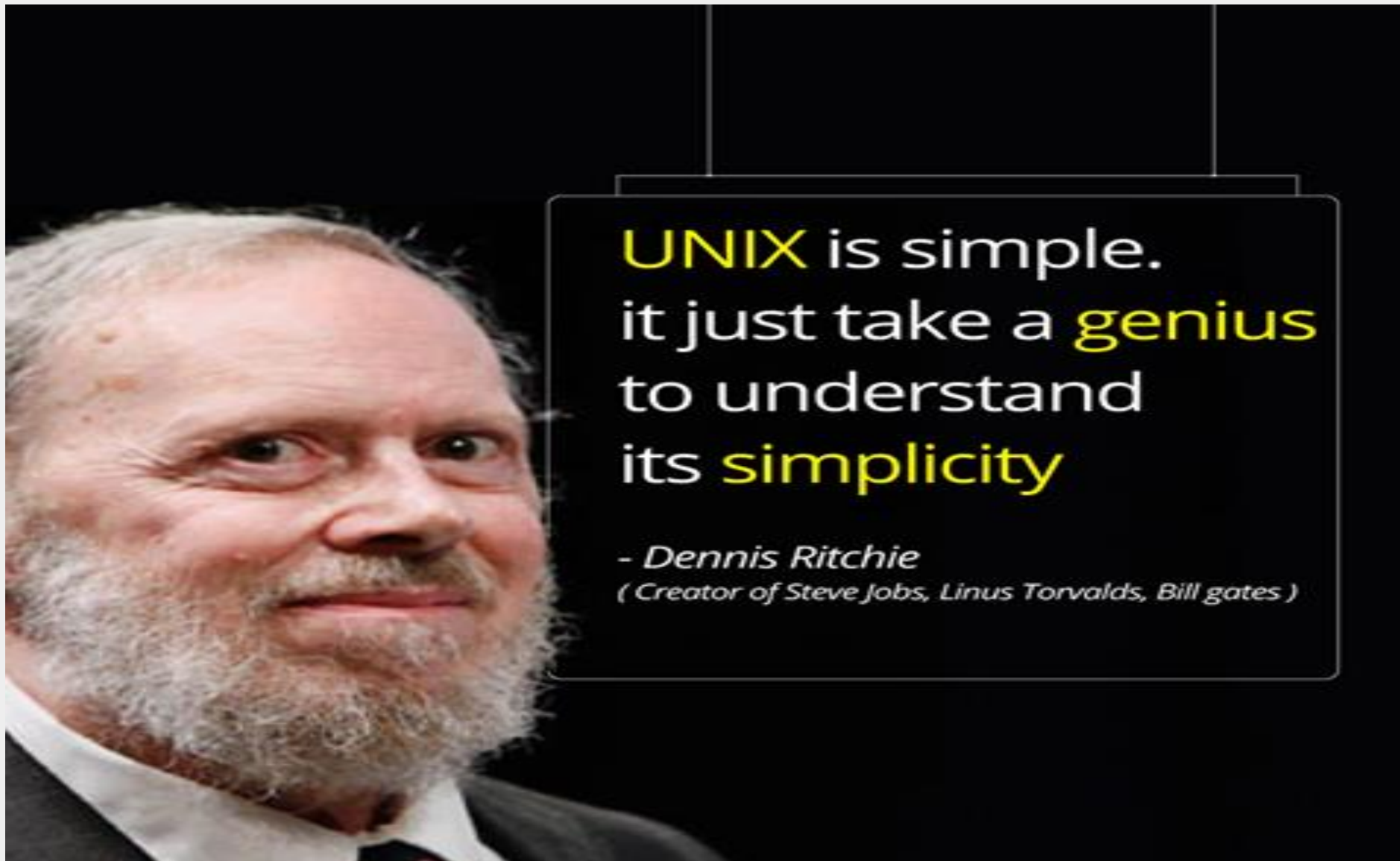




Persistent

# UNIX





# Objectives

- Use the common commands of Linux
- Customize your user environment
- Use the industry-standard vi editor
- Combine commands through pipes
- Schedule jobs

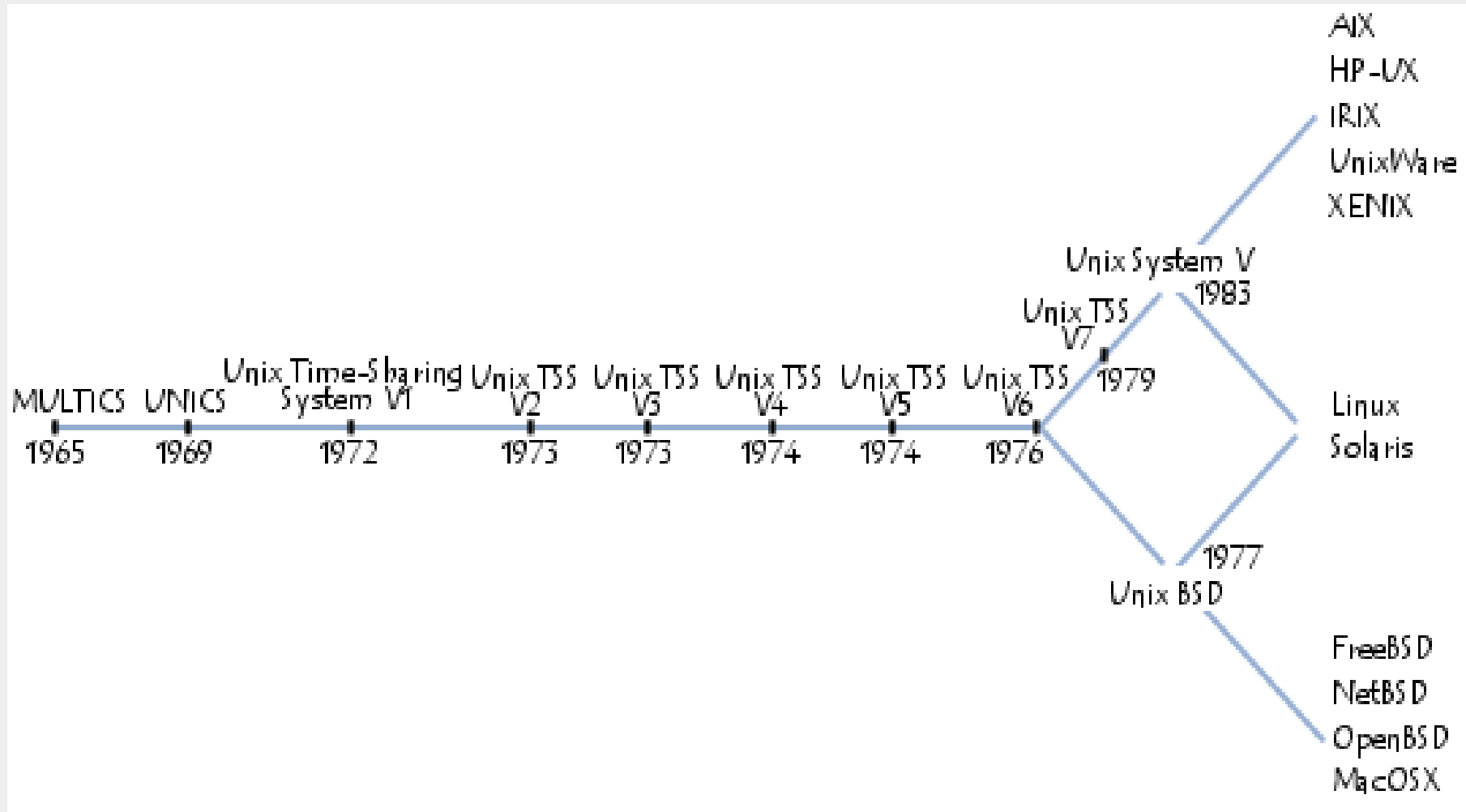
## What is Unix?

- Unix is an operating system developed in 1969 by AT&T employees at Bell Labs
- Unix got its name for UNICS which means Uniplexed Operating & Computing System
- Unix was mainly used by universities which helped it to gain acceptance within students and professionals alike
- Unix is a multi-user, multi-process operating system

## Unix Versions

- Years ago, Unix referred to a single OS, but now it is used to refer to a family of OSes that are offshoots of the original
- There are primarily two base versions of Unix available: AT&T System V and Berkeley Software Distribution(BSD)
- Most of the Unix flavors are built using one of these versions

## Overall appearance of the main Unix type systems



## Unix Variants

- Sun Solaris Unix
- IBM AIX
- Hewlett Packard HP-UX
- Red Hat Enterprise Linux
- Fedora Core
- SUSE Linux
- Debian GNU/Linux
- Mac OS X
- KNOPPIX
- Yellow Dog Linux (for Apple systems)
- SCO OpenServer
- SGI IRIX
- FreeBSD
- OpenBSD
- NetBSD
- OS/390 Unix
- Plan 9

## Unix Variants

- Each variant implements its version of Unix differently, but the core commands and their functionality is same
- Most versions of Unix utilize SVR4(System V) & add the BSD components for interoperability
- Since, there are so many variants of Unix, it creates incompatibilities between them. To standardize the Unix OS, IEEE has created a standard called Portable Operating Systems Interface (POSIX)
- Not all Unix versions follow the POSIX standard completely, but most do adhere to the major principles outlined in the standard



# IBM AIX

- Vendor :
  - Introduced by IBM in 1986.
- Underlying Base version :
  - Hybrid of both Unix System V and BSD
- Latest version :
  - AIX Version 7.1
- GUI For system Management
  - System Management Interface Tool (SMIT)

## IBM AIX – USPs

- Best Virtualization support across all Unix variants.
- Runs only on IBM Power Systems -- easily the most powerful of midrange Unix servers.
- Uses a Journaling Filesystem (JFS2) -- capable of supporting multiple TeraBytes filesystems.
- Has a built-in LVM (Logical Volume Manager) to manage multiple disks and large distributed storage systems.

# Solaris

- Vendor :
  - Originally Sun, now acquired by Oracle.
  - Till Solaris 10 – OpenSource; but after Oracle's acquiring ,
  - Solaris 11 – New development is closed source proprietary
- Underlying Base version :
  - Before 5.0 on BSD, but now Unix System V Release 4
- Latest version :
  - Solaris 11.0
- GUI for system Management :
  - NO GUI. Command line based OS.

## Solaris - USPs

- A large part of it is OpenSource. A separate community exists to keep the things going.
- Runs on Sparc and x86 environments.
- Has the newest file system around -- ZFS, potentially the best all-around Unix file system.

## Hewlett-Packard HP-UX

- Vendor :
  - Hewlett-Packard.
- Underlying Base version :
  - Unix System V Release 4
- Latest version :
  - HP-UX 11i v3
- GUI for system Management :
  - System Administration Manager (SAM).

## HP – UX – USPs

- First of the Unix systems to allow its customers to purchase specific operating environments as per requirements.
- first Unix to offer **access control lists** for file access permissions as an alternative to the standard Unix permissions system.
- Only version that uses a third-party file system – Veritas.
- Performance tuning using kctune is in many ways simpler to what AIX and Solaris offer.

## For more details :

- <http://www.unixguide.net/unixguide.shtml>
- <http://searchdatacenter.techtarget.com/tip/Comparing-Unix-versions-AIX-HP-UX-and-Solaris>

## Current & Future Developments

- Current Unix development focuses on Unix-based system called Linux
- It was developed partially by Linux Torvalds, but through a concerted internet-based effort on the part of many programmers, grew as a successor to both System V and BSD
- It contains features of both branches, as well as features of MIT's X Window System and other OSes
- It is highly portable and can be found on cellphones, PDA's, handheld PC, etc



## Common Things Between Linux & UNIX

- Both share many common applications such as:
  - GUI, file, and windows managers (KDE, Gnome)
  - Shells (ksh, csh, bash)
  - Various office applications such as OpenOffice.org
  - Development tools (perl, php, python, GNU(General Public License) c/c++ compilers)
  - Posix interface

## Difference between Linux and Unix

- Linux is a Unix clone
- Linux Is Just a Kernel
- License and cost
- User-Friendly
- File Systems
- End User Perspective

# Linux Flavours

- Two main groups of Linux distributions (or distros)
  - separated by their software installation and handling.
- .rpm based
  - . rpm stands for Red Hat Package Management
  - RedHat, Fedore Core, Mandrake, Suse etc. Also Centos (freeware, clone of RedHat).
- .deb based
  - .deb is short for Debian
  - Ubuntu, Knoppix, Mepis etc.
- Comparison
  - [http://en.wikipedia.org/wiki/Comparison\\_of\\_Linux\\_distributions](http://en.wikipedia.org/wiki/Comparison_of_Linux_distributions)

## Why Unix/ Linux?

- Unix is often preferred for its reliability, security & speed
- For critical businesses, databases, file servers, web servers, etc run on Unix (Amazon, Wikipedia, NYSE, Google, IBM, CERN, super computers)
- Since, Unix has been around for decades, it's a much more mature operating system. Therefore, bugs & security issues have been discovered and resolved
- Unix systems can survive and continue operating even after a failure of a hardware component
- Open source Linux systems are attractive choices for governments due to their low (or no) cost and extensive support benefit. It is also chosen by defense departments for its security (U.S. Army is the single largest install base for Red Hat Linux)
- Unix is also used in colleges and universities for the same reason
- It can be easily ported on a large variety of embedded devices, such as kiosks, entertainment systems, PDA's, cellphones, etc

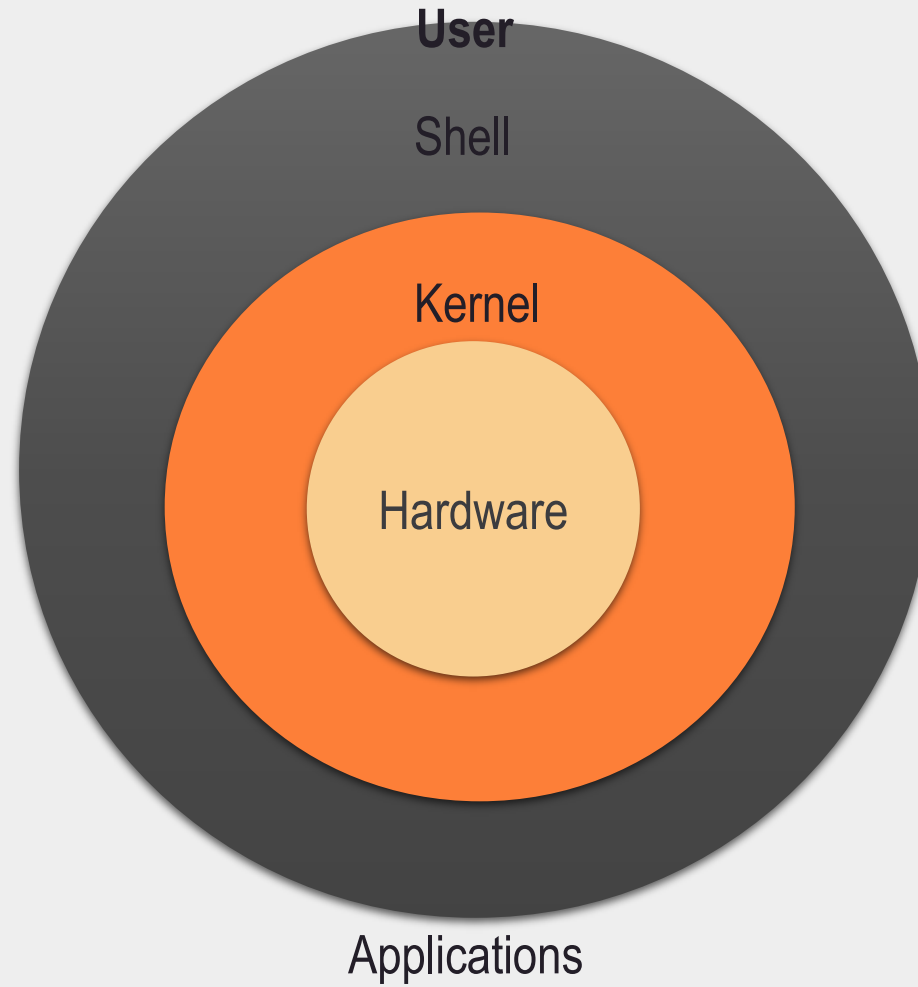
# Unix Philosophy

- During its evolution, Unix has developed a characteristic , consistent approach that is called as Unix philosophy
- This has greatly influenced the structure of the system and the way it works
- The UNIX philosophy is based on the idea that a powerful and complex computer system should still be *simple, general, and extensible*, and that making it so provides important benefits for both users and program developers

# Unix Architecture



# High Level Architecture

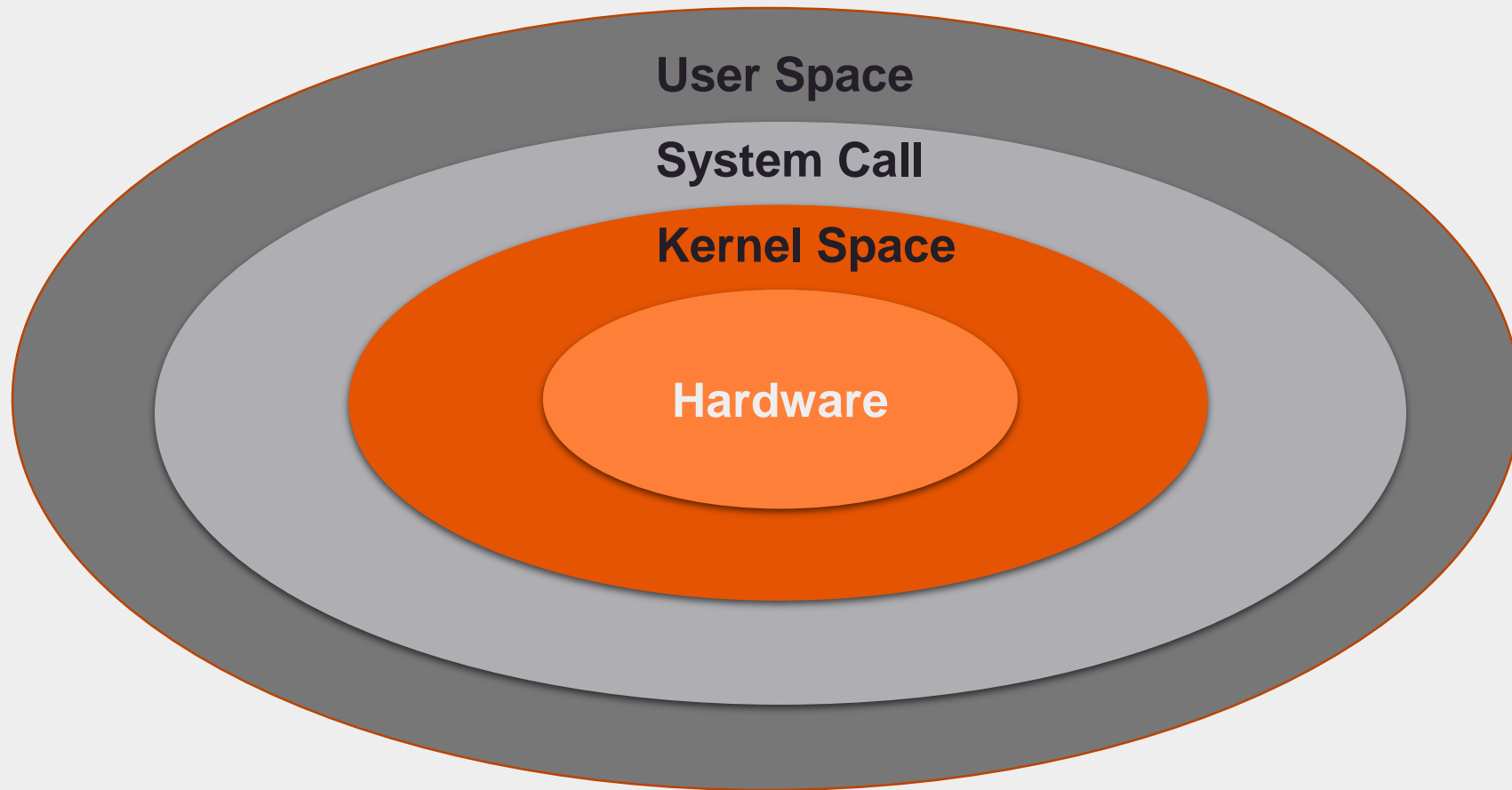


## Major Components

- While viewing the system as a set of layers, the OS is commonly called as the kernel, emphasizing its isolation from user programs
- The kernel consists of various subsystems such as file subsystem, device drivers, process control subsystem, memory manager etc
- The shell is a command line interpreter and acts as an interface between the user and the kernel. It reads the user commands and interprets them as requests to execute programs. Shell is also a programming language
- Other applications run along with the shell and use the services provided by the kernel



# Kernel Architecture Overview



## User Space

- The User Space is the space in memory where user processes run.
- This Space is protected.
  - The system prevents one process from interfering with another process.

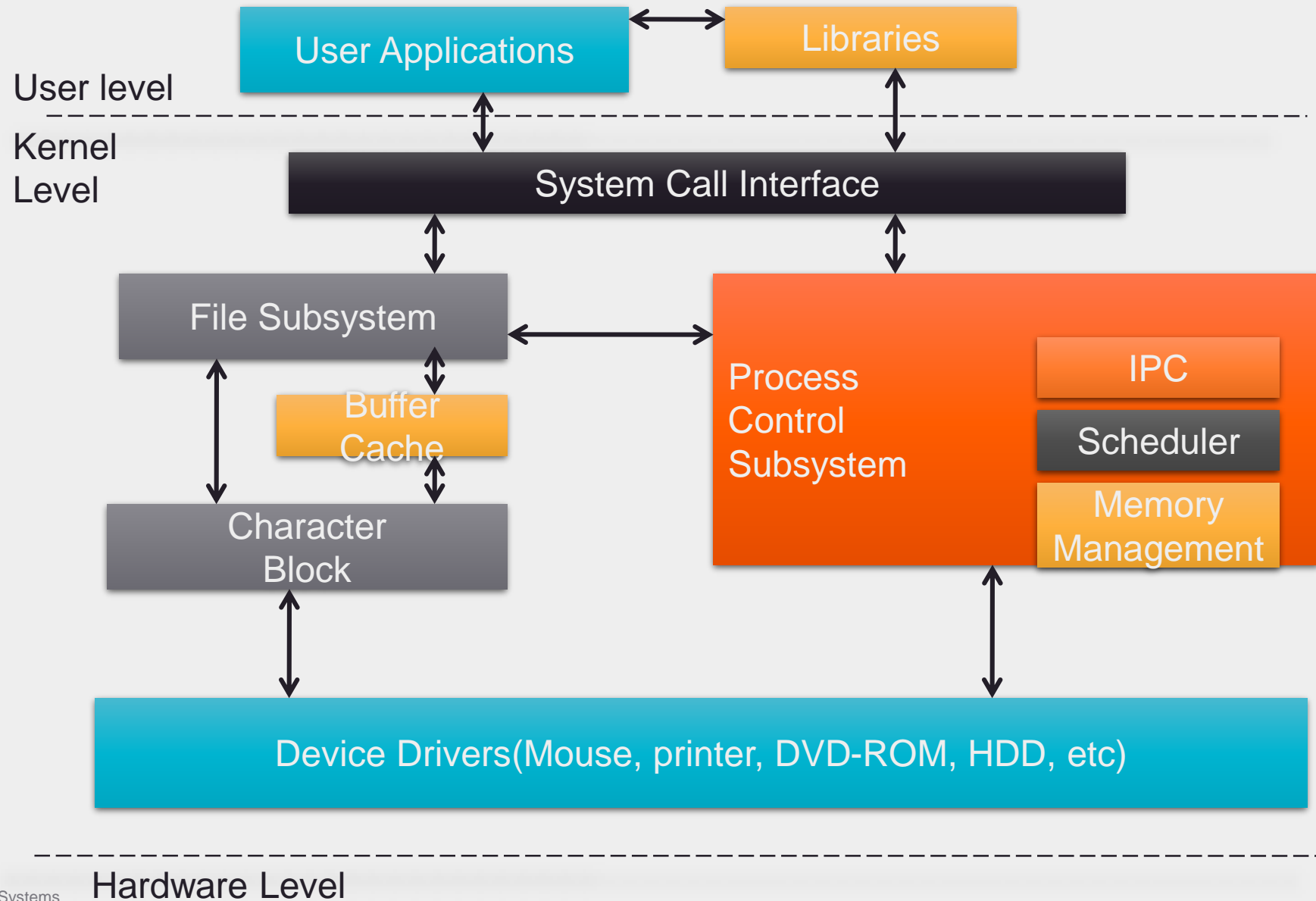
## Kernel Space

- The kernel Space is the space in memory where kernel processes run.
- The user has access to it only through the system call.

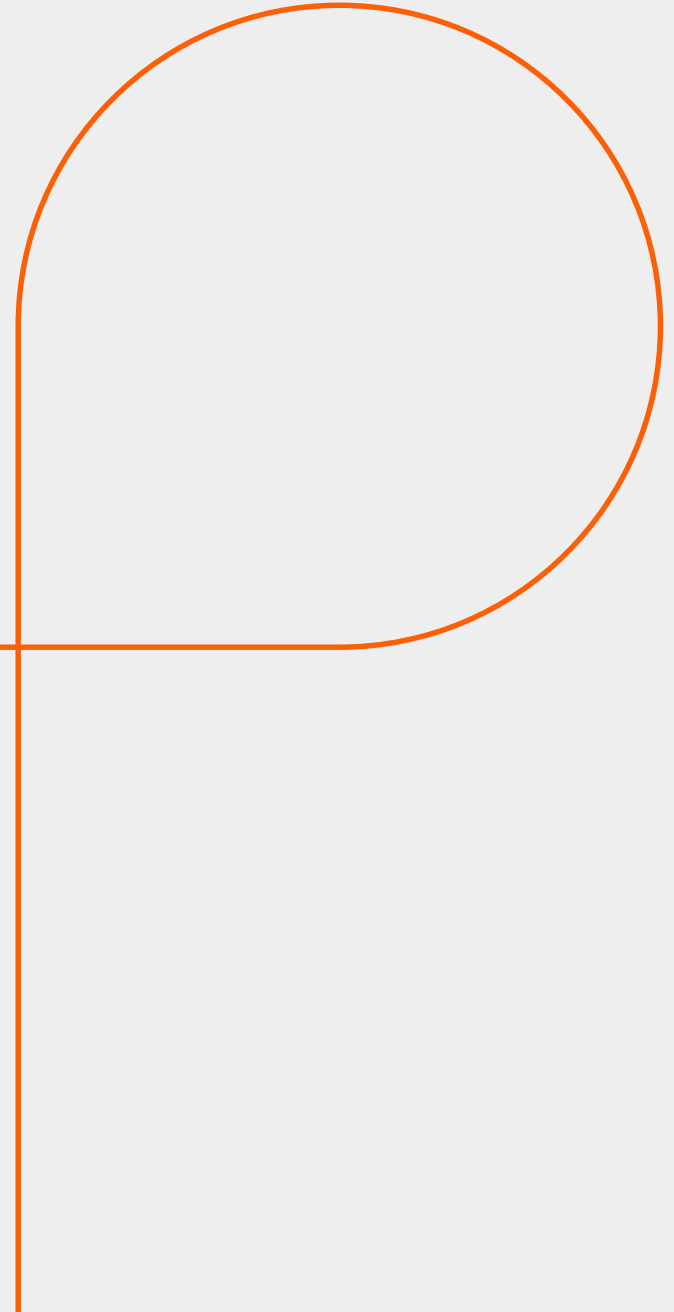
# System Call

- User Space and Kernel Space are in different locations in memory.
- When a System Call is executed, the arguments to the call are passed from User Space to Kernel Space.
- A user process becomes a kernel process when it executes a system call.

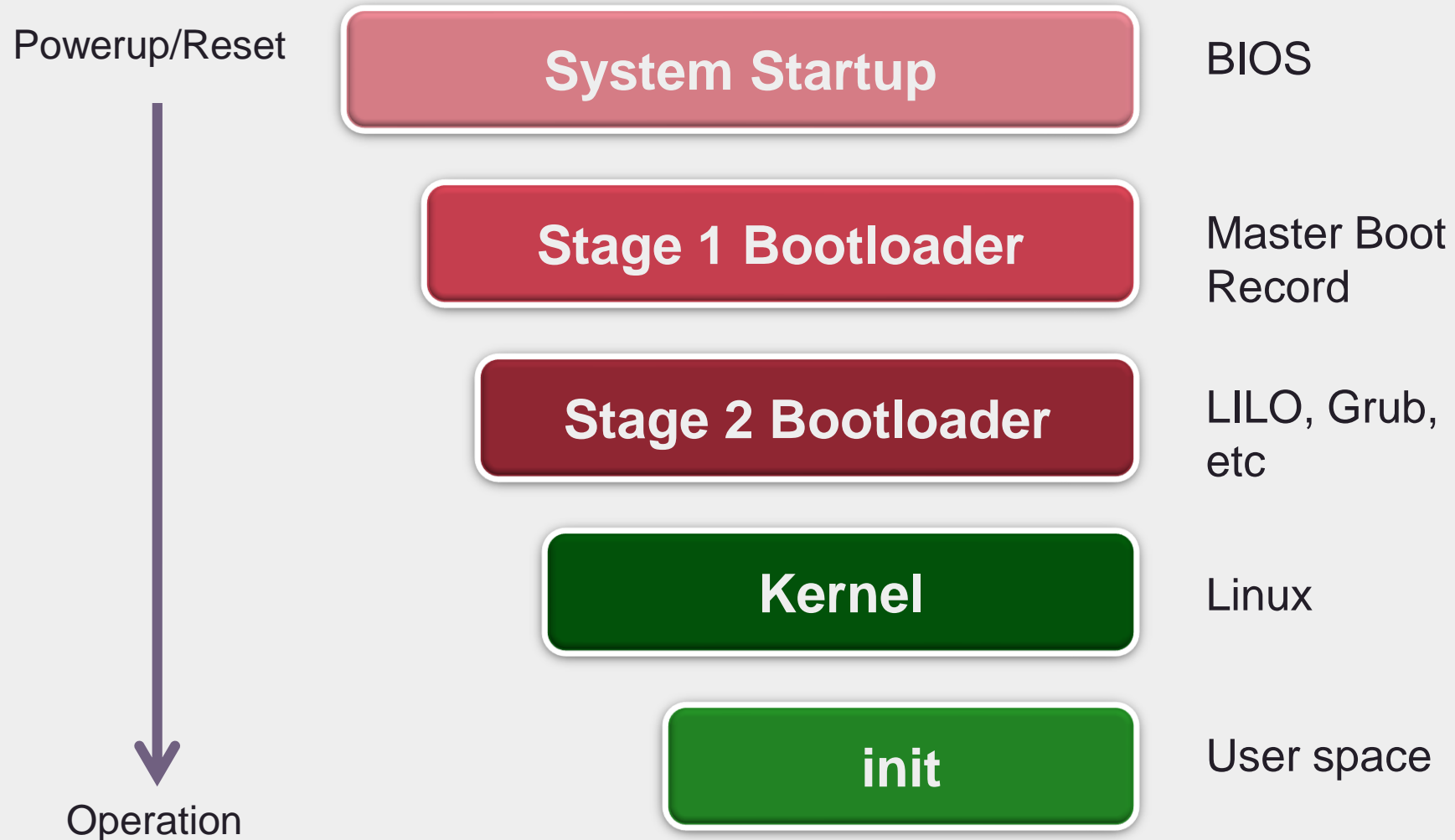
## Kernel Functional Overview



# Boot Process



# Boot Process



## Overview of 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 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 */sbin/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 a login prompt for the freshly booted Linux system



# Kernel Initialization

- Kernel boot time functions
  - Device detection
    - Getting details of devices connected to system
  - Device driver initialization
    - Drivers which are part of kernel
    - Drivers which are compiled as modules and included in initrd image
  - Mounting root file system
  - Starting init process
    - Parent of all processes
    - Control is passed from kernel to init process
    - The source code for the Linux kernel itself is usually stored under `/usr/src/linux`.

## init initialization

- Parent of all processes
  - Use **ps tree** command.
- looks into the initialization specification file (configuration file) - **/etc/inittab**.
- Calls the initialization scripts.
- Completes administrative tasks relative to the system.

## init initialization - Run Levels

- The file **/etc/inittab** identifies how init should interpret different run levels and what scripts and processes should be started in each run level.
- A run level is a grouping of processes (programs in the most basic sense) or daemons (processes that run all the time).
- init defines run level 0-6, S, emergency
- Show current and previous run level
  - /sbin/runlevel

## Run Levels

Run Level	Effect
0	Halt
1,S, emergency	Single user mode (Only root can log in.Used to perform maintainance)
2	Multi-user, without NFS networking
3	Full multi-user mode (includes networking)
4	User definable but duplicate of 3 by default (unused on Linux)
5	Full multi-user mode (with an X-based login screen)
6	Reboot

# Basic Commands

A decorative graphic consisting of a horizontal orange line that extends across the width of the slide. From the right end of this line, a vertical orange line descends, and a large orange circle is drawn, partially overlapping the horizontal line and the vertical line.

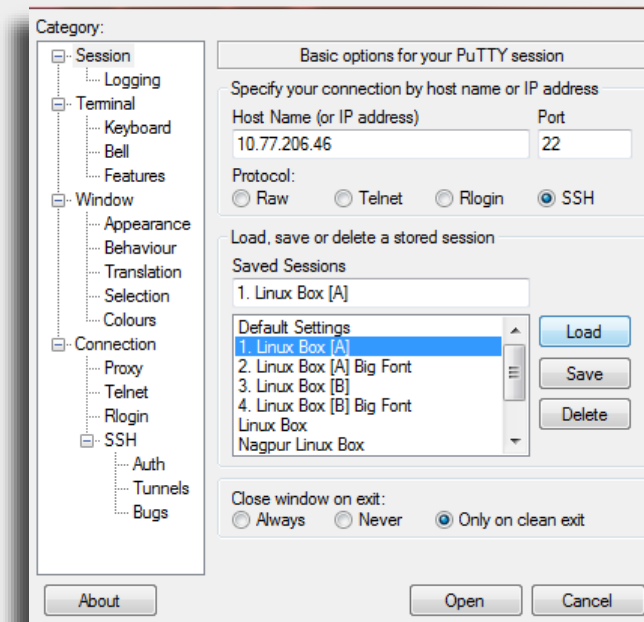
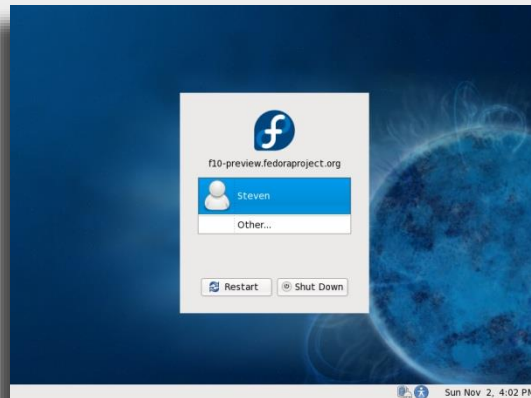
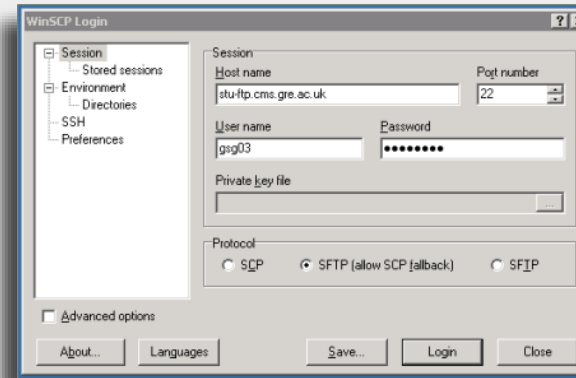
## Unix Interface

- A user can interact with a Unix system through a command-line interface or a graphical interface (GUI)
- Many times a command-line interface is a more efficient and preferred approach, especially by system administrators
- Command-line also gives the power of shell scripting, that can be used to automate several kinds of tasks
- Some machines, such as web servers, are administrated through command line, because command-line interface is not resource intensive unlike graphical interface. Such servers would require all the processing power for fast responses
- The only drawback is that you might have to remember the commands and switches to use Unix through command-line
- We would focus completely on using Unix through command-line

## Logging In

- The initial login screen can be in the form of a GUI or a command line
- Usually, your administrator will provide a login name and a password for you
- You can also login to a remote Unix system using utilities such as ssh, telnet, PuTTY, etc

# Login Screens





## Verifying Kernel version

To Find Release version :

```
[root@localhost /]# cat /etc/redhat-release
```

Red Hat Enterprise Linux Server release 5.1 (Tikanga)

To get kernel version :

```
[root@ps6351 sbin]# cat /proc/version
```

Linux version 2.6.18-53.el5 (brewbuilder@hs20-bc2-3.build.redhat.com) (gcc version 4.1.2 20070626 (Red Hat 4.1.2-14)) #1 SMP Wed Oct 10 16:34:02 EDT 2007

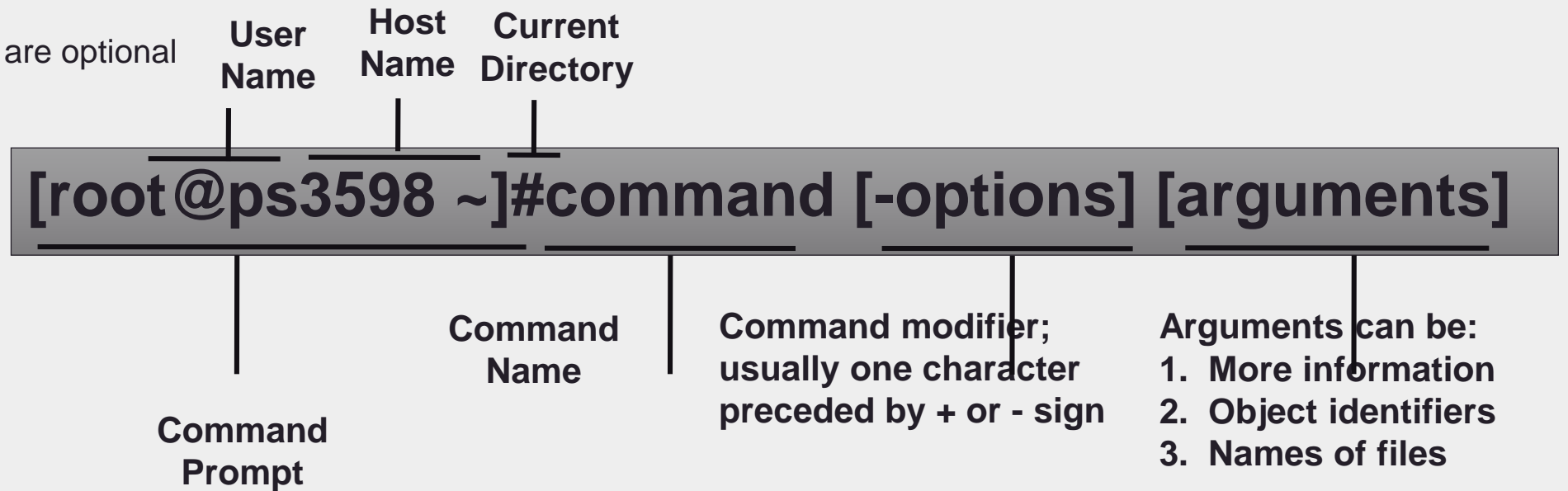
## Logging off & Shutting Down

- To log off, simply type *exit* or *logout* (if it is a login shell)
- You can also press Ctrl+D to logoff or terminate your session
- Unix provides several commands for shutting down a system, although not all may be available on some versions
- Only the root can execute the following commands

Command	Function
halt	Brings the system down immediately
init 0	Powers off the system using predefined scripts to clean up the system prior to shutdown (Not available on all Unix versions)
init 6	Reboots the system (Not available on all Unix versions)
poweroff	Shuts down the system by powering off (Not available on all Unix versions)
reboot	Reboots the system (Not available on all Unix versions)
shutdown	Shuts down the system

# Command Line Structure

- UNIX IS CASE SENSITIVE!!! Always use lowercase
- Example: the command “ls -l” is not the same as “LS -L”
- There must be a space between the command, options and arguments
- No space between the plus or minus sign and the option letter
- Optional letters must be typed exactly as they are indicated, uppercase or lowercase. GNU options can also have long form e.g. --color, --all
- Fields enclosed in [ ] are optional



## Helpful Commands

Command	Function
<b>Ctrl-A &amp; Ctrl-E</b>	Go to the start & end of the command line
<b>Ctrl-C</b>	Terminates the current command
<b>Ctrl-S</b>	Stops scrolling of the output on the screen
<b>Ctrl-Q</b>	Resumes scrolling of the output on the screen
<b>Ctrl + I or type clear</b>	Clears the entire screen
<b>TAB</b>	Completes the command when you type a part of it
<b>Up/Down arrow keys</b>	Cycle through command history

## Getting Help

- You can get help in Unix through different commands

Command	Function
<b>man</b>	Displays the manual page for the command
<b>info</b>	An advanced man command that displays improved manual pages created by Texinfo. Usually available with GNU/Linux
<b>whatis</b>	Searches a set of database files that contain short descriptions of system commands
<b>apropos</b>	Searches the whatis database for strings. Same as man -k

# Man Pages

- Linux distributions include man pages for most standard commands, system calls, and standard library functions. The man pages are divided into numbered sections ; for programmers, the most important are these:
  - (1) User commands
  - (2) System calls
  - (3) Standard library functions
  - (8) System/administrative commands

## Using man Command

- In Unix and its variants, man pages (manual pages) are the standard way of providing documentation and can be displayed through man command
- To see help on any command just type *man <command>*
- You can also view help by sections
- For example, *man 3 printf* will display the library function instead of the command
- On some systems, you might have to specify the -s switch to see help from a particular section

## Navigating inside man pages

Command / key	Action
e, j, Enter, or Down	move forward one line
y, k, or Up	move backward one line
f, Space, or Page Down	move forward one page
b, or Page Up	move backward one page
/characters	search the manual page for the specified characters
q	quit



## man Sections

- man pages are divided into following sections

Section	Command Types	Examples
1	General Commands	ar, chmod, chown, cat
2	System Calls	exit, close, connect
3	C Library Functions	abs, printf, atoi
4	Special Files (/dev)	fd, hd, lp, ram, tty
5	File Formats	aliases, cvs, inittab
6	Games	intro, ddate
7	Miscellanea	ascii, locale
8	System Administration	fdisk, fsck, crond

## Basic Commands

Command	Function
date	Displays the current date and time
cal	Displays a calendar
who/users	Lists all users who are currently logged in
ls	Lists the contents of a directory
cat	Concatenates the files and prints on standard output
pwd	Prints the name of current working directory
uname	Gives system information
w	provides information about logged in users
hostname	Shows the name of the system
type	A shell command that displays the type of command

## Assignments

- Display the date/time in following formats
  - Full month name, MMM, yy, MMM dd yy, dd/mm/yyyy, hh:mm day, hh:mm:ss AM/PM day dd/mm/yyyy
- Display the calendar for the following
  - current month, current year, September 1752
- Use the who command to display
  - no. of users logged in, time of last system boot, current runlevel
- Use ls to display
  - all files, including hidden files (these start with a dot), hidden files only, space occupied by each file in KB format, only directories, type of files displayed, sorted files