Chapter 1 Review: The Essence of Unix and Linux

List of Common Operating Systems you may have encountered:

Android – Linux-based operating system on phones and tablets (Google)
Mac OSX – Unix-based (BSD) operating system on Apple desktops and laptops
iPhone iOS – operating system on Apple phones
Windows – Microsoft's flagship operating system for desktops
MS-DOS – Microsoft's non-graphical operating system from the 80's whose shell interface still exists in the command prompt of Windows
Unix – an operating system developed by Dennis Ritchie and Ken Thompson at Bell Labs as a counterpoint to another OS, MULTICS. Unix spawned many off-shoot operating systems including, eventually modern-day Linux.
Linux – an open-source operating system that is UNIX-based written by Linus Torvalds
ChromeOS – a lean Unix-based operating system for ultra-portable laptops (casual users)

Two Unix terms for operating systems:

Unix-like – a cluster of operating systems that all behave similar to Unix or are based off Unix
POSIX-compliant – an IEEE standardization which many operating systems adhere to that ensure they all behave in a similar well-defined and predictable way that is Unix-like

Versions of Unix and predecessors:

Multics – a multi-user time-sharing operating system that predates Unix and was influential in Unix's design choices. The tree directory structure and shell interface of Multics were taken for Unix. Mostly however, Unix was influenced by Multics as an example of how not to design an operating system.
AIX- a version of Unix built by IBM
Solaris – a Unix-based operating system developed by Sun Microsystems
BSD – a less-business-oriented version of UNIX out of Berkeley that rivaled System V in the 80's
SystemV – one of the first commercial versions of the Unix operating system released in 1983

How did Unix differ from OS's that came before it?

Unix was a general-purpose OS that was customizable and whose code was shared and community-developed. It was portable to other machines besides the one it was originally written for

Versions of Linux:

Debian

- Ubuntu
  - Mint
- Raspbian
- Kali

RPM

- Red Hat Enterprise Linux
- CentOS
- Fedora
- Suse

Arch Linux

Definition of Operating System

Operating System Structure

- Hardware – the physical components that make-up a computer
- BIOS – a basic input output system that manages hardware and launches a bootloader
- bootloader – responsible for launching an operating system
- Kernel – the core foundational code of an operating system
- OS – a set of basic software that governs the way a computer is used
- Shell – an interactive system that end-users use to interact with and issue commands to an OS
  - Gui – a graphical shell
  - Terminal – a text shell modeled after the way output was received from printers before monitors

Servers vs. Clients

A server provides a service (computation or content) to a client. Clients are often desktop machines that regular people use while servers are administered by IT professionals to run some type of information service.

Networks – a group of computers that communicate to one another.

The History and Design of Unix

Unix was developed in the late 60's as a response to Multics. Multics was a time-sharing multi-user operating system. Unix was originally a single process machine. Unix, once converted to a C code base was the first portable (machine indepenedent) operating system. This is what made it take off. Prior to this, operating systems were tied to the hardware they were written for. After a while, Unix incorporated the time-sharing and multi-user functionality of Multics.

Unix was designed informally by a group of people and the code was freely distributed. People were encouraged to extend the system with their own tools. This made the ecosystem and the core OS grow as many features were re-absorbed into the base.

Unix was designed with a tree hierarchy file system, a small secure kernel, and a shell for user interaction. This design has been copied by most subsequent operating systems

The Unix design philosophy is do one thing and one thing well. Programs are chained together by pipes which feed streams of data into each other. The power of Unix and it’s utilities is in its ability to compose many simple programs together to do things that are greater than the sum of their parts. This modular design is at the heart of the OS versatility, customizability, and success.

In the 80’s Linux split into a codebase made by AT&T (System V) and one by Berkeley (BSD). AT&T sued Berkeley which slowed down the development of Unix. This prompted Richard Stallman to begin writing about the open-source software movement and a necessity to keep software free and open.

In the early 90’s Linus Torvald’s built Linux. Linux took off and ascended to the throne as the most-used Unix-like system. To this day, Unix-like systems run most servers on the internet as well as most supercomputers.

Shells of Linux

- GUI’s
  - GNOME
    - GNOME 2
      - Cinnamon
      - Unity
      - Mate
    - GNOME 3
  - KDE
- Terminals
  - Bourne
  - Korn
  - Cshell
  - Bash
  - Powershell (ported from Windows)

Linux Basics

Linux is composed of a kernel written originally by Linus Torvalds.
Bootloader: Grub
Text-Shell: Bash
Gui-Shell: KDE or GNOME
package-manager: rpm or debian
Process-manager: int.d or systemd, manages running programs in the os
process – a program that is running in system memory, represented as nodes on the filesystem in /proc
daemon – a process that runs in the background
filesystem – a tree structure that organizes data that users and the os interact with
file – a node in the filesystem that holds data (either text or binary)
user space – commands that can be issued by a user without invoking critical system routines
root space – a space where commands can only be executed with the approval of the kernel
permissions – files have permissions that restrict read, write, and execution access to them
groups – users can belong to groups and get access to certain files and directories
ownership – files are owned by a particular person and belong to a particular group
devices – devices are represented as nodes on the file system in /dev
package manager – a program that manages what programs are installed on the computer
stream – a data source that flows from a source to a destination, these objects are fundamental to interprocess communication in Linux and Unix
passwd userName – changes the password of the user named ‘userName’
forking – an activity by which a process spawns a child of itself
script – a file that executes commands

Using Bash

cd – move to your home directory
cd / - move to the root directory
cd /folderName – move to the folder named ‘folderName’
cd .. - move to the parent directory
cd . - move to the current directory
pwd – get the path of the current directory
touch fileName – create a file named ‘filename’
mkdir dirName – make a folder named ‘dirName’
rm fileName – remove a file named fileName
mv fileName renamedFileName – move a file from one place to another
cp fileName cName – copy a file from one place to another
echo “a message” - print out a message to standard output
> - redirects the standard output stream of one program to a file
>> - appends the standard output of one program to the end of a file.
date – returns the date
cal – returns a calendar
man programName – delivers a help page for a program
who – displays who is using the machine
whoami – displays what user you are at the moment
ifconfig – displays some basic network configuration information about the system you are on
cat file1 file2 file3 – outputs the contents of all the files provided as a single combined or concatenated stream
clear – clears the screen
useradd userName– creates a user named “userName”
history – outputs the previous commands typed into the shell
| - redirects standard output to another program
more – displays the output of a program one screen at a time
less – also displays output of a program one screen at a time but lets you traverse the contents
head – displays the beginning of a programs output
tail – displays the end of a program’s output
chmod – changes the permissions of a file
chown – changes the ownership of a file
top – displays a screen of the top running processes on the machine
whatis programName – provides a brief explanation of a program named “programName”

Important Terms

operating system:
personal computer (PC):
network:
server operating system:
server:
host:
clients:
mainframe:
server-based network:
peer-to-peer network:
multiuser systems:
log in:
multitasking systems:
portability:
file transfer protocol (FTP):
Telnet:
terminal:
authentication:
SystemV:
BSD:
Portable Operating System Interface for UNIX (POSIX):
kernel:
kernel mode:
user mode:
GNU:
Debian:
Fedora:
Knoppix:
Mandriva:
Red Hat Enterprise Linux:
CentOS
openSUSE Linux:
SUSE Linux Enterprise:
Ubuntu:
Mint:
Bourne Shell:
Korn Shell:
C shell:
Bash shell:
graphical user interface (GUI):
terminal window:
Internet Protocol (IP) address:
domain name:
Secure Shell (SSH):
command:
case sensitive:
syntax:
options:
arguments:
system administrator:
ordinary users:
superuser:
root:
partition:
swap space:
root directory:
ext4:
ntfs:
xfs:
file system formats:
virtual machine:
64-bit:
32-bit:
GNOME:
KDE: