**1280 Chapter 1 Notes**

* 1. **Operating System**
* Operating System
  + Allocates memory for the program
  + Loads the program into memory
  + Loads the program address into the CPU registers and sends program instructions to CPU
  + Accesses hardware on behalf of the program
  + It is a piece of software
* Modern OS’s are multiprogramming
  + Multiple programs to reside in memory
  + Rather than allow a single program to run to completion it lets it run for an instant, places a save state, and loads the next program in the queue.
  + Creates a **process** for each program and controls the switching of these processes
  + Programs often accesses I/O to read/write data. These keep CPU idle so the OS takes the program out of the CPU and loads the next process in the queue
  + *QUESTION: MORE CLARIFICATION ON THE I/O BUFFERS AND DEVICE DRIVERS*
  1. **The UNIX Operating System**
* Assembler
  + Near machine language code
  + One to one instructions
  + Every variable converted to its memory address
* Shell
  + Command Interpreter
  + Key in a word, and the shell interprets it as a command to be executed
  1. **Knowing Your Machine**
* Workstation
  + Meant to be used by single user (high powered)
  + UNIX can be used by hundreds of users using pc’s connected to a central UNIX computer
    - Why use?
      * Administered properly, keep files there to be backed up regularly
      * Use a program workstation has but your computer doesn’t
      * All incoming/outgoing mail handled by the workstation
  + Has own CPU, RAM, DVD, keyboard, mouse, printer
* **Terminal Emulation**
  + Abandons normal mode and behave like a terminal
  + Workstation doesn’t use it’s HD, RAM, CPU, except for terminal emulation resources
  + Run a program on a remote machine using their resources
* **Character**
  + The smallest piece of information we can deal with
  + **ASCII**
    - American Standard Code for Information Interchange
    - Take up 1 byte of memory
  1. **The System Administrator**
* In charge of a large system serving hundreds of users
  + Allocates user accounts, maintains file systems, takes backups, manages disk space, etc.
* User ID and password required if not admin
  + Files located in different spaces
* **Root**
  + Special login id
  + Near absolute power
  + Some programs can only be run from here
  1. **Logging In and Out**
* SSH
  + Secure shell used for logging in, Telnet do not log in with for security reasons
    - Doesn’t encrypt any information sent over the connection
  + The shell that we will be using mostly is BASH
    - Other shells include C-Shell, Bourne
  1. **A Hands-On Session**
* date
  + prints the current system date and time
* who
  + prints a list of the current users
  + displays date and time of login
* ps
  + displays list of the current processes
  + Bash shell
    - Unique PID of 4249
    - Logout kills the process
    - Another shell is korn shell
* Echo $SHELL
  + Displays the shell that is running
  + $SHELL is a shell variable
* UNIX maintains all data in containers called files, places all these filenames together it another file called directory
* Creating a File with echo
  + Echo date > foo
  + > is a redirection operator
    - it redirects messages to be displayed on the console to a file
* Displaying a file with cat
  + Cat command displayes the contents of files
  + Date and foo in this case are known as **arguments**
    - These arguments will affect the commands behavior
* Copying a file with cp
  + Cp foo foo.sh
  + Cp needs two arguments
* Displaying a list of filenames with ls
* Renaimg a file with mv
  + Mv foo.sh foo.shell
* Removing a file with rm
* The files foo.sh and foo.shell are ordinary files
  + Every file has an association with a directory
    - We day a file resides in a directory
    - A user is associated with a directory as well
      * They are placed a directory called the current directory
    - Directories are considered files
* Creating a directory with mkdir
  + Using ls –F will place a backslash after directories to see them easier
    - This argument begins with a hyphon since it is an option
      * Options change the default behavior of a command
  + Mkdir scripts

**Ch 1 Notes Cont’d**

* **Copying a file to a directory**
  + Cp foo scripts
  + This will copy it into the directory
* **Directory Navigation with pwd and cd**
  + Pwd command will print the working directory for you
    - The output shows a **pathname**
      * Represents a hierarchy of directory names
      * The first / represents the top-most directory called root
  + Cd will change the directory
    - Descend one level in the hierarchy (cd scripts)
  + Sh foo
    - Runs shell command
    - Interprest any argument as a file to take its input form, it runs the date command
  + The file foo is known as a shell script
    - If we place more command strings in this file they will all be executed in a batch
  + Exit terminates shell session

**1.7 How it all Clicked**

* Vi is the standard editor
* Berkeley created their own unix BSD UNIX
  + Berkeley Software Distribution
  + Sun used BSD system to create their own brand of UNIX (Sun OS)
    - Together their version of UNIX is known as Solaris
  + IBM has AIX
  + HP has HP-UX
  + DEC has digital UNIX
    - And now Tru64 UNIX
  + Apple Mac OS X
* AT&T unified many of its own flavours into its own release
  + System V Release 4 (SVR4)
  + AT&T then sold UNIX to Novell, who later turned over the UNIX trademark to a standards body called X/OPEN, now merged with The Open Group
* ARPANET (DARPA)
  + Packet switching technology
  + Data is split into packets, which can take different routes and yet be assembled in the right order
  + Birth of TCP/IP
    - Transmission Control Protocol/Internet Protocol
      * A collection of protocols used for networking computers that use different operating systems and different hardware. Ensures reliable transmission with full error-correcting facilities
  + DARPA commissioned UCB to implement TCP/IP on BSD UNIX. ARPANET converted to TCP/IP in 1983
    - In the same year Berkeley released the first version of UNIX with TCP/IP built in
    - This was key to the growth of the internet and UNIX
* In the mean time Windows was doing great things with GUI (graphical user interface) that uses the mouse rather than keyboard commands
  + X Windows was created for Unix’s survival to compete

**1.8 POSIX and the Single UNIX Specification**

* UNIX fragmentation and the absence of a single conforming standard adversely affected the development of portable applications
  + AT&T created System V Interface Definition (SVID)
  + x/Open (the Open Group) then created the X/Open Portability Guide
* POSIX
  + Portable Operating System Interface for Computer Environments
  + Developed at the behest of IEEE (Institution of Electrical and Electronics Engineers)
  + POSIX refers to operating systems in general but was based on UNIX.
    - POSIX.1
      * Specifies the C application program interface (system calls)
    - POSIX.2
      * Deals with the shell and utilities
  + Single Unix Specification, Version 3 (SUSV3) also known as POSIX.1
  + Write once adopt everywhere approach
    - Once software has been developed on any POSIX compliant system, it can be ported to another POSIX compliant UNIX machine with minimal modifications

**1.9 Linux and GNU**

* Richard Stallman
  + Runs Free Software Foundation
    - Formally known as GNU (GNU’s not UNIX)
    - Many important linux tools were written and supplied for free by GNU
* Linus Torvalds
  + Father of LINUX
* Linux is distributed under GNU General public license
  + Makes it mandatory to make the source code public
* All of the major computer vendors (barring Microsoft) have committed to support Linux

**1.10 The UNIX Architecture**

* Division of labour between the *kernel* and the *shell*
  + Kernel interacts with the machine’s hardware
  + Shell interacts with the user
* **Kernel**
  + Core of the operating system
    - System’s bootstrap program loads the kernel into memory at startup
    - Kernel comprises a set of routines (written in C mostly) that communicate with the hardware directly
  + Manages system’s memory
  + Schedules processes
    - Decides their priorities
  + A program’s gateway to the computer’s resources
* **Shell**
  + Command interpreter
  + Interface between the kernel and the user
* **The File and the Process**
  + These two simple entities support the UNIX system
  + The files have a place and processes have life
  + **Files** are containers for storing static information
    - Directories and devices are considered to be files
    - A file is related to another file by being part of a single hierarchical structure called the *file system*
  + **Process** represents a program in execution
    - Processes also form a hierarchy
* **The System Calls**
  + The code for performing disk I/O operations is available in the kernel
  + Programs access these kernel services by making **system calls**
  + **Open** system call opens both a file and a device
  + C programmer in UNIX has access to the entire system call library as well as the standard library function. They don’t have to import them like in windows
  + System Calls
    - User programs that need to access the hardware communicate with the kernel using a set of functions called system calls

**1.11 Features of UNIX**

* **A MultiUser System**
  + UNIX is a **multiprogramming** system
    - Permits multiple programs to remain in memory and compete for the attention of the CPU
  + For cycling through the jobs, the kernel uses the principle of **time-sharing**
    - Breaks up a unit of time into several slices, and a user’s job runs for the duration of a slice.
    - The moment the allotted time expires, the previous job is kept in abeyance and the next job is taken up.
    - This process goes on until the clock has turned full circle and the first job is taken up once again.
    - This switching happens several times in one second so every user has the feeling that the machine is completely dedicated to him
* **Multi-Tasking System**
  + A single user can run multiple tasks concurrently
  + One job runs in the foreground, the rest run in the background
  + You can switch jobs between background and foreground, suspend them, or even terminate them
* **A repository of applications**
* **Building Block Approach**
  + Small is beautiful
    - Complex task can be broken into a finite number of simple ones
    - Shell offers a mechanism called the pipe that allows the output of one command to serve as the input to another.
      * These special set of commands are called filters
    - This is why a lot of commands don’t require user input, they can be scheduled to run at a certain time of the day.
      * It’s output can be used by another program without user intervention
* **Pattern Matching**
  + Can back up programs using the **tar** command
  + \* is a special character (known as a metacharacter) that is used by the shell to match a number of characters
  + **Grep, sed, and awk**
    - Use a different metacharacter set for pattern matching
    - A pattern represented by printf.\*name matches all the lines that contain printf and name
      * This pattern is called a *regular expression*
* **Programming Facility**
  + A programming language in its own right that is used to design **shell scripts**
    - Programs that run UNIX commands in a batch
* **Documentation**
  + Principle help facility is the **man** command