1

UNIX System Overview

1.1) Introduction

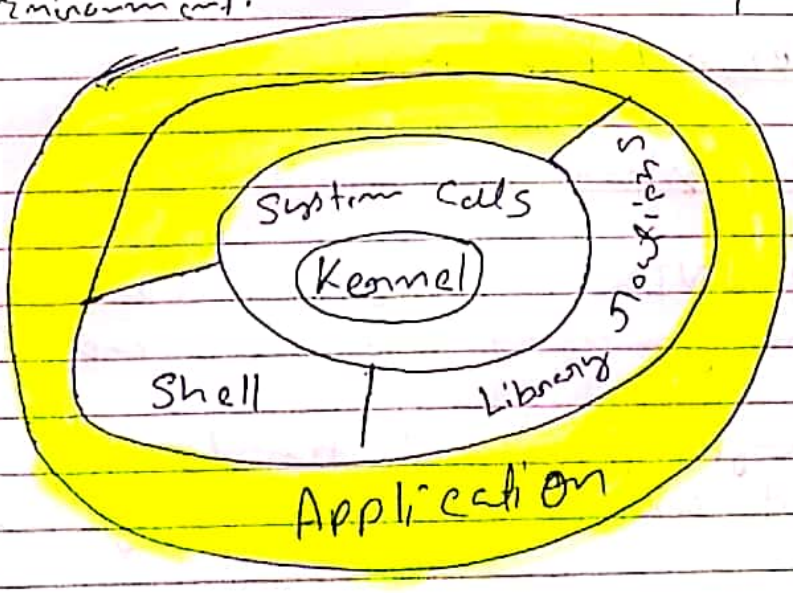
Operating System — Program

(Unix System from a Programmers perspective)

Operating System →

Software that controls the hardware resources of the Computer and provides an environment under which programs can run.

Generally, we call this software the Kernel, Since it is relatively small and resides at the core of the environment.



⇒ Libraries of common functions are built on top of the system call interface, but applications are free to use both.

⇒ The shell is a special application that provides an interface for running other applications

1.3) Logging In

/etc/passwd

→ Contains user name and relevant information.

* Shells

User input to shell

→ Terminal (Interactive shell)

→ Shell script

1.4) Files and Directories

* File System

⇒ The UNIX file system is a hierarchical arrangement of directories and file.

⇒ Everything starts in the directory called root, whose name is the single character /

⇒ A directory is a file that contains directory entries.

↳ We can think of each directory entry as containing a filename along with a structure of information describing the attributes of the file.

⇒ The stat and fstat function returns a structure of information containing all the attributes of a file.

* Working directory

⇒ Every process has a working directory.

↳ This is the directory from which all relative paths are interpreted.

* Home Directory

⇒ When we log in, the working directory is set to our home directory.

1.5) Input and Output

* File Descriptors

→ It is a small non-negative integer that the kernel uses to identify the files accessed by a process.

⇒ Whenever it opens an existing file or creates a new file, the Kernel returns a file descriptor that we use when we want to read or write the file.

* Standard Input, Standard Output & Standard Error

⇒ By convention all shells open three descriptors whenever a new program is run.

* Unbuffered I/O

⇒ It is provided by the functions open, read, lseek and close.

⇒ These functions all work with file descriptors.

* Standard I/O

⇒ The Standard I/O functions provide a buffered interface to the unbuffered I/O functions.