**UNIX AND SHELL PROGRAMMING**

**Assignment-1**

**1.Explain architecture of UNIX?**

### A: Unix Architecture:

The architecture of UNIX is basically divided into four main layers-

1. Kernel
2. Hardware
3. Shell
4. Files and directories

* The first layer kernel deals with all the hardware connections all the applications and commands are associated with it.
* Hardware is the second layer.
* The third layer known as shell acts as the bridge between the user, user commands and predefined UNIX commands.
* Last but not the least files and directories can be concluded as the UNIX commands which will be UNIX programs in most cases. The final layer is the user. This means the whole operating system is visible to the user from the shell itself.



**1.Kernel:** Amongst the four layer’s kernel is the most powerful one. The kernel contains mainly utilities along with the master control program. Kernel program has the power to start or stop a program and even handle the file system. It also suggests which program to be selected when two resources try to access the device at the same time for the same resource. As the kernel has special access to the OS this leads to the division of space between user-space and kernel-space.

Kernel structure is designed in such a way it should support primary UNIX requirements. Which are divided into two categories and listed below

1. Process management.
2. File management.

* **Process Management :** The resource allocation in CPU, memory, and services are few things which will be handled under process management.
* **File Management :** File management deals with managing all the data in files needed by the process while communicating with devices and regulating data transmission.

**The main operations done by the kernel are**

1. Kernel ensures the running of user-given programs is done on time.
2. Plays a role in memory allocation.
3. Manages the swapping between memory and disk.
4. Transports data between peripherals.
5. The kernel also requests service from the process.

That’s the reason why the kernel is called as the heart of the UNIX system. The kernel itself can be defined as a small program that contains enough data structure to pass arguments and receive results from a call and the process them on the calling process.

**2.Hardware:** Hardware can be defined as the system components which are seen through the human eye and be touched like keyboard, monitors, etc., Hardware also included speakers, clocks, devices in OS architecture.

**3.Shell:** The shell can easily be defined as the software program which acts as a communication bridge between kernel and user. When the user gives the commands the shell reads the commands, understands them and then sends a request to execute the program. Then when the program is executed it again sends the request to display the program to the user on-screen. The shell can also be called a command interpreter. As told above the shell calls the kernel there are all most 100 in build calls.

**Various tasks which shell ask the kernel to do are**

1. File opening.
2. File writing.
3. Executing programs.
4. Obtaining detailed information about the program.
5. Termination of the process.
6. Getting information about time and date.

**4.Files and Directories:** This includes user-written and shell programs as well as libraries of UNIX

**Directories**: Directories in Unix have name, path, files, and folder. Rules for both files and folders are the same. These are stored in the up-side-down hierarchical tree structure. The main working flow of directories is as follows.

* Displays home directories.
* Copies files to other directories.
* Renaming directories.
* Deleting directories.

**Files:** These are the files that contain data, text and program instructions. The main workflow of files is

* Store user information like an image drawn or some content written.
* Mostly located under a directory.
* It does not allow/store the data of other files.

**2.Explain the features of UNIX?**

**A: Portability:**

The system is written in high-level language making it easier to read, understand, change and, therefore move to other machines. The code can be changed and complied on a new machine. Customers can then choose from a wide variety of hardware vendors without being locked in with a particular vendor.

**Machine-independence:**

The System hides the machine architecture from the user, making it easier to write applications that can run on micros, mins and mainframes.

**Multi-Tasking:**

Unix is a powerful multi-tasking operating system; it means when a active task in in process, there can be a simultaneous background process working too. Unix handles these active and background threads efficiently and manages the system resources in a fair-share manner.

**Multi-User Operations:**

UNIX is a multi-user system designed to support a group of users simultaneously. The system allows for the sharing of processing power and peripheral resources, white at the same time providing excellent security features.

**Hierarchical File System:**

UNIX uses a hierarchile file structure to store information. This structure has the maximum flexibility in grouping information in a way that reflects its natural state. It allows for easy maintenance and efficient implementation.

**UNIX shell:**

UNIX has a simple user interface called the shell that has the power to provide the services that the user wants. It protects the user from having to know the intricate hardware details.

**Pipes and Filters:**

UNIX has facilities called Pipes and Filters which permit the user to create complex programs from simple programs.

**Utilities:**

UNIX has over 200 utility programs for various functions. New utilities can be built effortlessly by combining existing utilities.

***Software Development Tools:***

UNIX offers an excellent variety of tools for software development for all phases, from program editing to maintenance of software.

**3.** **Explain following commands with suitable examples**

**a) wc b) rm c) cat d) stty**

**A: a) wc :**

* “wc” stands for “Word Count”.
* As the name implies, it is mainly used for counting purpose.
* It is used to find out number of lines, word count, byte and characterscount in the files specified in the file arguments.
* By default it displays four-columnar output**.**
* First column shows number of lines present in a file specified, second column shows number of words present in the file, third column shows number of characters present in file and fourth column itself is the file name which are given as argument.

**Syntax:**

**wc [OPTION]... [FILE]...**

**Example:** Let us consider two files having name **state.txt** and **capital.txt** containing 5 names of the Indian states and capitals respectively.

**File1:** **$ cat state.txt**

Andhra Pradesh

Arunachal Pradesh

Assam

Bihar

Chhattisgarh

**File2:$ cat capital.txt**

Hyderabad

Itanagar

Dispur

Patna

Raipur

**Passing only one file name in the argument**

**$ wc state.txt**

5 7 63 state.txt

OR

**$ wc capital.txt**

5 5 45 capital.txt

**Passing more than one file name in the argument**

$ wc state.txt capital.txt

5 7 63 state.txt

5 5 45 capital.txt

10 12 108 total

**b) rm:**

* “rm” stands for remove here.
* rm command is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX. To be more precise, rm removes references to objects from the filesystem, where those objects might have had multiple references.
* By default, it does not remove directories.
* This command normally works silently and you should be very careful while running rmcommand because once you delete the files then you are not able to recover the contents of files and directories.

**Syntax:**

**rm [OPTION]... FILE...**

**Example:** Let us consider 5 files having name **a.txt, b.txt** and so on till **e.txt**.

**$ ls**

a.txt b.txt c.txt d.txt e.txt

**Removing one file at a time**

**$ rm a.txt**

**$ ls**

b.txt c.txt d.txt e.txt

**Removing more than one file at a time**

**$ rm b.txt c.txt**

**$ ls**

d.txt e.txt

**c)cat:**

* “cat” stands for concatenate.

* This command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files. So let us see some frequently used cat commands.

**Examples:**

**1) To view a single file**   
**Command:**   
  $cat filename

**Output** It will show content of given filename

**2) To view multiple files**   
**Command:**   
  $cat file1 file2

**Output** This will show the content of file1 and file2.

**3) To view contents of a file preceding with line numbers.**   
**Command:**   
 $cat -n filename

**Output** It will show content with line number

**4) Create a file**  
**Command:**   
 $ cat >newfile

**Output** Will create and a file named newfile

**5) Copy the contents of one file to another file.**   
**Command:**   
  $cat [filename-whose-contents-is-to-be-copied] > [destination-filename]

**Output** The content will be copied in destination file

**6) Cat command can suppress repeated empty lines in output**   
**Command:**   
 $cat -s geeks.txt

**Output** Will suppress repeated empty lines in output

**7) Cat command can append the contents of one file to the end of another file.**   
**Command:**   
  $cat file1 >> file2

**Output** Will append the contents of one file to the end of another file

**8) Cat command can display content in reverse order using tac command.**   
**Command:**   
  $tac filename

**Output** Will display content in reverse order

**9) Cat command can highlight the end of line.**   
**Command:**   
  $cat -E “filename”

**Output** Will highlight the end of line

**10) If you want to use the -v, -E and -T option together, then instead of writing -vET in the command, you can just use the -A command line option.**   
**Command**  $cat -A "filename"

**d) stty:**

This command shows or changes terminal characteristics.

**Syntax:**

stty [-F DEVICE | --file=DEVICE] [SETTING]...stty [-F DEVICE | --file=DEVICE] [-a|--all]stty [-F DEVICE | --file=DEVICE] [-g|--save]

**Example:** It will display the characteristics of the terminal.

