1. **Create the following files: chapa, chapb, chapc, chapd, chape, chapA, chapB, chapC, chapD, chapE, chap01, chap02, chap03, chap04, chap05, chap11, chap12, chap13, chap14, and chap15.**

* Use the ‘touch’ command to create a chapa, chapb, chapc, chapd, chape, chapA, chapB, chapC, chapD, chapE, chap01, chap02, chap03, chap04, chap05, chap11, chap12, chap13, chap14, and chap15 files.

1. **With reference to question 1, What is the command for listing all files ending in small letters?**

* Use ‘ls \*[a-z]’ command to listing all files ending in small latters.

1. **With reference to question 1, What is the command for listing all files ending in capitals?**

* Use ‘ls \*[A-Z]’ command to listing all files ending in small latters.

1. **With reference to question 1, What is the command for listing all files whose last but one character is 0?**

* Use ‘ls \*0’ command to listing all files ending in small latters.

1. **In an organization one wants to know how many programmers are there. The employee data is stored in a file called ‘personnel’ with one record per employee. Every record has field for designation. How can grep be used for this purpose?**

* Use ‘grep "programmer" personnel | wc –l’ command to know how many programmers are there.

1. **In the organization mentioned in question 6 how can sed be used to print only the records of all employees who are programmers.**

* Use ‘sed -n '/programmer/p' personnel’ command to print only the records of all employees who are programmers.

1. **In the organization mentioned in question 6 how can sed be used to change the designation ‘programmer’ to ‘software professional’ everywhere in the ‘personnel’ file.**

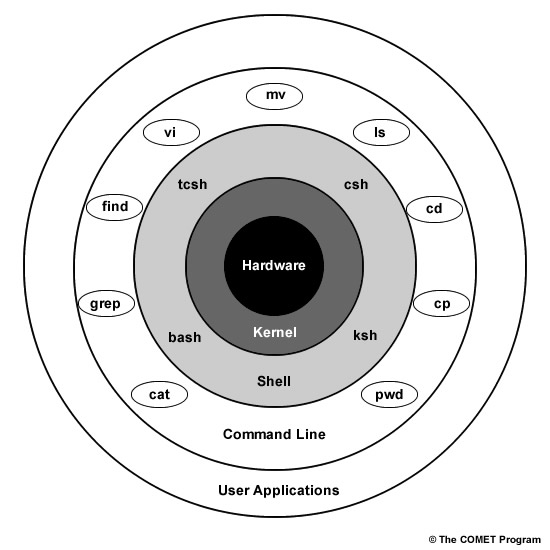
* Use ‘sed 's/programmer/software professional/g' personnel > updated\_personnel’ command to change th designation ‘programmer’ to ‘software professional’ everywhere in the ‘personnel’ file.

1. **How do you get the status of all the processes running on the system? i.e. using what option?**

* Use ‘ps –e’ or ‘ps aux’ command to get status of all the process running on the system.

1. **Explain Unix Architecture.**

* Unix is made up of 3 main parts: the kernel, the shell, and user commands and applications**.**



The kernel and shell are the heart and soul of the operating system.

The kernel ingests user input via the shell and accesses the hardware to perform things like memory allocation and file storage.

The shell is an interface that interprets the command line input and calls the necessary programs to do the work. The commands that you enter are programs themselves, so once the work is done, the command line will return to a prompt and await further input.

There are several different shells, and syntax and shortcuts vary between them. For example, the "csh" shell listed in the image above is called "C shell" and has syntax similar to the C programming language. All shells support similar basic functions.

One example of how the shell and kernel work together is copying a file. If you want to copy a file named "file1" and name the copy "file2", you would enter "cp file1 file2" at the command line. The shell will search for the program "cp" and then tell the kernel to run that program on "file 1" and name the output "file 2". When the copying is finished, the shell returns you to the prompt and awaits more commands.

1. **Define single-user system.**

* Here 2 types of single-user system.

1. Single User Single-tasking system.
2. Single User Multy-tasking system.
3. Single User Single-tasking system.

* An operating system that allows a single user to perform only one task at a time is called a Single-User Single-Tasking Operating System. Functions like printing a document, downloading images, etc., can be performed only one at a time. Examples include MS-DOS, Palm OS, etc.

Advantage: This operating system occupies less space in memory.

Disadvantage: It can perform only a single task at a time.

1. Single User Multy-tasking system.

* An operating system that allows a single user to perform more than one task at a time is called Single-User Multitasking Operating System. Examples include Microsoft Windows and Macintosh OS.

Advantage: It is time saving as it performs multiple tasks at a time yielding high productivity.

Disadvantage: This operating system is highly complex and occupies more space.

1. **Name a few significant features of UNIX?**

* Multiuser/Multitasking: Supports many users and tasks concurrently.
* Hierarchical File System: Organizes files and folders in a tree structure.
* Command-Line Interface: Interact using text commands.
* Portability: Can run on different hardware.
* Modular Design: Components can work together or independently.
* Small Utilities: Simple tools combined for complex tasks.
* Networking: Early networking capabilities laid groundwork for the internet.
* Security/Permissions: Control over file access and modification.
* Text-Based Configuration: Configurations in human-readable text.
* Process Control: Manage running programs efficiently.

1. **Can you write a command to erase all files in the current directory including all its subdirectories?**

* To delete all files in the current directory, including its subdirectories, you can use the ‘find’ command in combination with the ‘rm’ command.

1. **Describe a link in UNIX.**

* In UNIX operating systems, a "link" refers to a mechanism used to create multiple references (pointers) to a single file or directory within the file system. There are two main types of links in UNIX: hard links and symbolic (soft) links.
* Hard Link: A hard link is an additional reference to an existing inode (data structure that stores file information) of a file or directory. All hard links pointing to the same inode are essentially equivalent; there's no primary or original copy. When you create a hard link, it effectively increases the reference count of the inode. Hard links can only be created for files (not directories) that exist on the same file system. Example: ‘ln sourcefile hardlink’
* Characteristics of hard links:
* They share the same data and metadata (file permissions, ownership, etc.) as the original file.
* If you delete the original file, the hard link(s) will still point to the data until all references are removed.
* You cannot create hard links across different file systems or partitions.
* Symbolic Link (Soft Link): A symbolic link, also known as a soft link, is a separate file that acts as a pointer to another file or directory. Unlike hard links, symbolic links are essentially shortcuts or references to the file's pathname rather than the inode itself. If you create a symbolic link, and the original file's pathname changes or it's deleted, the symbolic link becomes broken.

Example: ‘ln -s sourcetarget symboliclink’

* Characteristics of symbolic links:
  + They can point to files or directories, and even across different file systems.
  + They can span across partitions and file systems.
  + If the original file is deleted or moved, the symbolic link becomes "dangling" or broken.