SHEKHAWATI INSTITUTE OF TECHNOLOGY, SIKAR (RAJASTAHN)

Master of Computer Application (IV SEM)

I Midterm Exam 2018 (IV SEM MCA)

Subject Code & Name: MCA-403 & Open source Operating System

M.M:20 Time:1:30 Hour

---

1. Explain (a) OSOS (b) Kernel
2. Describe the architecture of Linux.
3. What do you mean by Linux metacharacters?
4. What is shell? Describe shell interpretive cycle.
5. Describe interactive and non-interactive shell.
6. Explain command for compile and run C program.
7. Write a shell script to calculate the factorial of a number.

Answers:

Ans 1. (a) **Open Sourse Operating System:** Open source refers to a program or software in which the source code (the form of the program when a programmer writes a program in a particular programming language) is available to the general public for use and/or modification from its original design free of charge. Open source code is typically created as a collaborative effort in which programmers improve upon the code and share the changes within the community.

Ex: Android is open source, so phones of different manufacturer have different look and feel to it. Samsung has touchwiz ui, htc has sense, etc.

(b) **Kernel** is the central part of an operating system. A kernel is the core component of an operating system. Using inter process communication and system calls, it acts as a bridge between applications and the data processing performed at the hardware level. It manages the operations of the computer and the hardware - most notably memory and CPU time. There are two types of kernels:

- A microkernel, which only contains basic functionality;

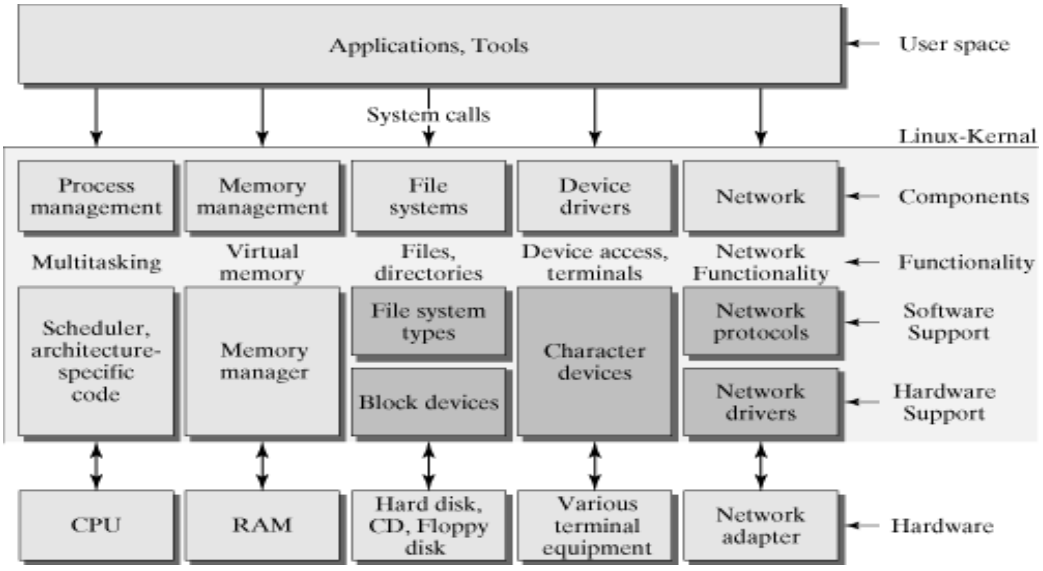
- A monolithic kernel, which contains many device drivers.

When an operating system is loaded into memory, the kernel loads first and remains in memory until the operating system is shut down again. A computer user never interacts directly with the kernel. It runs behind the scenes and cannot be seen, except for the text logs that it prints.

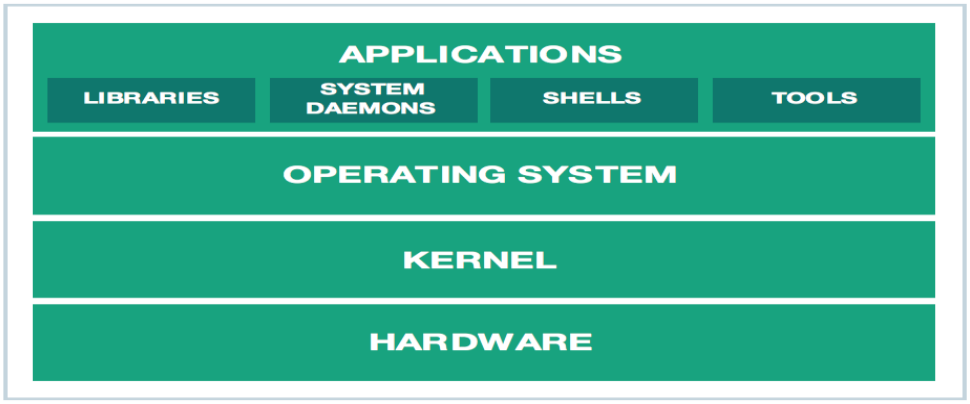
Ans 2. **Architecture of Linux**

Linux System Architecture is consists of following layers:

- **Hardware layer** – Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
- **Kernel** – Core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
- **Shell** – An interface to kernel, hiding complexity of kernel's functions from users. Takes commands from user and executes kernel's functions.
- **Utilities** – Utility programs giving user most of the functionalities of an operating systems.



The operating systems on these hardware platforms are what enable them to run applications



Ans 3. **Special Characters (Shell Metacharacters)**

Special characters, or metacharacters, have a special meaning to the shell. They can be used as wildcards to specify the name of a file without having to type out the file's full name. Some of the most commonly used metacharacters are "*", "?", "[]", and "-".

The Asterisk

The * (asterisk) metacharacter is used to match any and all characters. Typing the following command will list all files in the working directory that begin with the letter l regardless of what characters come after it:

```
$ ls l*
```

The * (asterisk) metacharacter can be used anywhere in the filename. It does not necessarily have to be the last character.

The Question Mark

The ? (question mark) metacharacter is used to match a single character in a filename. Typing the following will list all of the files that start with "livefirelabs" and end with a single character:

```
$ ls livefirelabs?
```

Like the asterisk, the ? (question mark) metacharacter can be used as a substitution for any character in the filename.

Brackets

Brackets ([...]) are used to match a set of specified characters. A comma separates each character within the set. Typing the following will list all files beginning with "a", "b", or "c":

```
$ ls [a,b,c]*
```

The Hyphen

Using the - (hyphen) metacharacter within [] (brackets) is used to match a specified range of characters. Typing the following will list all files beginning with a lowercase letter of the alphabet:

```
$ ls [a-z]*
```

If there are directory names meeting the specified range, their name and contents will also be displayed.

Ans 4. **Shell** is a UNIX term for the interactive user interface with an operating system. The shell is the layer of programming that understands and executes the commands a user enters. In some systems, the shell is called a command interpreter. A shell usually implies an interface with a command syntax (think of the DOS operating system and its "C:>" prompts and user commands such as "dir" and "edit").

Shell Name	Developed by	Where	Remark
BASH (Bourne-Again SHell)	Brian Fox and Chet Ramey	Free Software Foundation	Most common shell in Linux. It's Freeware shell.
CSH (C SHell)	Bill Joy	University of California (For BSD)	The C shell's syntax and usage are very similar to the C programming language.
KSH (Korn SHell)	David Korn	AT & T Bell Labs	--
TCSH	See the man page. Type $ man tcsh	--	TCSH is an enhanced but completely compatible version of the Berkeley UNIX C shell (CSH).

Shell interpretive cycle:

As you log on to a UNIX machine, you first see a prompt. You may feel that the system is idling, but actually the shell command is constantly running at the terminal. It never terminates until you log out.

Typical activities performed by the Shell in its interpretive cycle:

- Issues the prompt and waits for the cmd

- Scans the cmd line for metacharacters, expands the abbreviations & recreate a simplified cmd
- Passes this cmd to the kernel for execution
- Waits till the cmd gets executed
- Re-enters into the waiting mode

Ans 5.

An interactive shell: An interactive shell is when you type in the name of the shell after you have logged in to the system. For example

```
1       bash
```

will start an interactive bash shell.

An interactive (bash) shell executes the file **.bashrc** so you have to put any relevant variables or settings in this file.

A non-interactive shell: A non-interactive shell is a shell that can not interact with the user. It's most often run from a script or similar. This means that .bashrc and .profile are not executed. It is important to note that this often influences your **PATH** variable. It is always a good practice to use the full path for a command but even more so in non-interactive shells.

Ex: You can detect if you are in an interactive or non-interactive shell with

```
1 [[ $- == *i* ]] && echo 'Interactive' || echo 'not-interactive'
```

Ans 6. **compile & execute C programs under Linux:**

Procedure :

You can type you C program using any of the editors that are available under Linux such as *vi* or *emacs* or any other editor.

Once you have written and saved your C program using any editor return to the prompt. An *ls* command should display your C program. It should have the .c extension. Now at the prompt type the following

$ gcc -o firstprogram firstprogram.c

If your file is named firstprogram.c then type '-o firstprogram' as the parameter to gcc. This is basically your suggested name for the executable file that gcc would create. In case you

typed something like the following

$ gcc firstprogram.c

You would be having a a.out in the same directory as the source C file. This is the default name of the executable that gcc creates. This would create problems when you compile many programs in one directory. So you override this with the -o option followed by the name of the executable

$ gcc -o hello secondprogram.c

Would create an executable by the name *hello* for your source code named secondprogram.c

Running the executable that you created is as simple as typing the following at the prompt.

$ ./firstprogram

OR

$ ./hello

Or whatever you named your executable.

Ans 7.

```
#!/bin/bash
fact=1
#taking input from user
echo -e "enter a number"
read n
#if enter value less than 0
if [ $n -le 0 ] ; then
echo "invalid number"
exit
fi
#factorial logic
if [ $n -gt 0 ] ; then
for((i=$n;i>=1;i--))
do
fact=`expr $fact \* $i`
done
```

```
fi
echo "The factorial of $n is $fact"
```

Output:

```
enter a number
4
The factorial of 4 is 24
```