NAME:- Sourav Paul
GROUP:- D2
REG. NO:- 20214056
BRANCH:- CSE DEPT.

Shell Theory Assignment

Assignment-A

Write down the answer of the following questions.

1. **What are the Functions of operating system?**

Ans:- The main functions of the Operating System are as follows:-

- **Security**

The operating system uses a password protection to protect user data it also prevents unauthorized access to programs and user data, but for external functionality we need to install malware software to protect the system.

- **Control over system performance**

The operating system monitors overall system setup to help in improving the performance and it also records the response time between service requests and system response so that it has a complete view of the system.

This can help improve performance by providing important information that is needed at the time of troubleshooting problems.

- **Job Accounting**

Operating systems always keep track of time and resources that are used by various tasks and users, this information can be used to track resource usage for a particular user or a group of users.

- **Error detecting aids**

Operating systems constantly monitor the system which helps us to detect errors and also avoid the malfunctioning of computer systems.

- **Coordination between other software and users**

Operating systems help in coordinate and assign interpreters, compilers, assemblers, and other software to the various users of the computer systems.

- **Memory Management**

  The operating system controls the primary memory or main memory. Primary memory is a large array of bytes or words where each byte or word is assigned a certain address. It is a fast storage, and it can be accessed directly by the CPU which is present inside the system. If a program wants to be executed, it should be first loaded in the main memory.

- **Processor Management**

The OS manages the order in which processes have access to the processor, and how much processing time that each process must stay in the multiprogramming environment. This is called process scheduling.

- **Device Management**

An OS manages device communication through respective drivers.
The following activities are performed by the operating system for device management.

- Keeping track of all devices connected to the system.
- The OS designates a program that is responsible for every device which is called the Input/output controller.

- **File Management**

A file system is arranged into directories for efficient navigation and usage. These directories contain other directories and other files.

2) What is kernel and it's types?

Kernel is the important part of an Operating System. The kernel is the first program that is loaded after the boot loader whenever we start a system. The Kernel is present in the memory until the Operating System is shut-down.

Kernel is classified into two main types:

- **Micro Kernel** - It is kernel types which has minimalist approach. It has virtual memory and thread scheduling. It is more stable with less services in kernel space. It puts rest in user space.
- **Monolithic Kernel** - It is one of types of kernel where all operating system services operate in kernel space. It has dependencies between systems components. It has huge lines of code which is complex.
- **Hybrid Kernel -** It is the combination of both monolithic kernel and microkernel. It has speed and design of monolithic kernel and modularity and stability of microkernel.
- **Exo Kernel -** It is the type of kernel which follows end-to-end principle. It has fewest hardware abstractions as possible. It allocates physical resources to applications.
- **Nano Kernel -** It is the type of kernel that offers hardware abstraction but without system services. Micro Kernel also does not have system services therefore the Micro Kernel and Nano Kernel have become analogous.

3) What is deadlock, which are the necessary conditions of deadlock?

Ans:- If a process is in the waiting state and is unable to change its state because the resources required by the process is held by some other waiting process, then the system is said to be in Deadlock.

There are four different conditions that result in Deadlock. These four conditions are also known as Coffman conditions and these conditions are not mutually exclusive.

1. **Mutual Exclusion:** A resource can be held by only one process at a time. In other words, if a process P1 is using some resource R at a particular instant of time, then some other process P2 can't hold or use the same resource R at that particular instant of time. The process P2 can make a request for that resource R but it can't use that resource simultaneously with process P1.

2. **Hold and Wait:** A process can hold a number of resources at a time and at the same time, it can request for other resources that are being held by some other process. For example, a process P1 can hold two resources R1 and R2 and at the same time, it can request some resource R3 that is currently held by process P2.

3. **No Preemption:** A resource can't be preempted from the process by another process, forcefully. For example, if a process P1 is using some resource R, then some other process P2 can't forcefully take that resource. If it is so, then what's the need for various scheduling algorithm. The process P2 can request for the resource R and can wait for that resource to be freed by the process P1.

4. **Circular Wait:** Circular wait is a condition when the first process is waiting for the resource held by the second process, the second process is waiting for the resource held by the third process, and so on. At last, the last process is waiting for the resource held by the first process. So, every process is waiting for each other to release the resource and no one is releasing their own resource. Everyone is waiting here for getting the resource. This is called a circular wait.

4) What is semaphore?

Ans:- A semaphore is an integer variable, shared among multiple processes. The main aim of using a semaphore is process synchronization and access control for a common resource in a concurrent environment.

It’s a synchronization tool that does not require busy waiting. Hence, the operating system does not waste the CPU cycles when a process can’t operate due to a lack of access to a resource.

5. What is a thread process and child process?

Ans:- A forked process is considered a child process whereas a threaded process is called a sibling. Forked process shares no resource like code, data, stack etc with the parent process whereas a threaded process can share code but has its own stack.

6. What are the File Locking systems using Semaphore?

Ans. The file that we lock but never store anything in, we call a semaphore file. The way we actually use a semaphore file is by opening it and locking it before we access some other real resource (like a counter file), and then not closing the semaphore file until we're done with the real resource.

Brief description

a. Function

Linux shell functions allow you to package a set of commands into one code block which can be called any number of times. This makes your shell programs small in length and increases re-usability of code.

b. Library Function

Library functions typically provide a richer set of features. For example, the fread() library function reads a number of elements of data of specified size from a file. While presenting this formatted data to the user, internally it will call the read() system call to actually read data from the file.

c. System Call

Ans. A system call is a request from computer software to an operating system's kernel. The Application Program Interface (API) connects the operating system's functions to user programs. It acts as a link between the operating system and a process, allowing user-level programs to request operating system services.

d. Kernal Call

Ans. A system call is a request from computer software to an operating system's kernel. The Application Program Interface (API) connects the operating system's functions to user programs. It acts as a link between the operating system and a process, allowing user-level programs to request operating system services.

e. Fork System Call

Ans. What is the fork () system call? The fork() system call is used in Unix-based OS to create a new process by duplicating the calling process. The new process is an exact copy of the parent process, with its own address space and memory.

f. Procedural Call

Ans. A procedure call transfers control from the call site in the caller to the start of the callee; on exit from the callee, control returns to the point in the caller that immediately follows its invocation. If the callee invokes other procedures, they return control in the same way.

g. System Program

Ans. System programs communicate and coordinate the activities and functions of hardware and software of a system and also controls the operations of the hardware. An operating system is one of the examples of system software.

h. Application Programming Interface(API).

Ans. API stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses.

8. Write down the uses of pipe command and write a c program that illustrates how to execute two commands concurrently with a command pipe.

Ans. Some uses of pipe command are as follows:-

- Listing all files and directories and give it as input to more command.

- Use sort and uniq command to sort a file and print unique values.
- Use head and tail to print lines in a particular range in a file.
- Use ls and find to list and print all lines matching a particular pattern in matching files.
- Use cat, grep, tee and wc command to read the particular entry from user and store in a file and print line count.

```
#include <stdio.h>

#include <unistd.h>

#include <sys/types.h>

#include <stdlib.h>

int main()
{
int pfds[2];
char buf[30]; if(pipe(pfds)==-1) {

perror("pipe failed"); exit(1);
}
if(!fork())

{
close(1); dup(pfds[1]; system (“ls –l”); }

else
{
printf("parent reading from pipe \n"); while(read(pfds[0],buf,80))
printf("%s \n" ,buf);
}
}
```

`

Assignment-B

1.What is Shell?

Ans. 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.

2. What is Shell Scripting?

Ans. Using a shell script is most useful for repetitive tasks that may be time consuming to execute by typing one line at a time. A few examples of applications shell scripts can be used for include: Automating the code compiling process. Running a program or creating a program environment.

Example of Shell Script : gzip is a command to create, extract or view .gz files, zip is a command used to create or extract zip file,
find is a command that is helpful to search for a file,
echo is a command displays a line of text.

3. Shell programs are stored in which file?

Ans. A shell script is a text file that contains a sequence of commands for a UNIX-based operating system. It is called a shell script because it combines a sequence of commands, that would otherwise have to be typed into the keyboard one at a time, into a single script.

4. What are the different types of Shells available?

Ans :- Different Types of Shells in Linux are as follows:-

- The Bourne Shell (sh)
- The GNU Bourne-Again Shell (bash)
- The C Shell (csh)
- The Korn Shell (ksh)
- The Z Shell (zsh)

5) What are the advantages of C Shell over Bourne Shell?

Ans. The Bourne shell family has a much richer (although more cryptic) programming language than the C shell family. While the C shell's Spartan programming interface is easier to learn, it can be quite a hindrance for most programming tasks, save for the simplest of scripts. Command editing.

6. What is the importance of writing Shell Scripts?

Ans. Using a shell script is most useful for repetitive tasks that may be time consuming to execute by typing one line at a time. A few examples of applications shell scripts can be used for include: Automating the code compiling process. Running a program or creating a program environment.

7. In a typical UNIX environment how many kernels and shells are available?

Ans. There can be multiple shells available, but there is only one kernel in the UNIX environment.

8.Is separate compiler required for executing a shell program?

Ans. A separate compiler is not required to execute a shell program. The shell itself interprets the command in the shell program and executes them.

9. How many shell scripts come with UNIX operating system?

Ans. There are approximately 280 shell scripts that come with the UNIX operating system. 10.When should shell programming/scripting not be used?

10) When should shell programming/scripting not be used?

Ans: Generally, shell programming/scripting should not be used in the below instances. When the task is very much complex like writing the entire payroll processing system. Where there is a high degree of productivity required.

11. Basis of shell program relies on what fact?

ANS- The basis of shell programming relies on the fact that the UNIX shell can accept commands not just only from the keyboard but also from a file.

12. How to accomplish any task via shell script?

ANS- Any task can be accomplished via shell script at the dollar ($) prompt and vice versa.

13. What are Shell Variables?

ANS- A shell variable is a special variable that is set by the shell and is required by the shell in order to function correctly. Some of these variables are environment variables whereas others are local variables.

Q-14 What are the default permissions of a file when it is created?

ANS- by default, when we create new files, they are given rw-rw-r– permissions. The r, w, and x signify the read, write, and execute permissions, respectively.

Q-15 What can be used to modify file permissions?

ANS- The chmod command enables you to change the permissions on a file.

Q-16 What are the two types of Shell Variables? Explain in brief.

ANS- In shell scripting there are tWO main types of variables are present. They are –

*Local Variables : A local variable is a special type of variable which has its scope only within a specific function or block of code. Local variables can override the same variable name in the larger scope.

*Global Variables or Environment Variables :A global variable is a variable with global scope. It is accessible throughout the program. Global variables are declared outside any block of code or function.

Q-17 How are shell variables stored? Explain with a simple example.

ANS- A shell variable is created with the following syntax: “variable_name=variable_value”. For example, the command “set COMPUTER_NAME=mercury” creates the shell variable named “COMPUTER_NAME” with a value of “mercury”. For values with spaces, quotation marks must be used.

Q-18 What is the lifespan of a variable inside a shell script?

ANS- A variable is inside the shell script only until the end of execution.

Q-19 What are positional parameters? Explain with an example.

ANS- A positional parameter is a parameter denoted by one or more digits, other than the single digit 0 . Positional parameters are assigned from the shell's arguments when it is invoked, and may be reassigned using the set builtin command.

Q-20 How to make variables as unchangeable?

ANS- Shell provides a way to mark variables as read-only by using the read-only command. After a variable is marked read-only, its value cannot be changed.

/bin/sh: NAME: This variable is read only.

Q-21 How variables can be wiped out?

ANS- Variables can be wiped out or erased using the unset command.

Example:

$ a =20

$ unset a

Upon using the above command the variable ‘a’ and its value 20 get erased from shell’s memory.

Q-22 Explain about file permissions.

ANS- There are 3 types of file permissions as shown below: Permissions Weight
r – read 4
w – write 2
x - execute 1

The above permissions are mainly assigned to owner, group and to others i.e. outside the group. Out of 9 characters first set of 3 characters decides/ indicates the permissions which are held by the owner of a file. The next

set of 3 characters indicates the permissions for the other users in the group to which the file owner belongs to.

And the last 3 sets of characters indicate the permissions for the users who are outside the group. Out of the 3 characters belonging to each set, the first character indicates the “read” permission, the second character indicates “write” permission and the last character indicates “execute” permission.

Example: $ chmod 744 file1
This will assign the permission rwxr–r–to file1.

Q-23 What is a file system?

ANS-A file system is a process of managing how and where data on a storage disk, which is also referred to as file management or FS. It is a logical disk component that compresses files separated into groups, which is known as directories. It is abstract to a human user and related to a computer; hence, it manages a disk's internal operations. Files and additional directories can be in the directories. Although there are various file systems with Windows, NTFS is the most common in modern times. It would be impossible for a file with the same name to exist and also impossible to remove installed programs and recover specific files without file management, as well as files would have no organization without a file structure. The file system enables you to view a file in the current directory as files are often managed in a hierarchy.

Q-24 What are the different blocks of a file system? Explain in brief.

ANS-

*Super Block: This block mainly tells about a state of the file system like how big it is, maximum how many files can be accommodated, etc.

*Boot Block: This represents the beginning of a file system. It contains the bootstrap loader program, which gets executed when we boot the host machine.

*Inode Table: As we know all the entities in a UNIX are treated as files. So, the information related to these files is stored in an Inode table.

*Data Block: This block contains the actual file contents.

Q-25 What are the three modes of operation of vi editor? Explain in brief.

ANS- The three modes of operation of vi editors are:-

Command Mode: In this mode, all the keys pressed by a user are interpreted as editor commands.

Insert Mode: This mode allows for the insertion of a new text and editing of an existing text etc.

The ex-command Mode: This mode allows a user to enter the commands at a command line.

Q-26 What is the alternative command available to echo and what does it do?

ANS- tput is an alternative command to echo.
Using this, we can control the way in which the output is displayed on the screen.

Q-27 How to find out the number of arguments passed to the script?

ANS- The number of arguments passed to the script can be found by the below command. echo $ #

Q-28 What are control instructions and how many types of control instructions are available in a shell? Explain in brief.

ANS- Control Instructions are the ones, which enable us to specify the order in which the various instructions in a program/script are to be executed by the computer. Basically, they determine a flow of control in a program.

There are 4 types of control instructions that are available in a shell.

Sequence Control Instruction: This ensures that the instructions are executed in the same order in which they appear in the program.

Selection or Decision Control Instruction: It allows the computer to take the decision as to which instruction is to be executed next.

Repetition or Loop Control Instruction: It helps a computer to execute a group of statements repeatedly.

Case-Control Instruction: This is used when we need to select from several alternatives.

Q-29 What are the three different security provisions provided by UNIX for a file or data?

ANS- Three different security provisions provided by UNIX for a file or data are:

It provides a unique user id and password to the user, so that unknown or unauthorized person should not be able to access it.

At the file level, it provides security by providing read, write & execute permissions for accessing the files.

Lastly, it provides security using file encryption. This method allows encoding a file in an unreadable format. Even if someone succeeds in opening a file, but they cannot read its contents until and unless it is decrypted

Q-30 What is IFS?

ANS- IFS stands for Internal Field Separator. And it is one of the system variables. By default, its value is space, tab, and a new line. It signifies that in a line where one field or word ends and another begins.

Q31 :What are Metacharacters in a shell? Explain with some examples.

Ans: Metacharacters are: pipe (|), ampersand (&), semicolon (;), less-than sign (<), greater- than sign (>), left parenthesis (( ), right parenthesis ( )), dollar sign ($), backquote (`), backslash (\ ), right quote ('), double quotation marks ("), newline character, space character, and tab character.

Q32: How to execute multiple scripts? Explain with an example

Answer: In a shell, we can easily execute multiple scripts i.e. one script can be called from the other. We need to mention the name of a script to be called when we want to invoke it.

Example: In the below program/script upon executing the first two echo statements of script1, shell script executes script2. Once after executing script2, the control comes back to script1 which executes a pwd command and then terminates.

Q33:Which command needs to be used to know how long the system has been running?

Ans: The uptime command
Current time. Up (the current status of the system) Total time the system has been running

Q34 How to find the current shell which you are using?

Ans: Using the echo Command

1. echo $SHELL. The $SHELL variable contains the name of the default shell. We can display its value: $ echo $SHELL /bin/bash. ...

2. echo $0. We can also use the echo command with the $0 variable: $ echo $0 bash. This approach works well on the command line but not from within a script.

Q35: How to find all the available shells in your system?

Ans: There are various methods to list all the shells which are installed in our Linux system.

Method 1: Using cat command
Example:- cat /etc/shells

Method 2: Use of grep command with Regular Expressions.
Example:- grep '^[^#]' /etc/shells

Q36: How to read keyboard inputs in shell scripts?

Ans: read -p "Prompt" variable1 variable2 variableN Multiple Input (number.sh)

```
#!/bin/bash
# read three numbers and assigned them to 3 vars
read -p "Enter number one : " n1
read -p "Enter number two : " n2
read -p "Enter number three : " n3
# display back 3 numbers - punched by user.
echo "Number1 - $n1"
echo "Number2 - $n2"
echo "Number3 - $n3”
```

Q37:How many fields are present in a crontab file and what does each field specify?

Ans:A crontab file consists of commands, one per line, that execute automatically at the time specified by the first five fields at the beginning of each command line. These first five fields, described in the following table, are separated by spaces. They indicate when the command will be executed

Q38:What are the two files of crontab command?

Ans:There are two different crontab files, each located in a different area. The two are: The user crontab file located in /var/spool/cron/crontabs. The system crontab file located in /etc/cron.

Q39: What command needs to be used to take the backup?

Ans:Windows command line allows you to back up files and directories using the xcopy command. Xcopy has different switches using which we can specify the conditions on what files need to be copied. You can use a USB drive or an external hard disk as the backup location.

Q40:What are the different commands available to check the disk usage?

Ans:Two related commands that every system administrator runs frequently are df and du . While du reports files' and directories' disk

usage, df reports how much disk space your filesystem is using. The df command displays the amount of disk space available on the filesystem with each file name's argument.

Q41: What are the different communication commands available in Unix/Shell?

Ans:Communications Commands

biff – Mail notification.
comstat – Incoming mail daemon. Mail.rc – Configuration file for mail.
wall – Send message to all users.
write – Send messages to specific user. mesg – allow/disallow write or talk. ate – vt100 terminal emulator.

Q42:How to find out the total disk space used by as specific user, say for example username is John?

Ans: The total disk space used by John can be found out as : du –s/home/John

Q43: What is Shebang in a shell script?

In computing, a shebang is the character sequence consisting of the characters number sign and exclamation mark (#!) at the beginning of a script. It is also called sharp-exclamation, sha- bang, hashbang, pound-bang, or hash-pling.

Q44: What is the command to be used to display the shell's environment variables?

Ans: Using Environment Variables in Bash Shell

To list all the environment variables, use the command " env " (or " printenv "). You could also use " set " to list all the variables, including all local variables.

Q45: How to debug the problems encountered in shell script/ program?

Ans:When a script does not work properly, we need to determine the location of the problem. The UNIX/Linux shells provide a debugging

mode. Run the entire script in debug mode or just a portion of the script. To run an entire script in debug mode, add -x after the #!/bin/[shell] on the first line:

For Example :
#!/bin/sh -x

Q46: How to open a read-only file in Unix/shell?

Ans: Anything that opens a file, can open a read only file: cat, more, less, awk, pg, et all. The chmod command you put does not make a file read only, it makes it read/write for the user/owner.

Q47:How can the contents of a file inside jar be read without extracting in a shell script?

Ans: Answer: The contents of the file inside a jar can be read without extracting in a shell script as shown below.

tar –tvf <File Name>.tar

Q48: What is the difference between diff and cmp commands?

cmp' and 'diff' both command are used to list the differences, the difference between both the command is that 'cmp' is used to find the difference between files whereas 'diff' is used to find the difference between directories.

cmp will list the line and column number that are different between two files.

cmp – Basically it compares two files byte by byte and displays the very first mismatch.

Q49: Explain in brief about sed command with an example.

Ans:- SED command in UNIX stands for stream editor and it can perform lots of functions on file like searching, find and replace, insertion or deletion. Though most common use of SED command in UNIX is for substitution or for find and replace. By using SED you can edit files even without opening them, which is much quicker way to find and replace something in file, than first opening that file in VI Editor and then changing it.

- SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution).

- SED command in unix supports regular expression which allows it perform complex pattern matching.

Q50: Explain in brief about awk command with an example.

Ans:-The AWK command has a simple syntax that consists of a series of patterns and actions. Patterns are used to match specific lines of input, while actions are used to specify what should be done with those lines. Here is an example of basic syntax of AWK command – **awk 'pattern { action }' input_file**