

Scheme of Evaluation
Internal Assessment Test 1 – Nov 2021

Sub:	Unix Programming						Code:	18CS56	
Date:	12/11/2021	Duration:	90mins	Max Marks:	50	Sem:	V	Branch:	ISE

Note: Answer Any five full questions.

Question #		Description	Marks Distribution		Max Marks
1	a)	Explain with neat diagram the architecture of Unix Operating System. Diagram Explanation	3M 3M	6M	10M
	b)	With suitable example, bring out the differences between absolute and relative path names. Any two differences with example	2*2M	4M	
2	a)	List and explain the silent features of Unix Operating system. Any 6 features	1M*6	6M	10M
	b)	Compare and contrast between internal and external commands in Unix Operating system. Any two differences with example	2M*2	4M	

3	a)	<p>What is parent child relationship? With the help of neat diagram, explain Unix file system.</p> <p>Diagram</p> <p>Explanation</p>	2M 3M	5M	
3	b)	<p>Assume the current to be 5_sem, which has a subdirectory unix(contains file.txt) and a directory called “data”). Write the commands to perform following</p> <ol style="list-style-type: none"> a) Go to the subdirectory called “unix” b) Copy the file “file.txt” into “file.txt.copy” c) Rename the file “file.txt.copy” to file2. d) Move the file “file2” into the “data” subdirectory. e) Create a new subdirectory called “new” in the unix directory. f) Move the file “file2” in the “data” directory into the new directory. g) Move the file “file2” in the “new” directory back into the “unix” and change the name to “file.old” h) Delete the file “file.old” <p>Remove the “new” subdirectory.</p> <p>Each command</p>	0.5*10	5M	10M
4	a)	<p>With example, explain the following commands printf b)who c) man d) pwd e) date f)mkdir</p> <p>Each command explanation with example</p>	1M*6	6M	
4	b)	<p>Write the output of following commands</p> <ol style="list-style-type: none"> a) date+ “ Date is: %a%h%Y” b) cal 8 1947 c) echo \$HOME d) \$bc obase=2 14 <p>[ctrl+d]</p> <p>each command output</p>	1*4M	4M	10M

5	a)	<p>Explain the different types of files supported in Unix.</p> <p>Regular Files</p> <p>Directory Files</p> <p>Device Files</p>	<p>2M</p> <p>2M</p> <p>1M</p>	5M	
5	b)	<p>Draw the tree structure of the file system created by the following commands (assume you are in the directory /usr/office)</p> <p>mkdir left</p> <p>mkdir middle</p> <p>mkdir right</p> <p>cd left</p> <p>mkdir left middle right</p> <p>cd ../middle</p> <p>mkdir dir1 dir2 /usr/office/right/dir3</p> <p>Why is not possible to issue the command rmdir /usr/office/right?</p> <p>Tree drawing</p>	1M*5	5M	10M
6	a)	<p>Explain any five file related commands with example</p> <p>cat,mv,cp,od,wc,rm</p> <p>Explanation with example</p>	2M*5	10M	10M

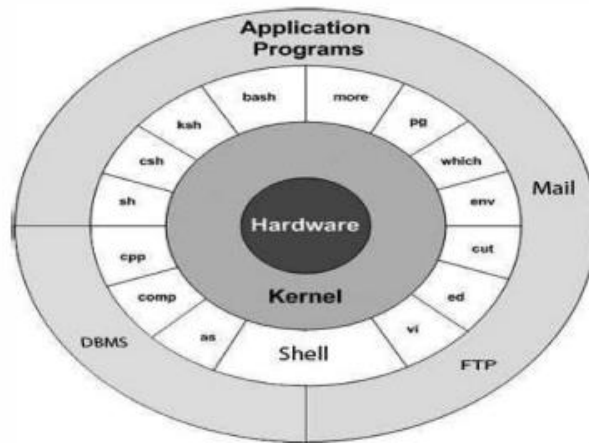
Scheme Of Evaluation
Internal Assessment Test 1 – Nov 2021

Sub:	Unix Programming						Code:	18CS56	
Date:	12/11/2021	Duration:	90mins	Max Marks:	50	Sem:	V	Branch:	ISE

Note: Answer Any full five questions

Q1 a) Explain with neat diagram the architecture of Unix Operating System.

UNIX ARCHITECTURE



Kernel: is the core of operating system. A collection of routines mostly written in C.

It is loaded into memory when the system is booted and communicates directly with the hardware. The kernel manages system memory, processes, decides priorities.

Shell: interface between Kernel and User. It functions as command interpreter i.e it receives and interprets the command from user and interacts with the hardware. There is only one kernel running on the system, there could be several shells in action- one for each user who is logged in.

Files and Process: file is an array of bytes and it contain virtually anything. Unix considers even the directories and devices as members of file system. The dominant file type is text and behavior of system is mainly controlled by text files.

The second entity is the process, which is the name given to a file when it is executed as a program. Process is simply a time image of an executable file.

1.1 System Calls: Though there are thousands of commands in the unix system, they all use a handful of functions called system calls. User programs that need to access the hardware use the services of the kernel, which performs the job on users behalf. These programs access the kernel through a set of functions called system calls.

Ex: open()— system call to access both file and device. Write()—system call to write a file.

Q. 1 b) With suitable example, bring out the differences between absolute and relative path names.

RELATIVE PATHNAMES

- Pathnames that don't begin with / specify locations relative to your current working directory.
- Uses either the current or parent directory as reference and specifies path relative to it.
- A relative pathname uses one of these cryptic symbols.
 - . (a single dot)→ this represents the current directory.
 - .. (two dots)→this represents the parent directory

Ex .1: Assume the current directory is /home/kumar/progs/data/text, using cd .. will move one level up

```
Spwd
/home/kumar/progs/data/text
$ cd ..
Spwd
/home/kumar/progs/data
```

ABSOLUTE PATHNAMES:

- If the first character of a pathname is / the files location must be determined with respect to root(/) . Such a pathname is called absolute pathname.

`cat /home/kumar`

- When you have more than one / in a pathname for such / you have to descend one level in the file system. Thus Kumar is one level below home and two levels below root.
- When you specify a file y using frontslashes to demarcate the various levels,you have a mechanism of identifying a file uniquely.No two files in a UNIX system can have same absolute pathnames.
- When you specify the date command, the system has to locate the file date from a list of directories specified in the PATH variable and then execute it.
- However if you know the location of a command in prior, for example date is usually located in /bin or /usr/bin . Use absolute pathname i,e precede its name with complete path

`$/bin/date`

Q.2 a) List and explain the silent features of Unix Operating system.

FEATURES OF UNIX

Several features of UNIX have made it popular. Some of them are:

- **Portable:** UNIX can be installed on many hardware platforms. Its widespread use can be traced to the decision to develop it using the C language. Because C programs are easily moved from one hardware environment to another, it is relatively simple to port it to different environments.
- **Multuser:** The UNIX design allows multiple users to concurrently share hardware and software
- **Multitasking:** UNIX allows a user to run more than one program at a time. In fact more than one program can be running in the background while a user is working foreground.

- **Networking:** While UNIX was developed to be an interactive, multiuser, multitasking system, networking is also incorporated into the heart of the operating system. Access to another system uses a standard communications protocol known as Transmission Control Protocol/Internet Protocol (TCP/IP).
- **Organized File System:** UNIX has a very organized file and directory system that allows users to organize and maintain files.
- **Device Independence:** UNIX treats input/output devices like ordinary files. Input or output to a program can be from any device or file. The source or destination for file input and output is easily controlled through a UNIX design feature called redirection.
- **Utilities:** UNIX provides a rich library of utilities that can be used to increase user productivity.
- **Services:** UNIX also includes the support utilities for system administration and control.

Q.2 b) Compare and contrast between internal and external commands in Unix Operating system.

UNIX commands are classified into two types

- Internal Commands - Ex: echo
- External Commands - Ex: ls, cat

Internal Command:

Internal commands are something which is built into the shell. For the shell built in commands, the execution speed is really high. It is because no process needs to be spawned for executing it.

- For example, when using the "cd" command, no process is created. The current directory simply gets changed on executing it.

External Command:

External commands are not built into the shell. These are executable present in a separate file. When an external command has to be executed, a new process has to be spawned and the command gets executed.

- For example, when you execute the "cat" command, which usually is at /usr/bin, the executable /usr/bin/cat gets executed.

type command:

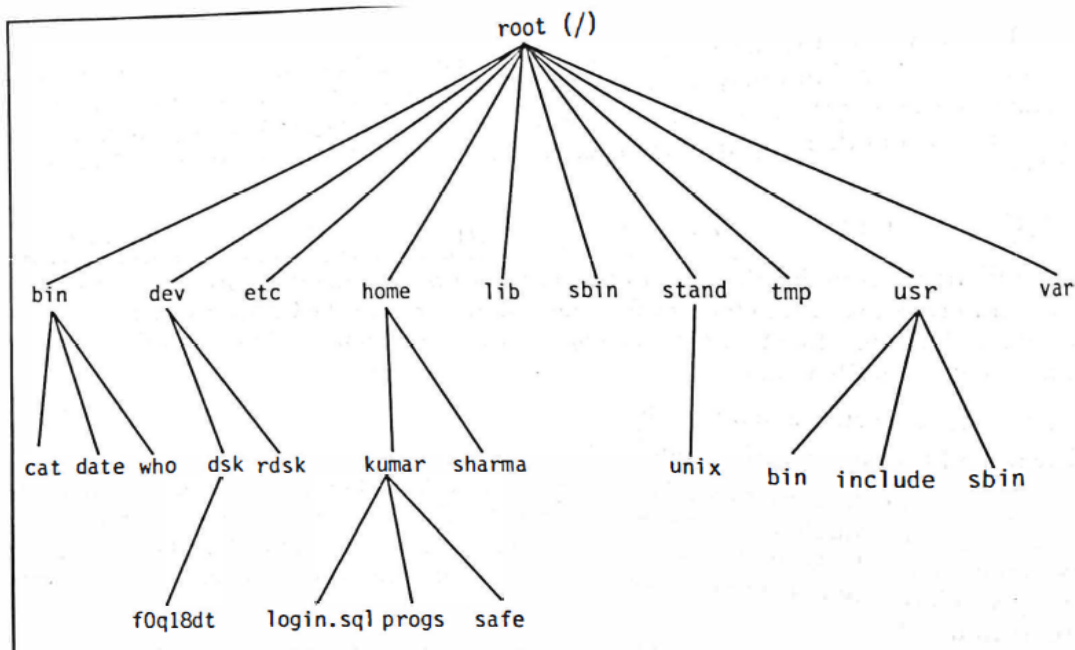
```
$ type cd
cd is a shell builtin
$ type cat
cat is /bin/cat
```

For the internal commands, the type command will clearly say its shell built-in, however for the external commands, it gives the path of the command from where it is executed.

Q. 3a) What is parent child relationship? With the help of neat diagram, explain Unix file system

PARENT CHILD RELATIONSHIP/ ORGANIZATION OF FILES

All files in UNIX are related to one another. The file system in unix is a collection of all ordinary, directory and device files and organized in a hierarchical structure as shown in below fig.



The implicit feature of every UNIX file system is that there is a top which serves as reference point for all files. This top is called **root** & is represented by a / (front slash). Root is actually a directory. The root

directory has a number of sub directories under it. These sub directories in turn have more sub directories and others files under them.

For instance bin and usr are two directories directly under root, while a second bin and kumar are sub directories under usr.

Every file apart from root must have a parent. Thus the home directory is the parent of kumar, while / is the parent of home and grandparent of kumar. If you create a file login.sql under the kumar directory, then kumar will be the parent of this file.

The first group contains the files that are made available during system installation

- **/bin and /usr/bin:** these are the directories where all the commonly used UNIX commands are found.
- **/sbin and /usr/sbin:** If there's a command that you can't execute but the system administrator can execute, it would be probably in one of these directories.
- **/etc:** this directory contains the configuration files of the system. You can change a very important aspect of system functioning by editing a text file in this directory. Your login name and password are stored in files `/etc/passwd` and `etc/shadow`
- **/dev:** This directory contains all device files. These files don't occupy space on disk. there could be more sub directories like `pts`, `dsk` and `rdsk` in this directory
- **/lib and /usr/lib:** Contains shared library files and sometimes other kernel-related files.
- **/usr and /include:** contains the standard header files used by C programs. The statement `#include<stdio.h>` used in most C programs refers to the file `stdio.h` in this directory.
- **/usr/share/man:** this is where the man pages are stored. There are separate subdirectories here (like `man1`, `man2` etc) that contains the pages for each section. For instance, the man page of `ls` can be found in `/usr/share/man/man1`

User also work with their own files, they write programs, send and receive mail and also create temporary files. These files are available in the second group shown below

- **/tmp:** the directory where users are allowed to create temporary files. These files are wiped away regularly by the system
- **/var:** The variable part of the file system. Contains all your print jobs and your outgoing and incoming mail.
- **/home:** On many systems users are housed here. Kumar would have his home directory in `/home/kumar`

Q. 3 b) Assume the current to be `5_sem`, which has a subdirectory `unix` (contains `file.txt`) and a directory called "data"). Write the commands to perform following

- a) Go to the subdirectory called "unix"
- b) Copy the file "file.txt" into "file.txt.copy"
- c) Rename the file "file.txt.copy" to `file2`.
- d) Move the file "file2" into the "data" subdirectory.
- e) Create a new subdirectory called "new" in the `unix` directory.
- f) Move the file "file2" in the "data" directory into the new directory.
- g) Move the file "file2" in the "new" directory back into the "unix" and change the name to "file.old"
- h) Delete the file "file.old"

Remove the "new" subdirectory.

- a) \$cd unix
- b) \$cp file.txt file.txt.copy
- c) \$mv file.txt.copy file2
- d) \$mv file2 data
- e) \$mkdir new
- f) \$mv data/file2 new
- g) \$mv new/file2 .
\$mv file2 file.old
- h) \$rm file.old
- i) \$rmdir new

Q. 4 a) With example, explain the following commands

a) printf b)who c) man d) pwd e) date f)mkdir

- a) *printf* command in Linux is used to display the given string, number or any other format specifier on the terminal window

Input: printf "%s\n" "Hello, World! \n" "From Geeks For Geeks\n"

Output: Hello, World! \n
From Geeks For Geeks\n

b) who - show who is logged on

\$ who

amrood ttyp0 Oct 8 14:10 (limbo)

bablu ttyp2 Oct 4 09:08 (calliope)

qadir ttyp4 Oct 8 12:09 (dent)

c) man

man command in Linux is used to display the user manual of any command that we can run on the terminal. It provides a detailed view of the command which includes NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUES, ERRORS, FILES, VERSIONS, EXAMPLES, AUTHORS and SEE ALSO.

\$man [OPTION]... [COMMAND NAME]..

\$ man printf

Shows the manual pages of printf

d)pwd

To print the current working directory run the pwd command. The full path of the current working directory will be printed to standard output.

```
$pwd  
/home/Shilpa
```

d) date

date command is used to display the system date and time. date command is also used to set date and time of the system. By default the date command displays the date in the time zone on which unix/linux operating system is configured

```
$date
```

Output:

```
Tue Oct 10 22:55:01 PDT 2017
```

e)mkdir

mkdir: "making directory".

- **mkdir** is used to create directories on a file system.
- If the specified *DIRECTORY* does not already exist, **mkdir** creates it.
- More than one *DIRECTORY* may be specified when calling **mkdir**.

Smkdir gmit

gmit directory is created under present working directory.

Assume that pwd is /home/kumar , then gmit directory is created under kumar directory.

Q. 4b) Write the output of following commands

a) date+ “ Date is: %a%h%Y”

b) cal 8 1947

c) echo \$HOME

d) \$bc

```
obase=2
```

```
14
```

```
[ctrl+d]
```

a) %a: Displays the abbreviated name for weekdays (Sun to Sat).

%h: Displays abbreviated month name (Jan to Dec).

%Y: Display four-digit year.

```
Date is Mon Nov 2021
```

- b) Displays the calendar of Aug 1947
- c) /home/shilpa
- d) 1110

Q. 5a) Explain the different types of files supported in Unix.

BASIC FILE TYPES/CATEGORIES FILES

File is a collection of records. So, files are divided into three categories

a. Ordinary file

b. Directory file

c. Device file

The UNIX file system contains several different types of files:

a. Ordinary Files or regular files

It contains only data as a stream of characters.

An ordinary files itself divided into 2 types

Text file: contains only printable characters, and you can often view the contents and make sense out of them. All C and Java files are example of text file. A text file contains lines of characters where every line is terminated with the newline character, also known as **linefeed** (LF) when you press Enter while inserting text, the LF character is appended to every line. You won't see this character normally, but there is command (od) which can make it visible.

Binary file: it contains both printable and unprintable characters that cover the entire ASCII range(0 to 255).most UNIX commands are example of binary files.

b. Directory files

i. Contains no data, but keeps some details of the files and subdirectories that it contains.

ii. A directory file contains an entry for every file and sub directory that it houses. Each entry has two components

- The filename
- A unique Identification number for the file or directory(called the inode number)

iii. A directory contains the filename but not the contents of file.

iv. When you create or remove a file the kernel automatically updates its corresponding directory by adding or removing the enter i.e inode number associated with that file.

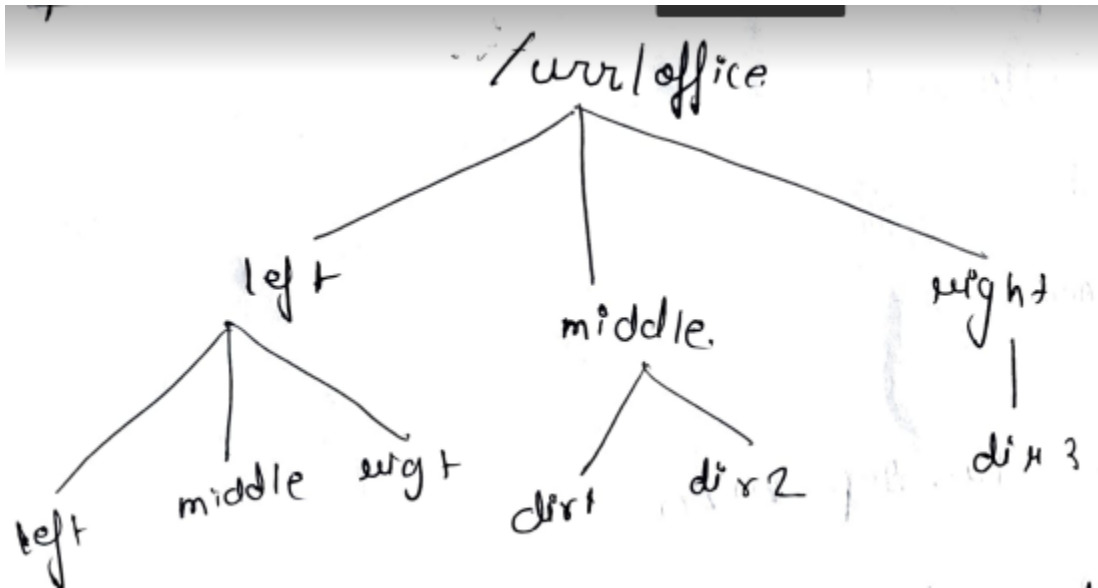
c. Device files

- i. Used to represent a real physical device such as a printer, tape drive or terminal, used for Input/Output (I/O) operations
- ii. Unix considers any device attached to the system to be a file - including your terminal:
- iii. By default, a command treats your terminal as the standard input file (stdin) from which to read its input
- iv. Your terminal is also treated as the standard output file (stdout) to which a command's output is sent.

Q.5 b) Draw the tree structure of the file system created by the following commands (assume you are in the directory /usr/office)

```
mkdir left
mkdir middle
mkdir right
cd left
mkdir left middle right
cd ../middle
mkdir dir1 dir2 /usr/office/right/dir3
```

Why is not possible to issue the command rmdir /usr/office/right?



As /usr/office/right is not empty, can't give command rmdir /usr/office/right

Q. 6 a) Explain any five file related commands with example

Cat command : Displaying and creating files

cat is one of the most well known commands of UNIX system.

Cat is useful for creating a file .

Its mainly used to display the contents of a small file on the terminal.

- **Using cat to create a file:**

Enter the command cat, followed by >(right chevron) character and the filename.

Example: take a filename named foo

```
$ cat > foo
```

> Symbol following command means that the output goes to filename following it.

[ctrl+d] /* to terminate or to signify end of the input.

```
$
```

- **Using cat to display a file**

Enter the cat command followed by filename

```
$ cat foo
```

Symbol following command means that the output goes to filename following it.

cp: copying a file

- cp command copies a file or a group of files.it creates an exact image of the file on the disk with the different name.
- The syntax requires atleast two filenames to be specified in the command line.
- When both are ordinary files, the first is copied to second file.

```
cp source file destination file
```

```
cp chap01 unit1
```

if destination file i.e unit1 does not exist, first it will be created before copying.if not it will be simply overwritten without any warning.

- Copying a file to another directory

ex: assume there is a file named chap01 and it has to be copied to progs directory

rm : deleting files

The rm command deletes one or more files.

Ex 1: The following command deletes three files chap01, chap02, chap03.

```
$ rm chap01 chap02 chap03
```

Ex 2: to delete files named chap01 and chap02 under progs directory

```
$ rm progs/chap01 progs/chap02
```

Ex 3: to remove all file

```
$ rm*
```

mv: RENAMING FILES.

The mv command renames or moves files. It has two distinct functions:

- h. It renames a file or directory
- i. it moves a group of files to a different directory

To rename a file chap01 to man01

```
$ mv chap01 man01
```

mv replace the filename in the existing directory entry with the new name.

No additional space is consumed on disk during renaming.

To rename a directory:

```
$ mv pts perdir
```

pts directory is renamed as perdir

wc command: COUNTING LINES,WORDS,CHARACTERS

wc command takes one or more filenames as arguments and displays four columnar output.

First we will create a file named infile

```
$ cat > infile
```

```
I am the wc command
```

```
I count characters,words and lines
```

```
[ctrl+D]
```

```
$wc infile
```

```
2    10    55   infile
```

od Command: DISPLAYING DATA IN OCTAL.

```
$ cat odfile
```

```
White space includes a
```

```
The ^G character rings a bell
```

```
$ od -b odfile
```

The -b option displays the octal values for each character.

```
000000 127 150 151 164 145 040 163 160 141 143 145 040 151 156 143 154
```

```
000000 165 144 145 163 040 141 040 011 012 124 150 145 040 007 040 143
```

Each line displays 16 bytes of data in octal , preceded by the offset in the file of the first byte in the line.

Sod -bc odfile

The -b and -c option combined

Each line is now replaced with two.

The octal values are shown in first line and printable characters and escape sequences are shown in second line

```
000000 127 150 151 164 145 040 163 160 141 143 145 040 151
      W  h  i  t  e          s  p  a  c  e          i
      156 143 154
      n  c  l
000000 165 144 145 163 040 141 040 011 012 124 150 145 040
      u  d  e  s          a          \t  \n  T  h  e
      007 040 143
      007          c
```

The octal equivalent of characters are displayed ex for W- 127, i-151, \t (tab)-011, \n(newline)-012
^G(Bell character)- 007