

Q.2 a. Explain the principal components: Kernel and Shell, of the UNIX operating system.

Answer: Refer Page No. 22 from Textbook

b. Explain absolute and relative pathnames with the help of examples.

Answer: Refer Page No. 73-75 from Textbook

Q.3 a. What is the function of ls command? List the different options and what is the use of -a, -l, -d and -R in it.

Answer:

ls command is used to print content of directory. Following are the different options and their uses:

- a list hidden files
- d list the name of the current directory
- F show directories with a trailing '/'
executable files with a trailing '*'
- g show group ownership of file in long listing
- i print the inode number of each file
- l long listing giving details about files and directories
- R list all subdirectories encountered
- t sort by time modified instead of name

b. Explain the three modes of operation in vi editor. (8)

Answer:

Modes of Operation in vi editor

There are three modes of operation in vi: Command Mode, Input Mode, and Line Mode:

(A) Command Mode: When vi starts up, it is in Command Mode. This mode is where vi interprets any characters we type as commands and thus does not display them in the xterm window. This mode allows us to move through a file, and to delete, copy, or paste a piece of text. To enter into Command Mode from any other mode, it suffices to press the [Esc] key. If we press [Esc] when we are already in Command Mode, then vi will beep or flash the screen.

(B) Input Modet: In Input Mode, vi accepts keystrokes as text and displays the text as it is entered from the keyboard. vi must be in Input Mode before we can insert text into a file. To enter into Input Mode, we need to put vi into Command Mode and type the key [i].

(C) Line Mode: Line Mode is invoked by typing a colon [:] or a slash [/] while vi is in

Command Mode. The cursor will jump to the last line of the screen and vi will wait for a command.

Q.4 a. List the steps that the shell undertakes in order to execute a given command.

Answer: Refer Page No. 146 (summarized points) from Textbook

b. What are pipes in Unix and what is their use ?

(8)

Answer: Refer Page No. 161-164 from Textbook

Q.5 a. What is hard link and soft link? How can they be created? (3+3+2)

Answer:

Hard links: a hard link is a pointer to the file's i-node. For example, suppose that we have a file a-file.txt that contains the string "The file a-file.txt":

```
% cat a-file.txt
```

The file a-file.txt

Soft links (symbolic links): a soft link, also called symbolic link, is a file that contains the name of another file. We can then access the contents of the other file through that name. That is, a symbolic link is like a pointer to the pointer to the file's contents. For instance, supposed that in the previous example, we had used the -s option of the ln to create a soft link:

```
% ln -s a-file.txt b-file.txt
```

b. Which command is used to change permission associated to File/Directories? List and explain methods to change permission of File/Directories.

Answer:

There are three types of permissions:

- r read** the file or directory
- w write** to the file or directory
- x execute** the file or search the directory

Each of these permissions can be set for any one of three types of user:

- u the user** who owns the file (usually you)

g members of the **group** to which the owner belongs

o all **other** users

The access permissions for all three types of user can be given as a string of nine characters:

user	group	others
r w x	r w x	r w x

Permission	File	Directory
r read	read a file	list files in ...
w write	write a file	create file in ... rename file in ... delete file ...
x execute	execute a shell script	read a file in ... write to a file in ... execute a file in ... execute a shell script in ...

Setting access permissions numerically

There is a shorthand way of setting permissions by using octal numbers. Read permission is given the value 4, write permission the value 2 and execute permission 1.

r	w	x
4	2	1

These values are added together for any one-user category:

1	=	execute only
2	=	write only
3	=	write and execute (1+2)
4	=	read only

- 5 = read and execute (4+1)
- 6 = read and write (4+2)
- 7 = read and write and execute (4+2+1)

So access permissions can be expressed as three digits. For example:

	user	group	others
chmod 640 file1	rw-	r--	---
chmod 754 file1	rx	r-x	r--
chmod 664 file1	rw-	rw-	r--

Q.6 a. What is the difference between a wild card and a regular expression?

Answer:

A wildcard is a generic term referring to something that can be substituted for all possibilities. This is like a joker being a wildcard in poker. In computer terms, usually a simple "wildcard" is just a * that can match one or more characters, and possibly a ? that can match any single character.

A regular expression certainly includes such functionality, but is a much more powerful pattern matcher that gives you the ability to restrict the type of characters and give more flexible specification on how many characters to match. Picture matching a phone number. A wildcard at best could do something like

???-???-????

while a regex can do

`\d{3}-\d{3}-\d{3}`

which will match any US phone number in that particular format, ensuring each character in the phone number is 0-9 (that's what the `\d` is).

But parentheses in regexes can be used for another purpose too. You can match sections of the pattern, which you can use later. Let's say you want the area code.

`(\d{3})-\d{3}-\d{3}`

will not only match the phone number, but it will also store the area code in the first "group". A regex parser, in a language like Perl or Java, will then let you retrieve the group, to do what you want with it.

b. What is the difference between Interval Regular Expression (IRE) and Tagged Regular Expression (TRE)?

Answer: Refer Page No. 263-266 from Textbook

Q.7 a. What is the exit status of a command? What is its normal value and which shell parameter stores its value?

Answer:

The exit status is an integer that represents the success or failure of a command. It has the value 0 when the command executes successfully and is stored in the parameter \$?

b. Explain the test command. (8)

Answer: Refer Page No. 278-283 from Textbook

Q.8 a. Explain in detail the simple awk filtering. (6)

Answer:

a filter action is the most efficient way to limit its scope. In AWK scripts, the action specified by such a conditional filter occurs only if the specified pattern matches the record in question.

The format for a conditional filter rule is as follows:

```
pattern { action }
```

The action here is a series of statements just like any other filter rule. The pattern can be blank (in which case it matches every record), or it can contain any combination of regular expressions or relational expressions. These two types of expressions are briefly explained in the following sections.

b. Explain with an example how if statement can be used in an awk command.

Answer: Refer Page No. 395 from Textbook

c. What is the difference between print and print \$0? (4)

Answer:

File 1 - firstfile

```
#!/usr/bin/perl -w  
use strict;
```

```
print $0 . "\n";  
print __FILE__ . "\n";
```

```
do 'filetwo';
```

File 2 - filetwo

```
#!/usr/bin/perl -w
```

```
use strict;
```

```
print $0 . "\n";
```

```
print __FILE__ . "\n";
```

Q.9 a. Explain the foreach loop in perl with the help of an example.

Answer: Refer Page No. 412 from Textbook

b. Explain the following:

(4+4)

(i) chop function

(ii) substr function

Answer:

(i) chop

This function removes the last character of a string and returns that character. If given a list of arguments, the operation is performed on each one and the last character chopped is returned.

chop VARIABLE

chop(LIST)

chop

(ii) Refer Page No. 406 from Textbook

TEXT BOOK

- I. UNIX Concepts and Applications, 4th Edition, Sumitabha Das, Tata McGraw Hill, 2008