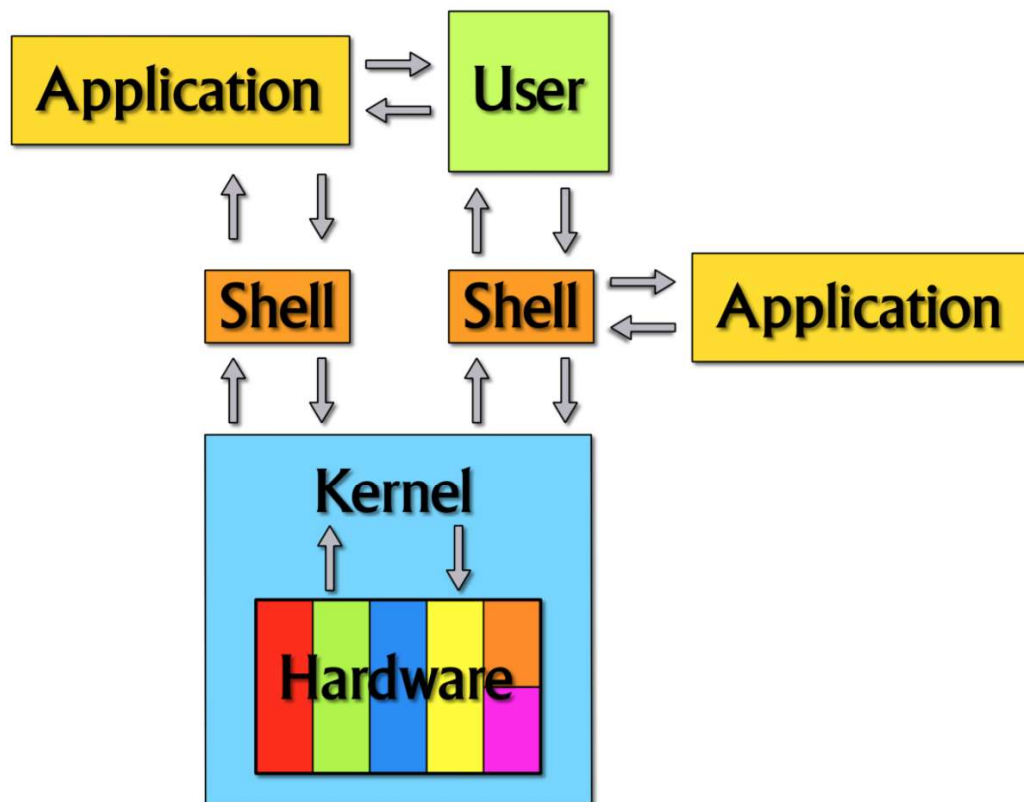


INTRODUCTION:-

SHELL:-

A shell is a program that takes commands typed by the user and calls the operating system to run those commands. A shell is a program that acts as the interface between the user and the Linux system, allowing the user to enter commands for the operating system to execute.

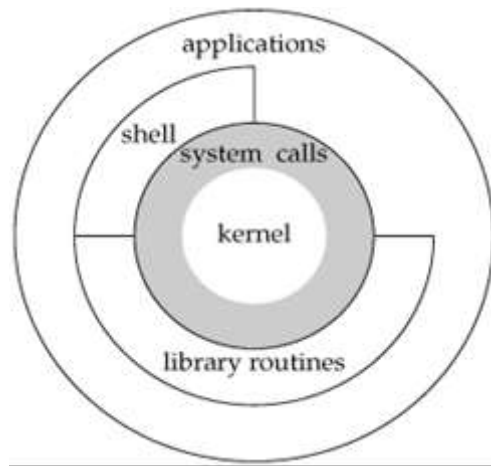
Typical operations performed by shell scripts include file manipulation, program execution, and printing text.



Shell as an interface between the O.S & User

USES OF A SHELL:-

1. Users can use shell scripts to automate administrative tasks.
2. Encapsulate complex configuration details.
3. Get at the full power of the operating system.
4. The ability to combine commands allows you to create new commands



Linux kernel architecture

TYPES OF SHELL:-

1. **Bourne Shell** :-A Bourne shell (sh) is a UNIX shell or command processor that is used for scripting. It was developed in 1977 by Stephen Bourne of AT&T and introduced in UNIX Version 7, replacing the Mashey shell (sh).

The Bourne shell is also known by its executable program name, "sh" and the dollar symbol, "\$," which is used with command prompt

```

Terminal
-rwxr-xr-x 1 bin      18296 Jun  8 1979 fsck
-rwxr-xr-x 1 bin      1458 Jun  8 1979 getty
-rw-r--r-- 1 root       49 Jun  8 1979 group
-rwxr-xr-x 1 bin     2482 Jun  8 1979 init
-rwxr-xr-x 1 bin     8484 Jun  8 1979 mkfs
-rwxr-xr-x 1 bin     3642 Jun  8 1979 mknod
-rwxr-xr-x 1 bin     3976 Jun  8 1979 mount
-rw-r--r-- 1 root      141 Jun  8 1979 passwd
-rw-r--r-- 1 bin       366 Jun  8 1979 rc
-rw-r--r-- 1 bin       266 Jun  8 1979 ttys
-rwxr-xr-x 1 bin     3794 Jun  8 1979 umount
-rwxr-xr-x 1 bin       634 Jun  8 1979 update
-rw-r--r-- 1 bin        40 Sep 22 05:49 utmp
-rwxr-xr-x 1 root     4520 Jun  8 1979 wall
# ls -l /unix*
-rwxr-xr-x 1 sys     53302 Jun  8 1979 /hptunix
-rwxr-xr-x 1 sys     52850 Jun  8 1979 /hptmunix
-rwxr-xr-x 1 root    50990 Jun  8 1979 /rkunix
-rwxr-xr-x 1 root    51982 Jun  8 1979 /rl2unix
-rwxr-xr-x 1 sys     51790 Jun  8 1979 /rphtunix
-rwxr-xr-x 1 sys     51274 Jun  8 1979 /rphtmunix
# ls -l /bin/sh
-rwxr-xr-x 1 bin     17310 Jun  8 1979 /bin/sh
#

```

Bourne shell

2. **C Shell** :-The C shell(csh) is a UNIX shell or command processor typically run in a text window, allowing the user to type commands. The C shell can also read commands from a file, called a script developed by Bill Joy.

```

Hamilton C shell x64
Hamilton C shell<tm> x64 Release 5.0.d
Copyright (c) 1988-2012 by Hamilton Laboratories. All rights reserved.
1 C% whereis factor
C:\Program Files\Hamilton C shell 2012 x64\Samples\factor.csh
2 C% cat '!'
cat 'whereis factor'
# Calculate the prime factors of an integer.
# Copyright (c) 1989-2012 by Hamilton Laboratories. All rights reserved.

proc factor(n)
    if (n > 3) then
        for i = 2 to floor(sqrt(n)) do
            if (n % i == 0) then
                echo $i
                return factor(n//i)
            end
        end
    end
    return n
end

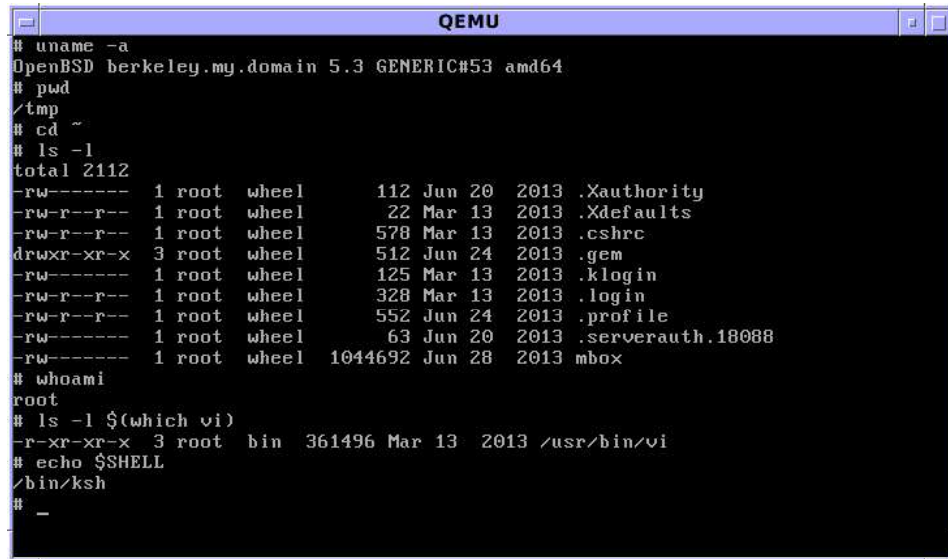
factor $argv
3 C% time factor 1234123412
2
3.98538853
0.00:00.11
4 C%

# whereis.csh, Release 5.0
# Copyright (c) 1988-2012 by Hamilton
#
# Look through the directories on the
# to the command name given, reporting
# file type extension is given as par
# possibilities, .csh, .exe, .com, .c
#
# This is an example of a self-loading
# it is referenced, it runs as a .csh
# the whereis proc. Successive calls
# procedure and will run slightly fas
proc whereis(name)
    local i, j

```

C Shell

3. **Korn Shell** :-Based on the Bourne Shell .Major differences in them are job control, command aliasing and command history,associative arrays, floating point arithmetic operations and a choice of three command line editing styles.



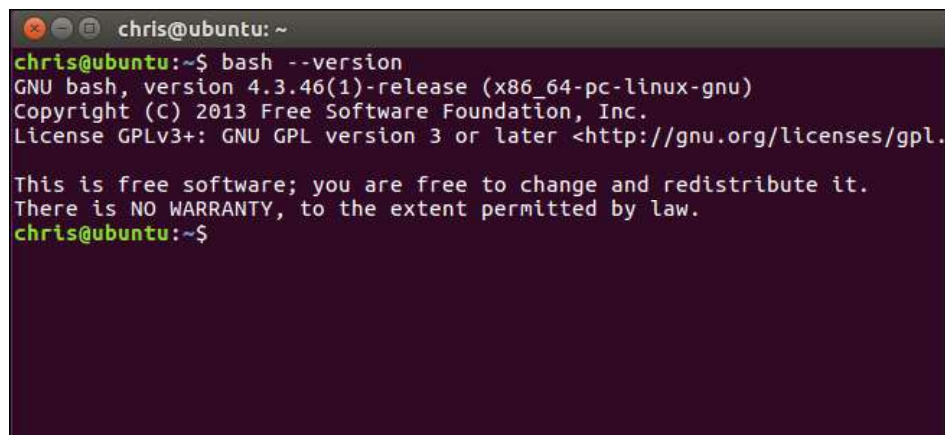
```

# uname -a
OpenBSD berkeley.my.domain 5.3 GENERIC#53 amd64
# pwd
/tmp
# cd ~
# ls -l
total 2112
-rw----- 1 root  wheel      112 Jun 20  2013 .Xauthority
-rw-r--r-- 1 root  wheel       22 Mar 13  2013 .Xdefaults
-rw-r--r-- 1 root  wheel     578 Mar 13  2013 .cshrc
drwxr-xr-x 3 root  wheel     512 Jun 24  2013 .gem
-rw----- 1 root  wheel     125 Mar 13  2013 .klogin
-rw-r--r-- 1 root  wheel     328 Mar 13  2013 .login
-rw-r--r-- 1 root  wheel     552 Jun 24  2013 .profile
-rw----- 1 root  wheel       63 Jun 20  2013 .serverauth.18088
-rw----- 1 root  wheel    1044692 Jun 28  2013 mbox
# whoami
root
# ls -l $(which vi)
-r-xr-xr-x 3 root  bin   361496 Mar 13  2013 /usr/bin/vi
# echo $SHELL
/bin/ksh
#

```

Korn Shell

4. **Bash Shell:-** Bash Shell also known as Bourne again Shell is a type of interpreter that processes shell commands. A shell interpreter takes commands in plain text format and calls Operating System services to do them.



```

chris@ubuntu: ~
chris@ubuntu:~$ bash --version
GNU bash, version 4.3.46(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl>.

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
chris@ubuntu:~$

```

Bash Shell

5. **Tcsh Shell:-** TCSH is a UNIX shell based on and compatible with the C shell (csh). It is essentially the C shell with programmable command line completion, command line editing, and a few other features. Unlike the other common shells, functions cannot be defined in a tcsh script and the user must use aliases instead.

```
FreeBSD 11.0-STABLE (THINKPAD) #0 r317283; Sat Apr 22 23:16:56 EEST 2017

Welcome to FreeBSD!
Want to run the same command again?
In tssh you can type "!!"
user % sudo pkg update
CORRECT>sudo pkg update (y/nlela)?
```

Tssh Shell

SHELL PROGRAMMING:-

There are two ways of writing shell programs.

1. Users can type a sequence of commands and allow the shell to execute them interactively.
2. Users can store those commands in a file that can then be invoked as a program(shell script).

SHELL SCRIPTING:-

A shell script is a computer program designed to be run by the Unix shell, a command line interpreter.

It is stored as a series of command(s) stored in a plain text file.

CAPABILITIES OF A SHELL SCRIPT:-

1. Shell script can take input from the user, file and output them on screen.
2. Useful to create our own commands. Save lots of time.
3. To automate some task of day today life.

4. The System Administration part can also be automated.

PRACTICAL USE CASES OF SHELL SCRIPTING:

1. Monitoring your Linux system.
2. Data backup and creating snapshots.
3. Find out what processes are eating up your system resources.
4. Find out available and free memory.
5. Find out all logged in users and what they are doing.
6. Find out if all necessary network services are running or not.

ADVANTAGES OF SHELL SCRIPTS

1. The command and syntax are exactly the same as those directly entered in command line, so programmer do not need to switch to entirely different syntax
2. Writing shell scripts are much quicker
3. Quick start
4. Interactive debugging etc.

DISADVANTAGES OF SHELL SCRIPTS

1. Prone to costly errors, a single mistake can change the command which might be harmful
2. Slow execution speed
3. Design flaws within the language syntax or implementation
4. Not well suited for large and complex task
5. Provide minimal data structure unlike other scripting languages. etc

QUIZ USING SHELL SCRIPT

CODE:-

```
clear
score=0           \User defined variable
while IFS='#' read -r question choices answer    \Internal Field Separator is set
to # so segregates ,question,choices,answers
do
    echo
    echo $question    \Print the question on the terminal
    echo
    echo $choices     \Prints the choices on the terminal
    echo
    echo "Your Answer:\c" \Asks for input from the user
    read student_answer </dev/tty \Reads the user input from the terminal
    if [ "$student_answer" = "$answer" ]; then \check if the answer matches
        score= `expr $score + 1` \If yes,Increment the score by 1
    fi
    clear \clear screen for the successive questions
done <quiz.txt \Acts as input for the while loop

echo
echo "Your Score is: $score" \Final score is displayed
echo
```

TEXT FILE - QUESTIONS OF THE QUIZ:-

1. Which of the following is NOT a valid deadlock prevention scheme?
 - a. Release all resources before requesting a new resource
 - b. Never request a resource after releasing any resource

Answer: b

2. A graphics card has on board memory of 1 MB. Which of the following modes can the card not support?

- a. 1600 x 400 resolution with 256 colours on a 17 inch monitor
- b. 1600 x 400 resolution with 16 million colours on a 14 inch monitor

Answer: b

3. Which of the following requires a device driver?

- a. Disk
- b. Register

Answer: a

4. More than one word are put in one cache block to?

- a. exploit the spatial locality of reference in a program
- b. exploit the temporal locality of reference in a program

Answer: a

5. Which of the following statements is false?

- a. Virtual memory increases the degree of multiprogramming
- b. Virtual memory reduces the context switching overhead

Answer: b

6. Which of the following need not necessarily be saved on a context switch between processes?

- a. Translation look-aside buffer
- b. Program counter

Answer: a

7. Where does the swap space reside ?

- a. Disk
- b. ROM

Answer: a

8. Which of the following does not interrupt a running process?

- a. Timer
- b. Scheduler process

Answer: b

9. Which of the following scheduling algorithms is non-preemptive?

- a. First-In First-Out

- b. Multilevel Queue Scheduling

Answer: a

10. Which one of the following statements describes the properties achieved?

- a. Progress but not mutual exclusion
- b. Mutual exclusion but not progress

Answer: b

RESULTS:

CODE:

```
clear
score=0
while IFS='#' read -r question choices answer
do
    echo
    echo $question
    echo
    echo $choices
    echo
    echo "Your Answer:"
    read student_answer </dev/tty
    if [ "$student_answer" = "$answer" ]; then
        score=`expr $score + 1`
    fi
    clear
done <quiz.txt

echo
echo "Your Score is: $score"
echo
~
~
~
~
~
```

Code

OUTPUT:

```
Which of the following is NOT a valid deadlock prevention scheme?
```

- a) Release all resources before requesting a new resource
- b) Never request a resource after releasing any resource

```
Your Answer:
```

Output 1

```
A graphics card has on board memory of 1 MB. Which of the following modes can the card not support?
```

- a) 1600 x 400 resolution with 256 colours on a 17 inch monitor
- b) 1600 x 400 resolution with 16 million colours on a 14 inch monitor

```
Your Answer:
```

Output 2

```
Which of the following requires a device driver?
```

- a) Disk
- b) Register

```
Your Answer:
```

Output 3

```
Your Score is: 5
```

Final Result