Shell Programming

Practical-5

1. Write a Shell script to accept a string as command line argument and reverse the same.

Solution:

1. Count no of character in string using WC = len
2. Loop from len to 0
3. Use Cut command to extract char at position I
4. store ith char and append next i-1

Practical-5

1. echo entered string is $1
2. s=`echo $1|wc -c`
3. d=" "
4. for ((i=$s ; i>0 ; i--))
5. do
6. t=`echo $1|cut -c $i`
7. d=dt
8. done
9. echo Reverse of the given string is $d

Practical-5

2. Write a shell script to calculate the loss percentage of an article, given the cost price and the selling price as command line arguments.

Solution:

- **Profit = Selling price – Cost Price**
- **Loss = Cost Price – Selling Price**
- **Profit percentage = Profit x 100 / Cost Price**
- **Loss percentage = Loss x 100/ Cost price**

Practical-5

3. Write a shell script to accept the name of the user and check out if the same has logged in or not.

Solution :

1. Who command :list of users
2. Use for loop to search user in list
3. If exist use Cut command and extract user name only
4. Print message

Practical-5

4. Write a shell script to check whether the file whose name is scanned exists and readable

1. Ls –l give list of file with permission
2. Use cut command to check readable or not and extract file name
3. Match file name with entered file name and compare with –r to check readable or not

Practical-5

5. Write a shell script to check if the input string is a palindrome.

- Read string
- Calculate length of string: wc –c
- Use loop to reverse string
- Compare both string
- If same palindrome
- Else not palindrome

Practical-5

6. Write a shell script to accept a number and a word as command line arguments and print the word the given number of times on each line.

   - Read no=3 and string=XY
   - For loop from 0 to 3
   - S1= XY XY XY
   - Loop 0 to 3
   - Print s1

Practical-5

7. Write a shell script to find the file or directory with the maximum size in the current directory.

1. Max=0

2. For loop :use Ls –l and get field that show file size(for I in cmd)

3. Any value in field has max value the set as Max(i>max then i=Max)

4. View field which has max value and apply cat to get file name field(cut,grep,ls)

Practical-5

8. Write a shell script to accept two filenames and check if both exist.
   - If the second filename exists,

     then the contents of the first filename should be appended to it.
   - If the second filename does not exist

     then create a new file with the contents of the first file.

Input Output Redirection

- **Output Redirection**
- The '**>**' symbol is used for output (STDOUT) redirection.
- ">>" appends output to an existing file
- **Example**
  - Echo “UVPCE” > Out.txt
  - **Out.txt**
    - UVPCE
  - Echo “GANPAT UNIVERSITY” >> out.txt
  - **Out.txt**
    - UVPCE
    - GANPAT UNIVERSITY

Input Output Redirection

File	File Descriptor
Standard Input STDIN	0
Standard Output STDOUT	1
Standard Error STDERR	2

You can re-direct error using its corresponding File Descriptor 2.

```
[sbp@linuxserv PRACTICAL-5]$ cat sample.sh
ech students
read name
echo name
[sbp@linuxserv PRACTICAL-5]$ sh sample.sh 2> error
shiv
name
[sbp@linuxserv PRACTICAL-5]$ cat error
sample.sh: line 1: ech: command not found
```

Input Output Redirection

- "<" is the input redirection operator

```
[sbp@linuxserv PRACTICAL-5]$ grep echo < P5.sh
echo entered file name is $1
echo file found
echo file does not exist
[sbp@linuxserv PRACTICAL-5]$
```

- ">&"re-directs output of one file to another.
- **Example:** both the output is written to file output file

```
[sbp@linuxserv PRACTICAL-5]$ ls -l d> p1 2>&1
[sbp@linuxserv PRACTICAL-5]$ cat p1
ls: d: No such file or directory
[sbp@linuxserv PRACTICAL-5]$ ls -l -d> p1 2>&1
[sbp@linuxserv PRACTICAL-5]$ cat p1
drwxrwxr-x 2 sbp sbp 4096 Mar  5 14:12 .
[sbp@linuxserv PRACTICAL-5]$
```

$? In Shell

Display exit status of the last command executed.

```
[sbp@linuxserv PRACTICAL-5]$ ls
P1.sh  P2_2.sh  P4.sh  P6.sh  P8.sh  p1  sample.sh
P2.sh  P3.sh    P5.sh  P7.sh  error  p8
[sbp@linuxserv PRACTICAL-5]$ cat error
SHELL PROGRAMMING
[sbp@linuxserv PRACTICAL-5]$ echo $?
0
[sbp@linuxserv PRACTICAL-5]$ cat error1
cat: error1: No such file or directory
[sbp@linuxserv PRACTICAL-5]$ echo $?
1
[sbp@linuxserv PRACTICAL-5]$
```

Practical-5

8. Solution:

- Read file1 and file2
- Check/read content of file2 using cat command
- If $? =0[if file2 is available)
- Append content of file1 to file2 and print it
- Else
- Create new file using touch command
- Store content of file1 into new file and display it

Practical-5

9. Write a shell script to accept a number in the command line and displays the sum up to that number. By default, the sum up to 50 should be displayed.

- Read no
- If no >0
- For i=0 to no c=c+I
- Print c
- Else
- For i=0 to 50
- c=c+i
- print c

Practical-5

10. Write a shell script to find the number of ordinary files and directory files in the current directory.

- For all item in list
- Get 1st character using Ls –l | cut –c 1
- If char == " –" print ordinary file
- Else if char=="d" print directory

Practical-5

11. Write a shell script to accept an alphabet from the user and list all the files/directory starting with that alphabet in the current directory.

- Read ch
- Print ls ch*

Pracatical-7

- **FIRST COME FIRST SERVE SCHEDULING**
- **AIM:**
- To write the program to implement CPU & scheduling algorithm for first come first serve scheduling.

Pracatical-7

- Completion Time: Process completes its execution.
- Turn Around Time = Completion Time – Arrival Time
- Waiting Time = Turn Around Time – Burst Time

Pracatical-7

Process	Duration	Oder	Arrival Time
P1	24	1	0
P2	3	2	0
P3	4	3	0

Gantt Chart :

P1 waiting time : 0
P2 waiting time : 24
P3 waiting time : 27

The Average waiting time :

(0+24+27)/3 = 17

Pracatical-7

Process	Duration	Oder	Arrival Time
P1	24	1	0
P2	3	2	0
P3	4	3	0

Gantt Chart :

P1 waiting time : 0
P2 waiting time : 24
P3 waiting time : 27

The Average waiting time :

(0+24+27)/3 = 17

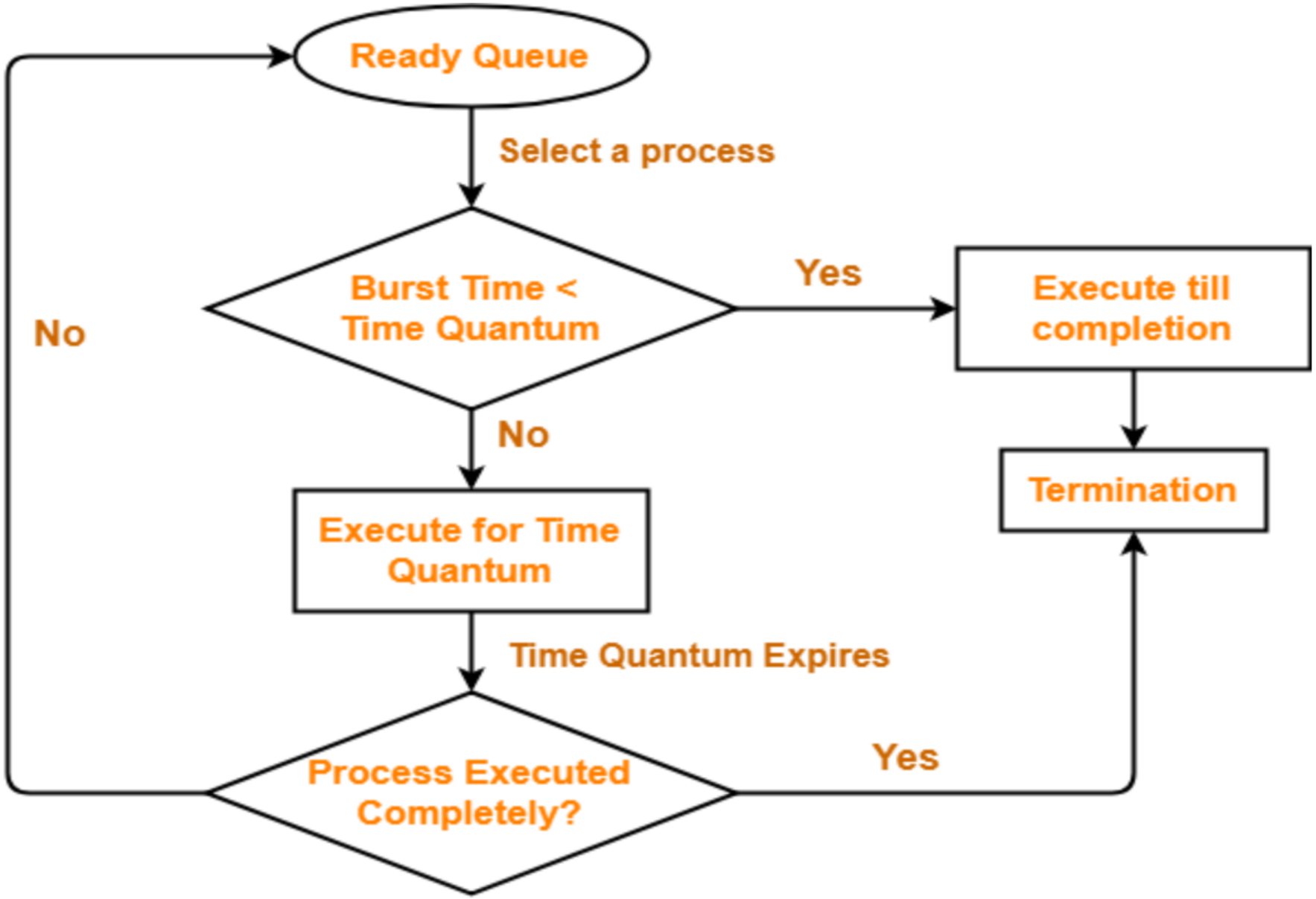
INPUT

Enter the number of processes -- 3
Enter Burst Time for Process 0 -- 24
Enter Burst Time for Process 1 -- 3
Enter Burst Time for Process 2 -- 3

OUTPUT

PROCESS	BURST TIME	WAITING TIME	TURNAROUND TIME
P0	24	0	24
P1	3	24	27
P2	3	27	30

Average Waiting Time-- 17.000000
Average Turnaround Time -- 27.000000


Ready Queue
Select a process
Burst Time < Time Quantum
Yes
No
Execute till completion
Termination
No
Execute for Time Quantum
Time Quantum Expires
Process Executed Completely?
Yes