Operating System

CSE202

Practical File
submitted to the
Amity University Uttar Pradesh

Submitted by:
Bhavesh Dalal
A2305320007

Supervisor :
Dr. Sunil Kumar Chowdhary

DEPARTMENT OF INFORMATION TECHNOLOGY
AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY
AMITY UNIVERSITY UTTAR PRADESH
2022

Index

S.No.	Name Of Experiment	Date of Allotm	Date of Evaluation	Faculty Signatu
1.	Basic Unix/Linux Commands			
2.	C Programming and some basic commands in Linux			
3.	WAP in C language to perform the following operation to implement FCFS Algorithm: Calculate: 1. Waiting time for each process 2. Total Waiting Time 3. Average waiting time 4. Turnaround time for each process 5 Total Turnaround time			
4.	WAP in C language to perform the following operation to implement SJF Algorithm: Calculate: 1. Waiting time for each process 2. Total Waiting Time 3. Average waiting time 4. Turnaround time for each process 5 Total Turnaround time			

5	WAP in C Language to perform the following operation to implement SRTF (Shortest Remaining Time First) Algorithm: Calculate: 1. Waiting time for each process 2. Total Waiting Time 3. Average waiting time 4. Turnaround time for each process 5 Total Turnaround time			
6	WAP in C Language to perform the following operation to implement Round Robin Algorithm: Calculate: 1. Waiting time for each process 2. Total Waiting Time 3. Average waiting time 4. Turnaround time for each process 5. Total Turnaround time			

Date: 20/1/2022

Lab Assignment 1

Aim: Basic Unix/Linux Commands

Platform Used: JS Bell Linux

Linux Commands:

→ls

→ cd /ls

→cd etc

→ls

→cat passwd

→cd ..ls

→cd user/

→cd tmp/

→cd mc-root/

→cd ~ls

→touch abcls

→ls –l

→chmod u+x abc

→chmod o+x abc

→chmod u-x abc

→chmod ugo+x abc

→chmod 777 abc

→chmod 660 abc

→pwd

→cd /

→date

→date +%h

→date +%d

→date +%H

→date +%M

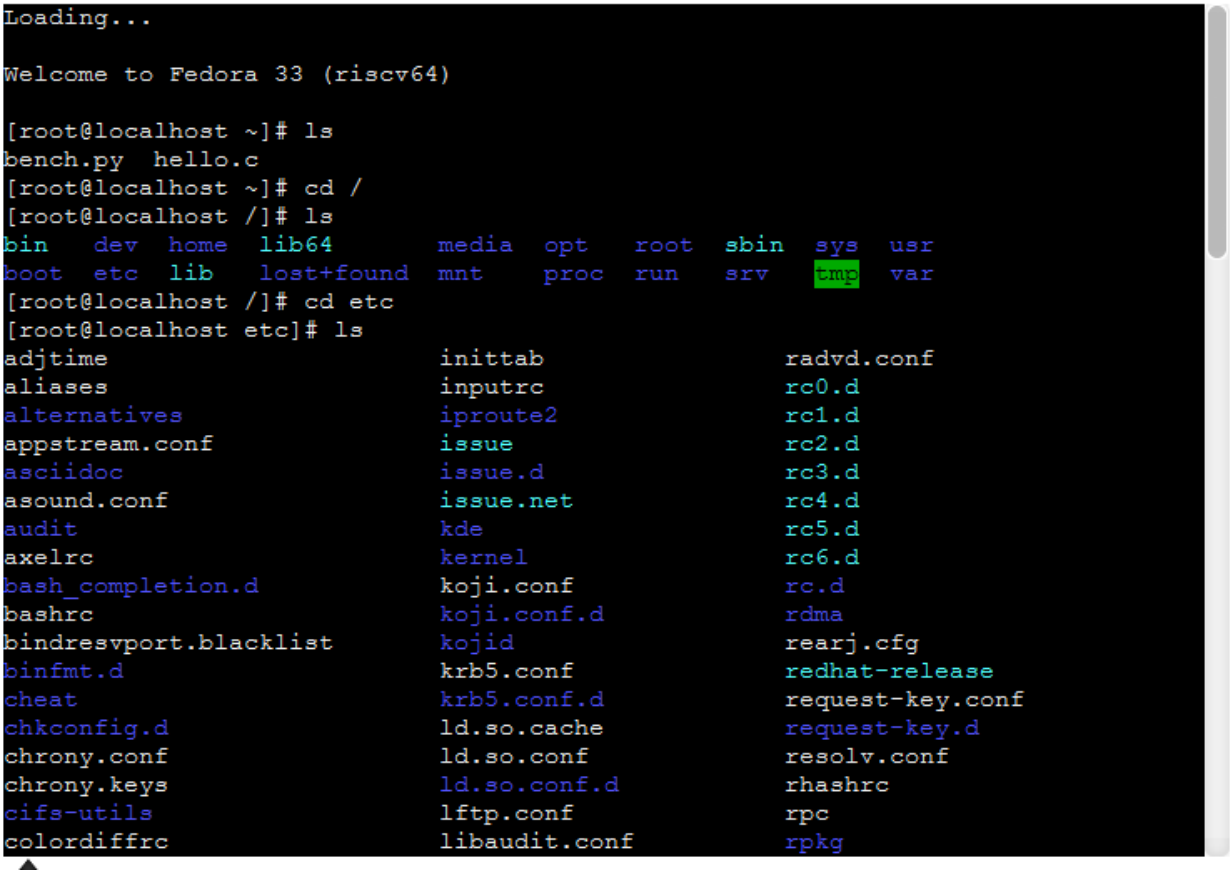
→cal 7 2022

→cal 2022

→hostname

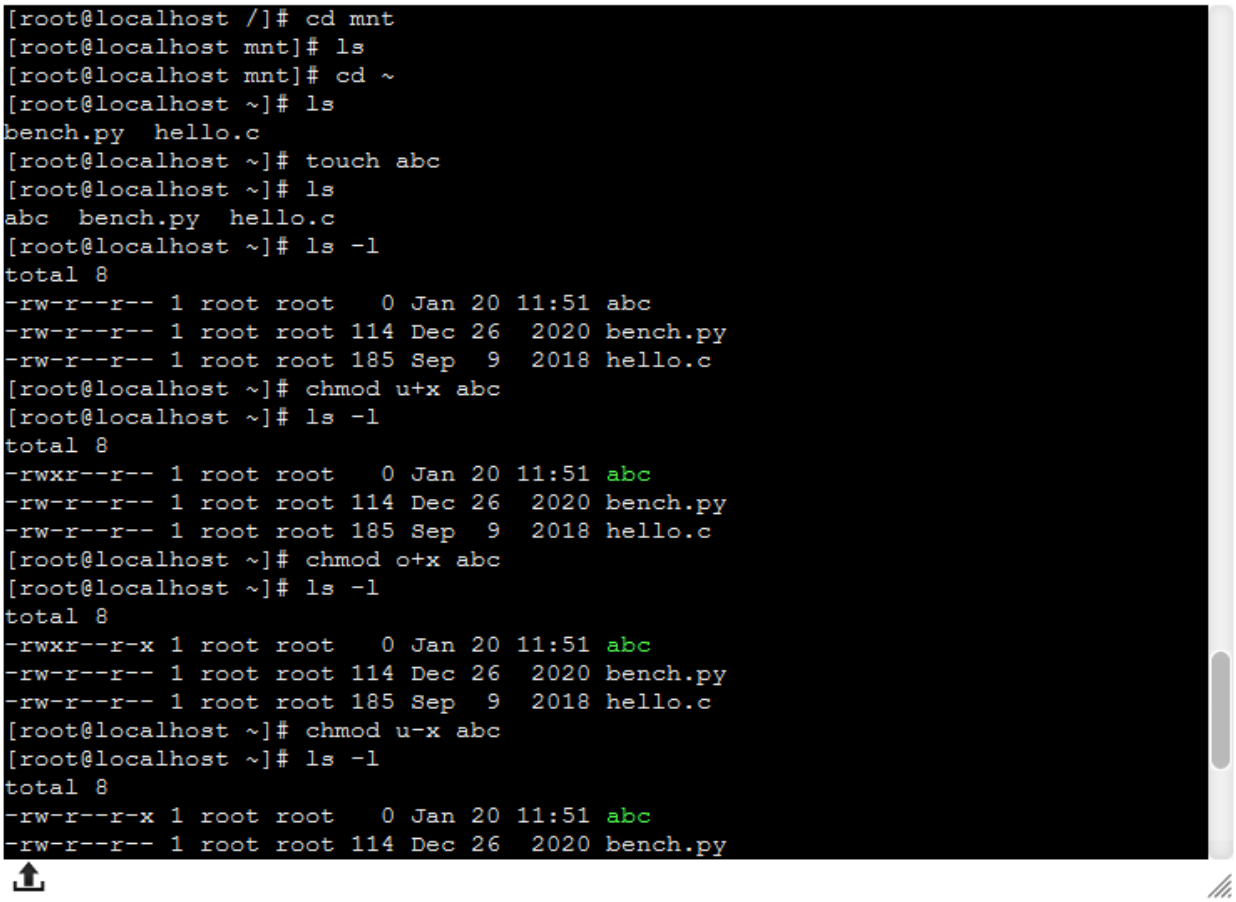
→bc

→quit


Loading...
Welcome to Fedora 33 (riscv64)
[root@localhost ~]# ls
bench.py hello.c
[root@localhost ~]# cd /
[root@localhost /]# ls
bin dev home lib64 media opt root sbin sys usr
boot etc lib lost+found mnt proc run srv tmp var
[root@localhost /]# cd etc
[root@localhost etc]# ls
adjtime
aliases
alternatives
appstream.conf
asciidoc
asound.conf
audit
axelrc
bash_completion.d
bashrc
bindresvport.blacklist
binfmt.d
cheat
chkconfig.d
chrony.conf
chrony.keys
cifs-utils
colordiffrc
inittab
inputrc
iproute2
issue
issue.d
issue.net
kde
kernel
koji.conf
koji.conf.d
kojid
krb5.conf
krb5.conf.d
ld.so.cache
ld.so.conf
ld.so.conf.d
lftp.conf
libaudit.conf
radvd.conf
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
rc6.d
rc.d
rdma
rearj.cfg
redhat-release
request-key.conf
request-key.d
resolv.conf
rhashrc
rpc
rpkg

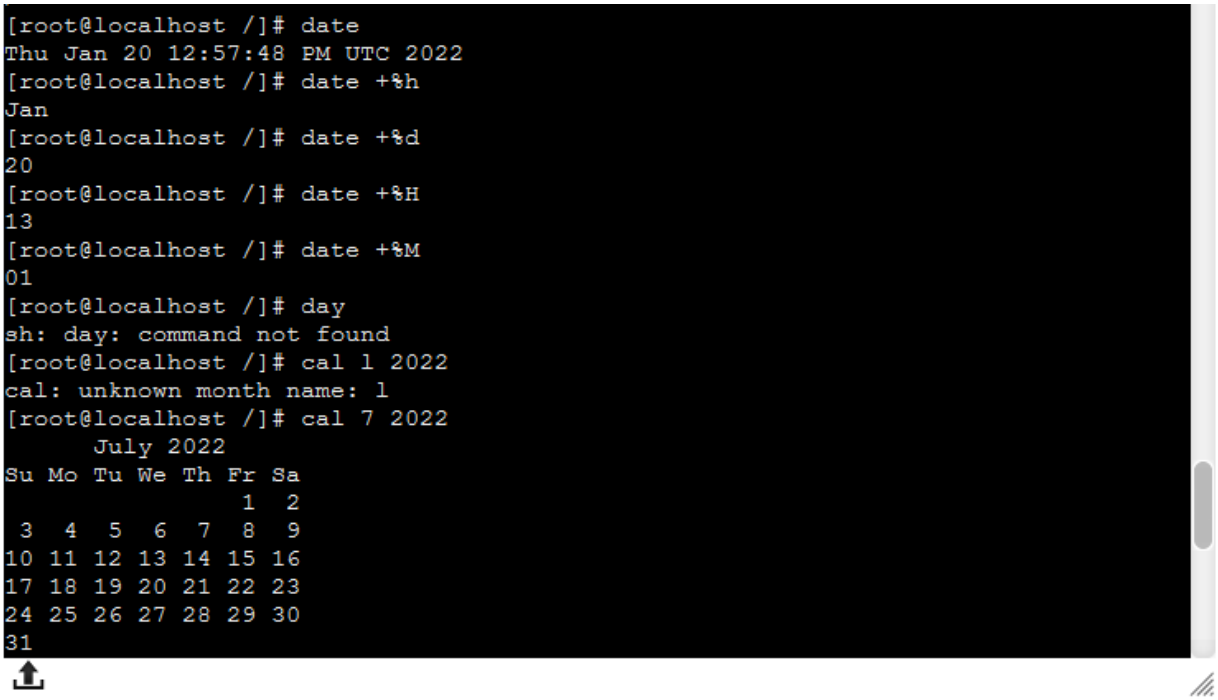
```
idmapd.conf                    profile.d               zlogout
ImageMagick-6                  protocols               zprofile
incron.conf                    pulse                   zshenv
incron.d                       qemu                    zshrc
init.d                         qemu-ga
[root@localhost etc]# cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
unbound:x:999:999:Unbound DNS resolver:/etc/unbound:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
systemd-coredump:x:998:995:systemd Core Dumper:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
systemd-timesync:x:997:994:systemd Time Synchronization:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:996:993:User for polkitd:/:/sbin/nologin
saslauth:x:995:76:Saslauthd user:/run/saslauthd:/sbin/nologin
dnsmasq:x:992:992:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
setroubleshoot:x:991:991::/var/lib/setroubleshoot:/sbin/nologin
```

```
[root@localhost /]# cd ..
[root@localhost /]# ls
bin   dev  home  lib64       media  opt   root  sbin  sys  usr
boot  etc  lib   lost+found  mnt    proc  run   srv   tmp  var
[root@localhost /]# cd usr/
[root@localhost usr]# ls
bin  games  include  lib  lib64  libexec  local  sbin  share  src  tmp
[root@localhost usr]# cd tmp/
[root@localhost tmp]# ls
mc-root
[root@localhost tmp]# cd mc-root/
[root@localhost mc-root]# ls
[root@localhost mc-root]# cd ..
[root@localhost tmp]# ls
mc-root
[root@localhost tmp]# cd ..
```


[root@localhost /]# cd mnt
[root@localhost mnt]# ls
[root@localhost mnt]# cd ~
[root@localhost ~]# ls
bench.py hello.c
[root@localhost ~]# touch abc
[root@localhost ~]# ls
abc bench.py hello.c
[root@localhost ~]# ls -l
total 8
-rw-r--r-- 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26 2020 bench.py
-rw-r--r-- 1 root root 185 Sep 9 2018 hello.c
[root@localhost ~]# chmod u+x abc
[root@localhost ~]# ls -l
total 8
-rwxr--r-- 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26 2020 bench.py
-rw-r--r-- 1 root root 185 Sep 9 2018 hello.c
[root@localhost ~]# chmod o+x abc
[root@localhost ~]# ls -l
total 8
-rwxr--r-x 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26 2020 bench.py
-rw-r--r-- 1 root root 185 Sep 9 2018 hello.c
[root@localhost ~]# chmod u-x abc
[root@localhost ~]# ls -l
total 8
-rw-r--r-x 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26 2020 bench.py

```
-rwxr--r-x 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
[root@localhost ~]# chmod u-x abc
[root@localhost ~]# ls -l
total 8
-rw-r--r-x 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
[root@localhost ~]# chmod ugo+x abc
[root@localhost ~]# ls -l
total 8
-rwxr-xr-x 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
[root@localhost ~]# chmod 777
chmod: missing operand after ‘777’
Try 'chmod --help' for more information.
[root@localhost ~]# ls -l
total 8
-rwxr-xr-x 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
[root@localhost ~]# chmod 777 abc
[root@localhost ~]# ls -l
total 8
-rwxrwxrwx 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
[root@localhost ~]#
```

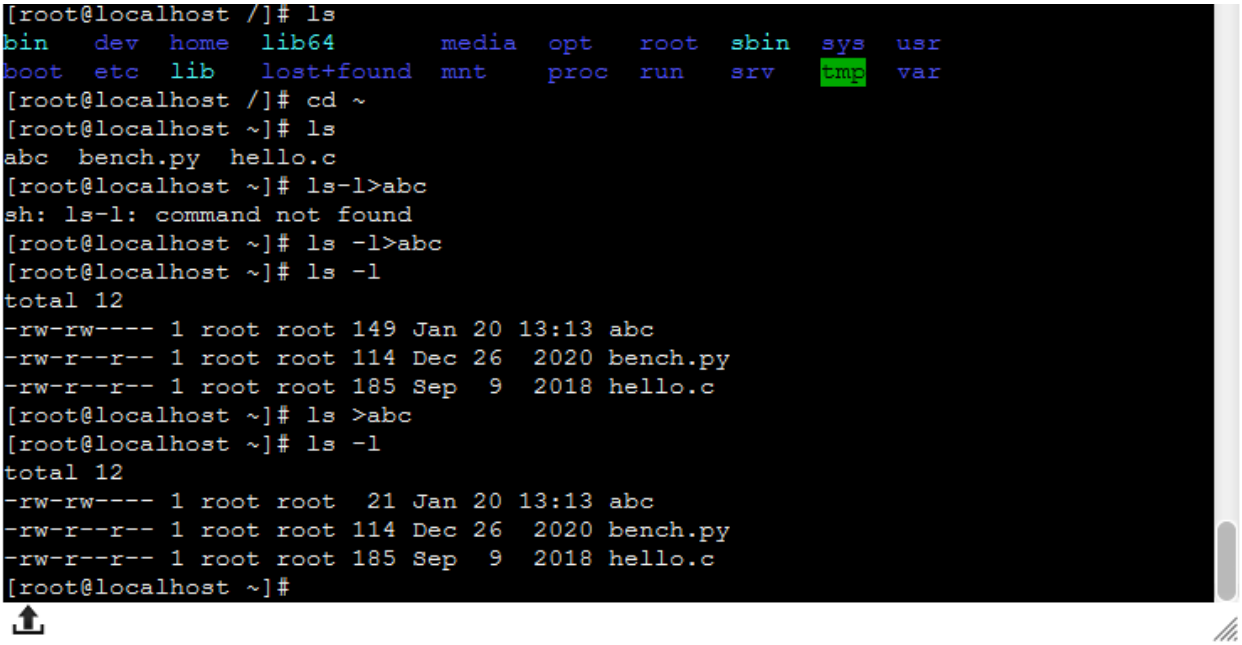
```
[root@localhost ~]# chmod 660 abc
[root@localhost ~]# ls -l
total 8
-rw-rw---- 1 root root   0 Jan 20 11:51 abc
-rw-r--r-- 1 root root 114 Dec 26  2020 bench.py
-rw-r--r-- 1 root root 185 Sep  9  2018 hello.c
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /
[root@localhost /]# pwd
/
[root@localhost /]#
```


[root@localhost /]# date
Thu Jan 20 12:57:48 PM UTC 2022
[root@localhost /]# date +%h
Jan
[root@localhost /]# date +%d
20
[root@localhost /]# date +%H
13
[root@localhost /]# date +%M
01
[root@localhost /]# day
sh: day: command not found
[root@localhost /]# cal l 2022
cal: unknown month name: l
[root@localhost /]# cal 7 2022
July 2022
Su Mo Tu We Th Fr Sa
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31

```
[root@localhost /]# cal 2022
                                2022

       January               February                March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                   1            1  2  3  4  5            1  2  3  4  5
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    6  7  8  9 10 11 12
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   13 14 15 16 17 18 19
16 17 18 19 20 21 22   20 21 22 23 24 25 26   20 21 22 23 24 25 26
23 24 25 26 27 28 29   27 28                 27 28 29 30 31
30 31

        April                  May                   June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2    1  2  3  4  5  6  7               1  2  3  4
 3  4  5  6  7  8  9    8  9 10 11 12 13 14    5  6  7  8  9 10 11
10 11 12 13 14 15 16   15 16 17 18 19 20 21   12 13 14 15 16 17 18
17 18 19 20 21 22 23   22 23 24 25 26 27 28   19 20 21 22 23 24 25
24 25 26 27 28 29 30   29 30 31              26 27 28 29 30

        July                  August               September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2       1  2  3  4  5  6                  1  2  3
 3  4  5  6  7  8  9    7  8  9 10 11 12 13    4  5  6  7  8  9 10
10 11 12 13 14 15 16   14 15 16 17 18 19 20   11 12 13 14 15 16 17
17 18 19 20 21 22 23   21 22 23 24 25 26 27   18 19 20 21 22 23 24
24 25 26 27 28 29 30   28 29 30 31            25 26 27 28 29 30
31
```

```
       October               November              December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                   1            1  2  3  4  5                  1  2  3
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    4  5  6  7  8  9 10
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   11 12 13 14 15 16 17
16 17 18 19 20 21 22   20 21 22 23 24 25 26   18 19 20 21 22 23 24
23 24 25 26 27 28 29   27 28 29 30            25 26 27 28 29 30 31
30 31
[root@localhost /]#
```

```
[root@localhost /]# hostname
localhost
[root@localhost /]# echo 'Meenakshi'
Meenakshi
[root@localhost /]# bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software
 Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
12/5 - 3
-1
2*37/8
9
quit
```


[root@localhost /]# ls
bin dev home lib64 media opt root sbin sys usr
boot etc lib lost+found mnt proc run srv tmp var
[root@localhost /]# cd ~
[root@localhost ~]# ls
abc bench.py hello.c
[root@localhost ~]# ls-l>abc
sh: ls-l: command not found
[root@localhost ~]# ls -l>abc
[root@localhost ~]# ls -l
total 12
-rw-rw---- 1 root root 149 Jan 20 13:13 abc
-rw-r--r-- 1 root root 114 Dec 26 2020 bench.py
-rw-r--r-- 1 root root 185 Sep 9 2018 hello.c
[root@localhost ~]# ls >abc
[root@localhost ~]# ls -l
total 12
-rw-rw---- 1 root root 21 Jan 20 13:13 abc
-rw-r--r-- 1 root root 114 Dec 26 2020 bench.py
-rw-r--r-- 1 root root 185 Sep 9 2018 hello.c
[root@localhost ~]#

Date: 27/1/2022

Lab Assignment 2

Aim: C Programming and some basic commands in Linux

Platform Used: JS Bell Linux

```
Loading...

Welcome to Fedora 33 (riscv64)

[root@localhost ~]# ls
bench.py  hello.c
[root@localhost ~]# mv hello.c hello.cpp
[root@localhost ~]# ls
bench.py  hello.cpp
[root@localhost ~]# mv bench.py bench.sh
[root@localhost ~]# ls
bench.sh  hello.cpp
[root@localhost ~]# ln hello.cpp hello.c
[root@localhost ~]# ls
bench.sh  hello.c  hello.cpp
```

```
[root@localhost ~]# cat hello.c
/* This C source can be compiled with:
   gcc -o hello hello.c
*/
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char **argv)
{
    printf("Hello World\n");
    return 0;
}
[root@localhost ~]# hello.cpp
sh: hello.cpp: command not found
[root@localhost ~]# cat hello.cpp
/* This C source can be compiled with:
   gcc -o hello hello.c
*/
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char **argv)
{
    printf("Hello World\n");
    return 0;
}
```

```
[root@localhost ~]# wc hello.c
 11  28 185 hello.c
[root@localhost ~]# wc hello.cpp
 11  28 185 hello.cpp
[root@localhost ~]# grep return hello.c
    return 0;
[root@localhost ~]# grep int main hello.cpp
grep: main: No such file or directory
hello.cpp:int main(int argc, char **argv)
hello.cpp:    printf("Hello World\n");
```

```
Loading...

Welcome to Fedora 33 (riscv64)

[root@localhost ~]# ls
bench.py  hello.c
[root@localhost ~]# sort -r hello.c
/* This C source can be compiled with:
    return 0;
    printf("Hello World\n");
int main(int argc, char **argv)
#include <stdlib.h>
#include <stdio.h>
   gcc -o hello hello.c
}
{
*/

[root@localhost ~]# head +2 hello.c
head: cannot open '+2' for reading: No such file or directory
==> hello.c <==
/* This C source can be compiled with:
   gcc -o hello hello.c
*/
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char **argv)
{
    printf("Hello World\n");
```

```
    return 0;
[root@localhost ~]# tail -2 hello.c
    return 0;
}
[root@localhost ~]# cmp bench.py hello.c
bench.py hello.c differ: byte 1, line 1
[root@localhost ~]# ps
  PID TTY          TIME CMD
   47 hvc0     00:00:01 sh
  105 hvc0     00:00:00 ps
[root@localhost ~]# pstree
init─┬─dhcpcd
     └─sh───pstree
[root@localhost ~]# kill 47
[root@localhost ~]# ps -ag
  PID TTY      STAT   TIME COMMAND
   47 hvc0     Ss     0:01 sh -l
  107 hvc0     R+     0:00 ps -ag
```

```
[root@localhost ~]# touch xyz
[root@localhost ~]# ls
bench.py  hello.c  xyz
[root@localhost ~]# date > xyz
[root@localhost ~]# cat xyz
Thu Jan 27 12:26:06 PM UTC 2022
[root@localhost ~]# cal 2022 > xyz
[root@localhost ~]# cat xyz
                                2022

       January               February                March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                   1            1  2  3  4  5            1  2  3  4  5
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    6  7  8  9 10 11 12
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   13 14 15 16 17 18 19
16 17 18 19 20 21 22   20 21 22 23 24 25 26   20 21 22 23 24 25 26
23 24 25 26 27 28 29   27 28                 27 28 29 30 31
30 31
        April                  May                    June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2    1  2  3  4  5  6  7              1  2  3  4
 3  4  5  6  7  8  9    8  9 10 11 12 13 14    5  6  7  8  9 10 11
10 11 12 13 14 15 16   15 16 17 18 19 20 21   12 13 14 15 16 17 18
17 18 19 20 21 22 23   22 23 24 25 26 27 28   19 20 21 22 23 24 25
24 25 26 27 28 29 30   29 30 31              26 27 28 29 30
```

```
        July                August              September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2       1  2  3  4  5  6                1  2  3
 3  4  5  6  7  8  9    7  8  9 10 11 12 13    4  5  6  7  8  9 10
10 11 12 13 14 15 16   14 15 16 17 18 19 20   11 12 13 14 15 16 17
17 18 19 20 21 22 23   21 22 23 24 25 26 27   18 19 20 21 22 23 24
24 25 26 27 28 29 30   28 29 30 31            25 26 27 28 29 30
31
       October             November             December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1           1  2  3  4  5                1  2  3
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    4  5  6  7  8  9 10
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   11 12 13 14 15 16 17
16 17 18 19 20 21 22   20 21 22 23 24 25 26   18 19 20 21 22 23 24
23 24 25 26 27 28 29   27 28 29 30            25 26 27 28 29 30 31
30 31
```

```
[root@localhost ~]# date >> xyz
[root@localhost ~]# cat xyz
                              2022

       January             February              March
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1           1  2  3  4  5          1  2  3  4  5
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    6  7  8  9 10 11 12
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   13 14 15 16 17 18 19
16 17 18 19 20 21 22   20 21 22 23 24 25 26   20 21 22 23 24 25 26
23 24 25 26 27 28 29   27 28                 27 28 29 30 31
30 31
        April                 May                  June
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2    1  2  3  4  5  6  7             1  2  3  4
 3  4  5  6  7  8  9    8  9 10 11 12 13 14    5  6  7  8  9 10 11
10 11 12 13 14 15 16   15 16 17 18 19 20 21   12 13 14 15 16 17 18
17 18 19 20 21 22 23   22 23 24 25 26 27 28   19 20 21 22 23 24 25
24 25 26 27 28 29 30   29 30 31              26 27 28 29 30

        July                August              September
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                1  2       1  2  3  4  5  6                1  2  3
 3  4  5  6  7  8  9    7  8  9 10 11 12 13    4  5  6  7  8  9 10
10 11 12 13 14 15 16   14 15 16 17 18 19 20   11 12 13 14 15 16 17
17 18 19 20 21 22 23   21 22 23 24 25 26 27   18 19 20 21 22 23 24
24 25 26 27 28 29 30   28 29 30 31            25 26 27 28 29 30
31
```

```
       October               November              December
Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa   Su Mo Tu We Th Fr Sa
                    1             1  2  3  4  5                  1  2  3
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    4  5  6  7  8  9 10
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   11 12 13 14 15 16 17
16 17 18 19 20 21 22   20 21 22 23 24 25 26   18 19 20 21 22 23 24
23 24 25 26 27 28 29   27 28 29 30            25 26 27 28 29 30 31
30 31
Thu Jan 27 12:26:56 PM UTC 2022
```

```
[root@localhost ~]# a=10
[root@localhost ~]# echo $a
10
[root@localhost ~]# a=MEENAKSHI
[root@localhost ~]# echo $a
MEENAKSHI
[root@localhost ~]# echo "Meenakshi Bhandari"
Meenakshi Bhandari
[root@localhost ~]# echo "Date is `date`"
Date is Thu Jan 27 12:42:46 PM UTC 2022
[root@localhost ~]# a=10.363536278276630
[root@localhost ~]# echo $a
10.363536278276630
[root@localhost ~]# echo "The Files and Directories are as `ls -al`"
The Files and Directories are as total 44
dr-xr-x---  2 root root  263 Dec 26  2020 .
drwxrwxrwx 21 root root  550 Jan 27 12:18 ..
-rw-r--r--  1 root root   18 Jul 30  2020 .bash_logout
-rw-r--r--  1 root root  141 Jul 30  2020 .bash_profile
-rw-r--r--  1 root root  429 Jul 30  2020 .bashrc
-rw-r--r--  1 root root  114 Dec 26  2020 bench.py
-rw-r--r--  1 root root  100 Jul 30  2020 .cshrc
-rw-r--r--  1 root root  106 Dec 20  2020 .fldev_cfg
-rw-r--r--  1 root root  185 Sep  9  2018 hello.c
-rw-r--r--  1 root root  129 Jul 30  2020 .tcshrc
-rw-r--r--  1 root root 2244 Jan 27 12:26 xyz
[root@localhost ~]#
```

Date: 3/2/2022

Lab Assignment 3

Aim: WAP in C Langauge to perform the following operation to implement FCFS Algorithm:

Calculate:

1. Waiting time for each process

2. Total Waiting Time

3. Average waiting time

4. Turnaround time for each process

5 Total Turnaround time

Platform Used: JS Bell Linux

Code:

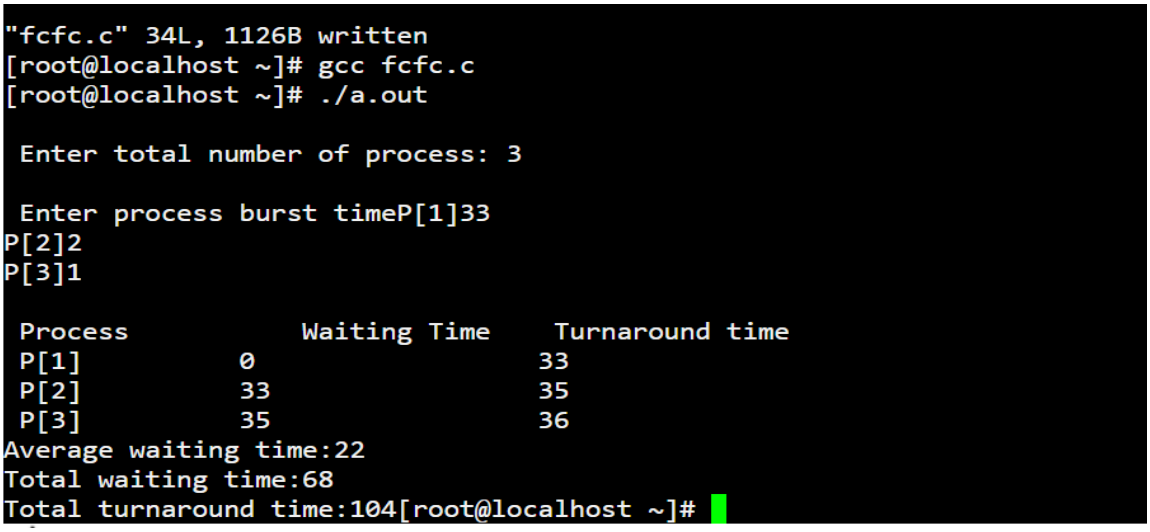
```
#include<stdio.h>

int main(){

int
n,burst_time[10],waiting_time[10],turnaround_time[10],avg_waitin
gtime=0,

total_turnaroundtime=0, total_waitingtime=0, i, j;

printf("\n Enter total number of process: ");

scanf("%d",&n);

printf("\n Enter process burst time");

 for(i=0;i<n;i++){
```

```
    printf("P[%d]",i+1);
    scanf("%d",&burst_time[i]);
    waiting_time[0]=0;
     for(j=0;j<i;j++){
        waiting_time[i]+=burst_time[j];
    }
 }
printf("\n Process\t    Waiting Time\t   Turnaround time");
for(i=0;i<n;i++){
 turnaround_time[i]=burst_time[i]+waiting_time[i];
 avg_waitingtime+=waiting_time[i];
 total_turnaroundtime+=turnaround_time[i];
 printf("\n P[%d]\t      %d\t
%d",i+1,waiting_time[i],turnaround_time[i]);
}
avg_waitingtime=avg_waitingtime/i;
total_waitingtime+=total_waitingtime;
total_turnaroundtime+=total_turnaroundtime;
printf("\n Average waiting time:%d",avg_waitingtime);
printf("\n Total waiting time:%d",total_waitingtime);
printf("\n Total turnaround time:%d",total_turnaroundtime);
```

return 0;

}

OUTPUT:



Date: 17/2/2022

Lab Assignment 4

Aim: WAP in C Langauge to perform the following operation to implement SJF Algorithm:

Calculate:

1. Waiting time for each process

2. Total Waiting Time

3. Average waiting time

4. Turnaround time for each process

5 Total Turnaround time

Platform Used: JS Bell Linux

Code:

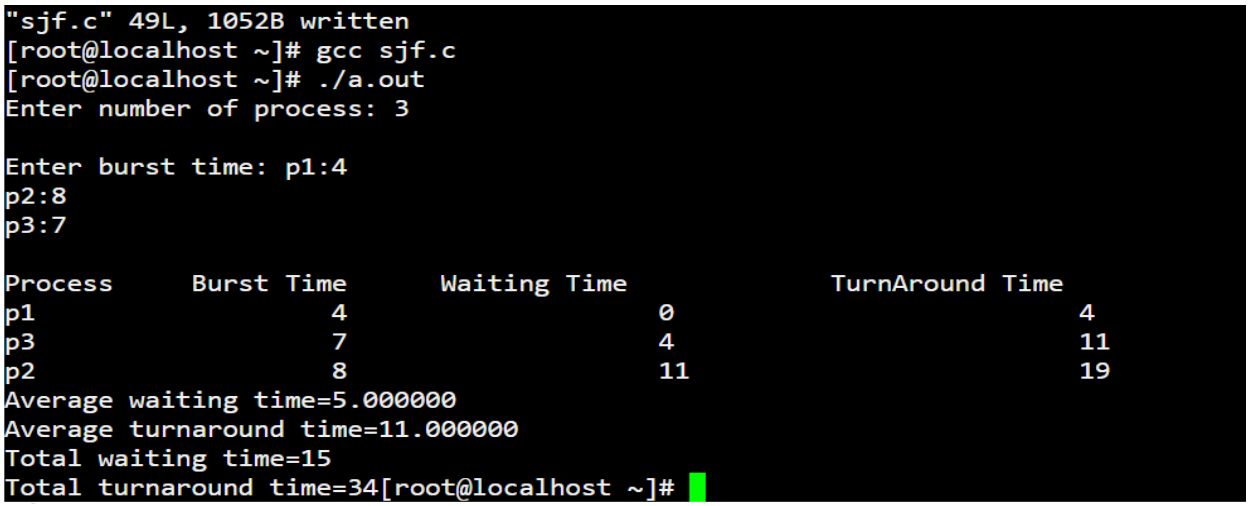
```
#include<stdio.h>
void main(){
    int
bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp,total_tat,total_wt;
    float avg_wt,avg_tat;
printf("Enter number of process:");
    scanf("%d",&n);
printf("\nEnter Burst Time:\n");
    for(i=0;i<n;i++)
```

```
    {
        printf("p%d:",i+1);
        scanf("%d",&bt[i]);
        p[i]=i+1;
    }
    for(i=0;i<n;i++){
        pos=i;
        for(j=i+1;j<n;j++)
        {
            if(bt[j]<bt[pos])
                pos=j;
        }
temp=bt[i];
        bt[i]=bt[pos];
        bt[pos]=temp;

        temp=p[i];
        p[i]=p[pos];
        p[pos]=temp;
    }
wt[0]=0;
```

```
    for(i=1;i<n;i++)
    {
        wt[i]=0;
        for(j=0;j<i;j++)
            wt[i]+=bt[j];
total+=wt[i];
    }
  avg_wt=total/n;
    total=0;
printf("\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time");
    for(i=0;i<n;i++)
    {
tat[i]=bt[i]+wt[i];
        total+=tat[i];
        total_wt+=wt[i];
        total_tat+=tat[i];
printf("\np%d\t\t  %d\t\t    %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
    }
 avg_tat=total/n;
printf("\n\nTotal Waiting Time=%d",total_wt);
```

```
printf("\n\nTotal turnaround Time=%f",total_tat);
printf("\n\nAverage Waiting Time=%f",avg_wt);
printf("\nAverage Turnaround Time=%f\n",avg_tat);
}
```

OUTPUT:



Date:11/03/22

Lab Assignment-5

Aim: WAP in C Language to perform the following operation to

implement SRTF (Shortest Remaining Time First) Algorithm:
Calculate:
1. Waiting time for each process

2. Total Waiting Time
3. Average waiting time
4. Turnaround time for each process 5 Total Turnaround time

CODE:

```
#include <stdio.h> int main()
{

int a[10],b[10],x[10],i,j,smallest,count=0,time,n; double
avg=0,tt=0,end;

printf("enter the number of Processes:\n");

scanf("%d",&n); printf("enter arrival time\n"); for(i=0;i<n;i++)
scanf("%d",&a[i]); printf("enter burst time\n"); for(i=0;i<n;i++)
scanf("%d",&b[i]); for(i=0;i<n;i++)
x[i]=b[i];

b[9]=9999;

for(time=0;count!=n;time++) {

smallest=9; for(i=0;i<n;i++) {

if(a[i]<=time && b[i]<b[smallest] && b[i]>0 )

smallest=i; }
```

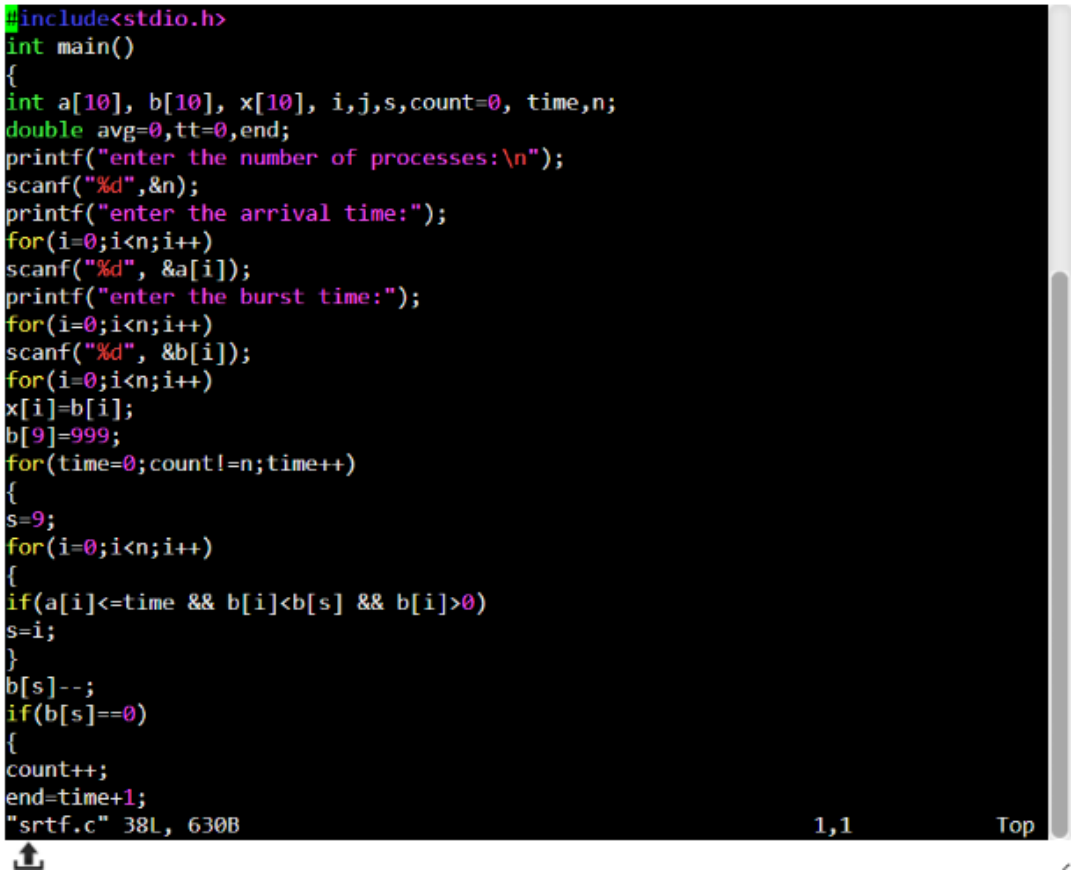
b[smallest]--; if(b[smallest]==0) {

count++;
end=time+1; avg=avg+end-a[smallest]-x[smallest]; tt= tt+end-a[smallest];

} }

printf("\n\nAverage waiting time = %lf\n",avg/n); printf("Average Turnaround time = %lf",tt/n); return 0;

}



```
scanf("%d", &a[i]);
printf("enter the burst time:");
for(i=0;i<n;i++)
scanf("%d", &b[i]);
for(i=0;i<n;i++)
x[i]=b[i];
b[9]=999;
for(time=0;count!=n;time++)
{
s=9;
for(i=0;i<n;i++)
{
if(a[i]<=time && b[i]<b[s] && b[i]>0)
s=i;
}
b[s]--;
if(b[s]==0)
{
count++;
end=time+1;
avg=avg+end-a[s]-x[s];
tt=tt+end-a[s];
}
}
printf("\n\n average waiting time :", avg/n);
printf(" average turn around time:", tt/n);
return 0;
}
```

38,0-1 Bot

OUTPUT-

```
s=i;
}
b[s]--;
if(b[s]==0)
{
count++;
end=time+1;
avg=avg+end-a[s]-x[s];
tt=tt+end-a[s];
}
}
printf("\n\n average waiting time :", avg/n);
printf(" average turn around time:", tt/n);
return 0;
}

"srtf.c" [New] 38L, 630B written
[root@localhost ~]# gcc srtf.c
[root@localhost ~]# ./a.out
enter the number of processes:
3
enter the arrival time:1
2
3
enter the burst time:23
54
76


 average waiting time : average turn around time:[root@localhost ~]#
```

Date:25/03/22

Lab Assignment-6

Aim: WAP in C Language to perform the following operation to

implement Round Robin Algorithm:

Calculate:
1. Waiting time for each process

2. Total Waiting Time
3. Average waiting time
4. Turnaround time for each process

5. Total Turnaround time

ALGORITHM:
Step 1: Start the process

Step 2: Accept the number of processes in the ready Queue and time quantum (or) time slice

Step 3: For each process in the ready Q, assign the process id and accept the CPU burst time

Step 4: Calculate the no. of time slices for each process where No. of time slice for process (n) = burst time process (n)/time slice

Step 5: If the burst time is less than the time slice then the no. of time slices =1.

Step 6: Consider the ready queue is a circular Q, calculate

a) Waiting time for process (n) = waiting time of process(n-1)+ burst time of process(n-1) + the time difference in getting the CPU from process(n-1)

b) Turnaround time for process(n) = waiting time of process(n) + burst time of process(n)+ the time difference in getting CPU from process(n).

Step 7: Calculate c) Average waiting time = Total waiting Time / Number of process d) Average Turnaround time = Total Turnaround Time / Number of process

Step 8: Stop the process

CODE-

```
#include<stdio.h>

int main() {

int count,j,n,time,remain,flag=0,time_quantum;
int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
printf("Enter Total Process:\t ");
scanf("%d",&n);
remain=n;
for(count=0;count<n;count++)
{

printf("Enter Arrival Time and Burst Time for Process Process Number %d :",count+1);

scanf("%d",&at[count]); scanf("%d",&bt[count]);
rt[count]=bt[count];

}
printf("Enter Time Quantum:\t"); scanf("%d",&time_quantum);
printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");
for(time=0,count=0;remain!=0;)
{
```

```
if(rt[count]<=time_quantum && rt[count]>0) {
time+=rt[count]; rt[count]=0; flag=1;
}
else if(rt[count]>0) {
rt[count]-=time_quantum;
time+=time_quantum; }
if(rt[count]==0 && flag==1) {
remain--;
printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-
at[count]- bt[count]);
wait_time+=time-at[count]-bt[count]; turnaround_time+=time-
at[count];
flag=0; }
if(count==n-1) count=0;
else if(at[count+1]<=time) count++;
else count=0;
}
printf("\nAverage Waiting Time= %f\n",wait_time*1.0/n);
printf("Avg Turnaround Time = %f",turnaround_time*1.0/n);
return 0; }
```

```c
#include<stdio.h>
int main()
{
int count,j,n,time,remain,flag=0,time_quanta;
int wait_time=0, turnaround_time=0, at[10],bt[10],rt[10];
printf("enter total processes:\t");
scanf("%d", &n);
remain=n;
for(count=0;count<n;count++)
{
printf("enter arrival time and burst time for process number %d:", count+1);
scanf("%d",&at[count]);
scanf("%d" , &bt[count]);
rt[count]=bt[count];
}
printf("enter the time quanta:\t");
scanf("%d", &time_quanta);
printf("\n\nProcess\t\t turnaround time \t\t waiting time\n\n");
for(time=0;count=0;remain!=0)
{
if(rt[count]<=time_quanta && rt[count]>0)
{
time+=rt[count];
rt[count]=0;
flag=1;
}
else if(rt[count]>0)
{
rt[count]-=time_quanta;
-- INSERT --                                                    1,1           Top
```

```
time+=rt[count];
rt[count]=0;
flag=1;
}
else if(rt[count]>0)
{
rt[count]-=time_quanta;
time+=time_quanta;
}
if(rt[count]==0 && flag==1)
{
remain--;
printf("P[%d]\t\t%d\t\t%d\n", count+1, time-at[count],time-at[count]-bt[count]);
wait_time+=time-at[count]-bt[count];
turnaround_time+=time-at[count];
flag=0;
}
if(count==n-1)
count=0;
else if (at [count+1]<=time)
count++;
else
count=0;
}
printf("\n Average waiting time=%f\n", wait_time*1.0/n);
printf(" average turnaround time=%f", turnaround_time*1.0/n);
return 0;
}

-- INSERT --                                                    51,1          Bot
```

OUTPUT-

```
count=0;
else if (at [count+1]<=time)
count++;
else
count=0;
}
printf("\n Average waiting time=%f\n", wait_time*1.0/n);
printf(" average turnaround time=%f", turnaround_time*1.0/n);
return 0;
}

"rr.c" 51L, 1103B written
[root@localhost ~]# gcc rr.c
[root@localhost ~]# ./a.out
enter total processes:  3
enter arrival time and burst time for process number 1:2
2
enter arrival time and burst time for process number 2:4

5
enter arrival time and burst time for process number 3:6
3
enter the time quanta:  2


Process          turnaround time                 waiting time


 Average waiting time=0.000000
 average turnaround time=0.000000[root@localhost ~]#
```