**ROUND ROBIN CPU Scheduling**

2.b) Write a Program to implement Round Robin CPU Scheduling algorithm.

**Aim: To implement Round Robin CPU Scheduling Program**

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

main()

{

int b[10], pno[10],ts,n,s[10],e[10],w[10],t[10],r[10];

int I,c=0,x=0;

float aw=0,at=0;

printf(“Enter number of processes”);

scanf(“%d”,&n);

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

pno[i]=i+1;

printf(“Enter the time slice”);

scanf(“%d”,&ts);

printf(“Enter the burst time of each process”);

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

scanf(“%d”,&b[i]);

s[0]=0;

x=0;

c=0;

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

{

if(b[i]<ts)

{

e[i]=x+b[i];

r[i]=0;

}

else

{

e[i]=ts+x;

r[i]=b[i]-ts;

}

x=e[i];

s[i+1]=e[i];

t[i]=e[i];

w[i]=s[i];

}

while(c>=0)

{

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

{

if(r[i]!=0)

{

w[i]=w[i]+x-e[i];

if(r[i]<ts)

{

e[i]=x+r[i];

r[i]=0;

}

else

{

e[i]=x+ts;

r[i]=r[i]-ts;

}

x=e[i];

t[i]=e[i];

}

if(r[i]!=0)

c++;

}

c--;

}

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

{

aw=aw+w[i];

at=at+t[i];

}

aw=aw/n;

at=at/n;

printf(“Time slice=%d”,ts);

printf(“\n pno \t bt \t st \t et \t wt \t tat”);

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

printf(“\n%d\t%d\t%d\t%d\t%d\t%d”,pno[i],b[i],s[i],e[i],w[i],t[i]);

printf(“\n Average waiting time=%f”,aw);

printf(“\nAverage turn around time=%f”,at);

}

**Experiment-3)**

**3.a) Implementation of fork (), wait (), exec (), and exit (), System calls.**

**Aim: To implement fork (), wait (), exec (), and exit (), System calls.**

**Program:**

#include<stdio.h>

#include<sys/types.h>

#include<stdlib.h>

#include<unistd.h>

int main()

{

pid\_t p;

int i;

p=fork();

if(p==1)

{

printf("fork error");

exit(-1);

}

else if(p==0)

{

for(i=0;i<5;i++)

{

execlp("ls","ls","-l",NULL);

printf("child process\n");

}

}

else

{

wait(0);

for(i=0;i<5;i++)

{

printf("parent process\n");

}

exit(0);

}

}