1.

#include<unistd.h>

#include<stdlib.h>

#include<iostream>

#include<sys/types.h>

using namespace std;

int main()

{

pid\_t pid;

pid=fork();

if(pid==0)

{

cout<<”pid of child process”<<getpid();

cout<<endl;

cout<<”ppid of child process”<<getppid();

cout<<endl;

}

else if(pid>0)

{

cout<<”pid of parent process”<<getpid();

cout<<endl;

cout<<”ppid of parent process”<<getppid();

cout<<endl;

}

else

{

Cout<<”fork failed”;

}

return 0;

}

2.

#include<unistd.h>

#include<stdlib.h>

#include<iostream>

#include<sched.h>

using namespace std;

int main()

{

int a;

a=sched\_getscheduler(getpid());

switch(a)

{

case SCHED\_OTHER:cout<<”another scheduling policy”<<endl;

break;

case SCHED\_RR:cout<<”round robin scheduler”<<endl;

break;

case SCHED\_FIFO:cout<<”first in first out”<<endl;

break;

}

return 0;

}

3.

#include<unistd.h>

#include<iostream>

#include<sched.h>

using namespace std;

int main()

{

int a;

a=sched\_setscheduler(getpid(),SCHED\_FIFO,0);

if(a==0)

{

Cout<<”priority set”;

}

else

{

Cout<<”priority not set”;

}

return 0;

}

4.

#include<iostream>

#include<cstring>

using namespace std;

class process{

public:

string name;

int arrTime;

int burstTime;

int respTime;

int compTime;

int waitTime;

int turnTime;

};

int main()

{

int i,noproc,avgwaitTime,avgturnTime,sum1=0,sum2=0;

cout<<”number of process=\n”;

cin>>noproc;

process p[ noproc ];

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

{

cout<<”\n enter the process “<<i<<”name =”;

cin>>p[i].name;

cout<<”\n enter the process”<<i<<”arrival time=”;

cin>>p[i].arrTime;

cout<<”\n enter the process”<<i<<”burst time=”;

cin>>p[i].burstTime;

}

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

{

if(i==0)

{

p[i].respTime=p[i].arrTime;

p[i].compTime = p[i].respTime + p[i].burstTime;

p[i].waitTime = p[i].respTime – p[i].arrTime;

p[i].turnTime = p[i].compTime – p[i].arrTime;

}

else

{

P[i].respTime = p[i-1].respTime + p[i-1].burstTime;

P[i].compTime = p[i].respTime + p[i].burstTime;

P[i].waitTime = p[i].respTime= - p[i].arrTime;

P[i].turnTime = p[i].compTime – p[i].arrTime;

}

}

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

{

sum1 = sum1 + p[i].waitTime;

sum2 = sum2 + p[i].turnTime;

}

avgwaitTime = sum1/noproc;

avgturnTime = sum2/noproc;

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

{

cout<<”\n process no”<<i+1;

cout<<”\n name”<<p[i].name;

cout<<”\n arrival time”<<p[i].arrTime;

cout<<”\n burst time”<<p[i].burstTime;

cout<<”\n response time”<<p[i].respTime;

cout<<”\n complete time”<<p[i].compTime;

cout<<”\n waiting time”<<p[i].waitTime;

cout<<”\n turnaround time”<<p[i].turnTime;

cout<<”\n average waiting time”<<avgwaitTime;

cout<<”\n average turnaround time”<<avgturnTime;

return 0;

}