

In this lab session, you will learn implementation of FIFO (non preemptive) CPU scheduling algorithm. The following code reads number of processes their cpu burst and displays average turnaround time and average waiting time. Visual C++ 6.0 IDE is recommended for editing/compiling the program(s).

Implementation of FCFS

```
#include
using namespace std;
const int n=5;
class fifo
{
private:
int cpu[n],arr_t[n],turn_tm[n],wait_tm[n];
int assign[n],comp[n];
float avg_turn_around_time,avg_waiting_time;
public:
void get()
{
for(int i=0;i {
cout<<"enter the cpu burst of p"< 15
cin>>cpu[i];
cout<<"enter the Arrival time of p"< cin>>arr_t[i];
if(i==0)
{
assign[i]=arr_t[i];
comp[i]=arr_t[i]+cpu[i];
}
else
{
assign[i]=comp[i-1];
comp[i]=comp[i-1]+cpu[i];
}
}
cout<<"\n=====
\n";
}
void turn_around_time()
{
float sum=0;
for(int i=0;i {
16
turn_tm[i]=comp[i]-arr_t[i];
sum=sum+turn_tm[i];
}
avg_turn_around_time=sum/n;
```

```

}
void waiting_time()
{
float sum=0.0;
for(int i=0;i {
wait_tm[i]=assign[i]-arr_t[i];
sum=sum+wait_tm[i];
}
avg_waiting_time=sum/n;
}
void show()
{
cout<<"Process \tcpu burst \tArrival Time\twaiting Time/tturn around
time"< for(int i=0;i {
cout<<"p"< urn_tm[i]< 17
cout< }
cout<<"The average waiting time is ="< cout< cout<<"The average turn around time is
="< }
};
int main()
{
fifo f;
f.get();
f.waiting_time();
f.turn_around_time();
f.show();
return 0;
}

```

Exercise

Implement SJF and Priority scheduling algorithm. Read no of processes their arrival time cpu burst and priority from user. The program should compute the average turnaround time and average waiting time.