4B OS Lab-05 Assignment

Muhammad Danish | cs182019 | 4B

CODE OUTPUT SCREENSHOTS:

```
C:\Users\user\Desktop\Semester 4\OS Lab\Lab 05\Assg\cs182019\cs182019_Lab05.exe
CPU process
Process Number : 1
There is 60% space left for CPU process.

IO process
Process Number : 2
There is 20% space left for IO process.

CPU process
Process Number : 3
There is 50% space left for CPU process.

IO process
Process Number : 4
There is 10% space left for IO process.

CPU process
Process Number : 5
There is 40% space left for CPU process.

IO process
Process Number : 6
There is 0% space left for IO process.

CPU process
Process Number : 7
There is 30% space left for CPU process.

CPU process
Process Number : 8
There is 20% space left for CPU process.

CPU process
Process Number : 9
There is 10% space left for CPU process.

CPU process
Process Number : 10
There is 0% space left for CPU process.


--------------------------------
Process exited after 0.08165 seconds with return value 0
Press any key to continue . . .
```

CODE SCREENSHOTS:

cs182019_Lab05.cpp

```
#include <string>
#include <queue>
#include <iostream>
using namespace std;

int cpuRatio=0;
int ioRatio=0;
int fullRatio = cpuRatio +ioRatio;
string process;
int readyQueueProcessNumber=1;

void LTS(queue<string> readyQ)
{
    cout <<"Ready Queue size: "<<readyQ.size()<<endl;
    cout <<"Ready Queue is 100% empty.\n";
    cout << "-----------X Proceses Are ----------X "<<endl;

    refresh:
    while(!readyQ.empty())
    {
        string process = readyQ.front().c_str();
        if(process=="CPU process" && cpuRatio == 70)
        {
            readyQ.pop();
            goto refresh;
        }
        else if(process=="IO process" && ioRatio == 30)
        {
            readyQ.pop();
            goto refresh;
        }
        else if(cpuRatio==70 && ioRatio==30)
        {
            cout << "Ready Queue is full there is no room for more processes.\n";
            // when the ready queue is full the function will terminate.
            break;
        }

        else if ( process=="IO process" && ioRatio < 30)
        {
                cout << process <<"\n";
                cout <<"Process Number : " <<readyQueueProcessNumber<<"\n";
```

Continued ……

cs182019_Lab05.cpp

```
34              cout << "Ready Queue is full there is no room for more processes.\n";
35              // when the ready queue is full the function will terminate.
36              break;
37          }
38
39          else if ( process=="IO process" && ioRatio < 30)
40          {
41                  cout << process <<"\n";
42                  cout <<"Process Number : " <<readyQueueProcessNumber<<"\n";
43                  ioRatio=ioRatio+10;
44                  readyQueueProcessNumber++;
45                  cout<<"There is "<<30-ioRatio<<"% " << "space left for IO process.\n\n";
46                  readyQ.pop();
47          }
48
49          else if (process=="CPU process" && cpuRatio < 70)
50          {
51                  cout << process <<"\n";
52                  cout <<"Process Number : " <<readyQueueProcessNumber<<"\n";
53                  cpuRatio=cpuRatio+10;
54                  readyQueueProcessNumber++;
55                  cout<<"There is "<<70-cpuRatio<<"% " << "space left for CPU process.\n\n";
56                  readyQ.pop();
57
58          }
59
60          else
61          {
62              readyQ.pop();
63          }
64
65      }
66  }
67  int main()
68  {
```

cs182019_Lab05.cpp

```
66  }
67  int main()
68  {
69      queue<string> JobQueue;
70
71      //Inserting into JobQueue.
72      // There are 23 processes in job queue.
73      // LTS will schedule only 10 processes in ratio of 70% for cpu and 30% for IO.
74      // LTS will schedule processes in FCFS mannner....
75          //... but when the respective process(io or cpu) limit is reached it will queue the another type of processes.
76      // there will be always 10 process , 7 cpu bound and 3 io bound. irrespective of processesa arranged in job queue.
77      JobQueue.push("CPU process");
78      JobQueue.push("IO process");
79      JobQueue.push("CPU process");
80      JobQueue.push("IO process");
81      JobQueue.push("CPU process");
82      JobQueue.push("IO process");
83      JobQueue.push("CPU process");
84      JobQueue.push("IO process");
85      JobQueue.push("CPU process");
86      JobQueue.push("IO process");
87      JobQueue.push("CPU process");
88      JobQueue.push("IO process");
89      JobQueue.push("CPU process");
90      JobQueue.push("IO process");
91      JobQueue.push("IO process");
92      JobQueue.push("CPU process");
93      JobQueue.push("IO process");
94      JobQueue.push("CPU process");
95      JobQueue.push("IO process");
96      JobQueue.push("CPU process");
97      JobQueue.push("IO process");
98      JobQueue.push("CPU process");
99      JobQueue.push("IO process");
100
101     //sending job queue in LTS.
102     LTS(JobQueue);
103
104
105     return 0;
106 }
```