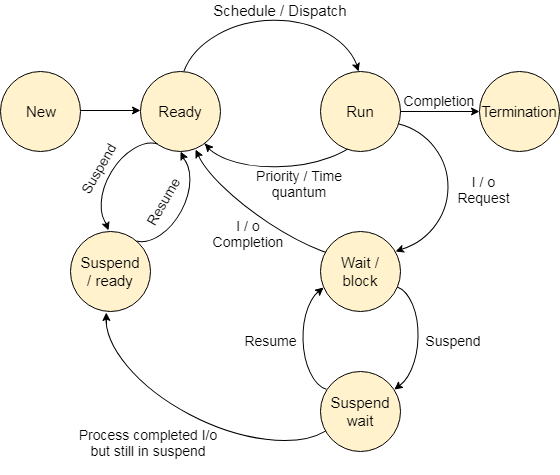
**Lab no 3**

**First Come First Serve Scheduling**

***Objectives:***

* What is FCFS scheduling?
* Implementations of first come first serve in PYTHON.
* Manipulate and find Waiting time, turnaround time, average wait time and average turnaround time

**First Come First Serve (FCFS):** is an operating system scheduling algorithm that automatically executes queued requests and processes in order of their arrival. It is the easiest and simplest CPU scheduling algorithm. In this type of algorithm, processes which request the CPU first get the CPU allocation first. This is managed with a FIFO queue. The full form of FCFS is First Come First Serve. As the process enters the ready queue, its PCB (Process Control Block) is linked with the tail of the queue and, when the CPU becomes free, it should be assigned to the process at the beginning of the queue.



**Advantages of FCFS**

Here, are pros/benefits of using FCFS scheduling algorithm:

* The simplest form of a CPU scheduling algorithm
* Easy to program
* First come first served

**Disadvantages of FCFS**

Here, are cons/ drawbacks of using FCFS scheduling algorithm:

* It is a Non-Preemptive CPU scheduling algorithm, so after the process has been allocated to the CPU, it will never release the CPU until it finishes executing.
* The Average Waiting Time is high.
* Short processes that are at the back of the queue have to wait for the long process at the front to finish.
* Not an ideal technique for time-sharing systems.
* Because of its simplicity, FCFS is not very efficient.

**Implementation:** 

* Input the processes along with their burst time (bt).
* Find waiting time (wt) for all processes.
* As first process that comes need not to wait so Waiting time for process 1 will be 0 i.e. wt.[0] = 0.
* Find waiting time for all other processes i.e. for process i -> wt[i] = bt[i-1] + wt[i-1] .
* Find turnaround time = waiting\_time + burst\_time for all processes.
* Find average waiting time = total\_waiting\_time / no\_of\_processes.
* Similarly, find average turnaround time = total\_turn\_around\_time / no\_of\_processes.

**Important Points:** 

* Non-preemptive
* Average Waiting Time is not optimal
* Cannot utilize resources in parallel: Results in Convoy effect (Consider a situation when many IO bound processes are there and one CPU bound process. The IO bound processes have to wait for CPU bound process when CPU bound process acquires CPU. The IO bound process could have better taken CPU for some time, and then used IO devices).

**Summary:**

* Definition: FCFS is an operating system scheduling algorithm that automatically executes queued requests and processes by order of their arrival
* It supports non-preemptive and pre-emptive scheduling Algorithm.
* FCFS stands for First Come First Serve
* A real-life example of the FCFS method is buying a movie ticket on the ticket counter.
* It is the simplest form of a CPU scheduling algorithm
* It is a Non-Preemptive CPU scheduling algorithm, so after the process has been allocated to the CPU, it will never release the CPU until it finishes executing

***CODE:***

**print("FIRST COME FIRST SERVE SCHEDULLING")**

**n= int(input("Enter number of processes : "))**

**d = dict()**

**for i in range(n):**

**key = "Process"+str(i+1)**

**a = float(input("Enter ready time of process"+str(i+1)+": "))**

**b = float(input("Enter executing time of process"+str(i+1)+": "))**

**l = [] #list**

**l.append(a)**

**l.append(b)**

**d[key] = l**

**d = sorted(d.items(), key=lambda item: item[1][0])**

**ET = []**

**for i in range(len(d)):**

**# first process**

**if(i==0):**

**ET.append(d[i][1][1])**

**# get prevET + newBT**

**else:**

**ET.append(ET[i-1] + d[i][1][1])**

**TAT = []**

**for i in range(len(d)):**

**TAT.append(ET[i] - d[i][1][0])**

**WT = []**

**for i in range(len(d)):**

**WT.append(TAT[i] - d[i][1][1]/1000)**

**avg\_WT = 0**

**for i in WT:**

**avg\_WT +=i**

**avg\_WT = (avg\_WT/n)**

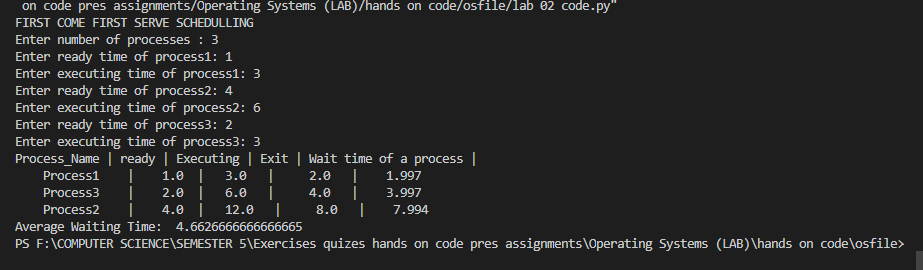
**print("Process\_Name | ready | Executing | Exit | Wait time of a process |")**

**for i in range(n):**

**print(" ",d[i][0]," | ",d[i][1][0]," | ",ET[i]," | ",TAT[i]," | ",WT[i])**

**print("Average Waiting Time: ",avg\_WT)**

***OUTPUT:***

******

**Task:**

* Consider the set of 5 processes whose arrival time and burst time are given below-

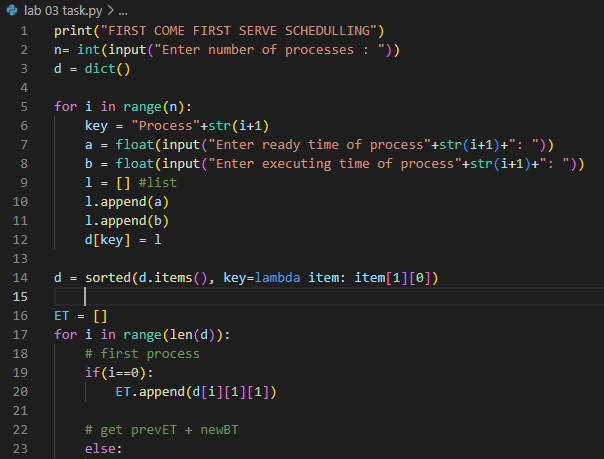
|  |  |  |
| --- | --- | --- |
| **Process Id** | **Arrival time** | **Burst time** |
| P1 | 3 | 4 |
| P2 | 5 | 3 |
| P3 | 0 | 2 |
| P4 | 5 | 1 |
| P5 | 4 | 3 |

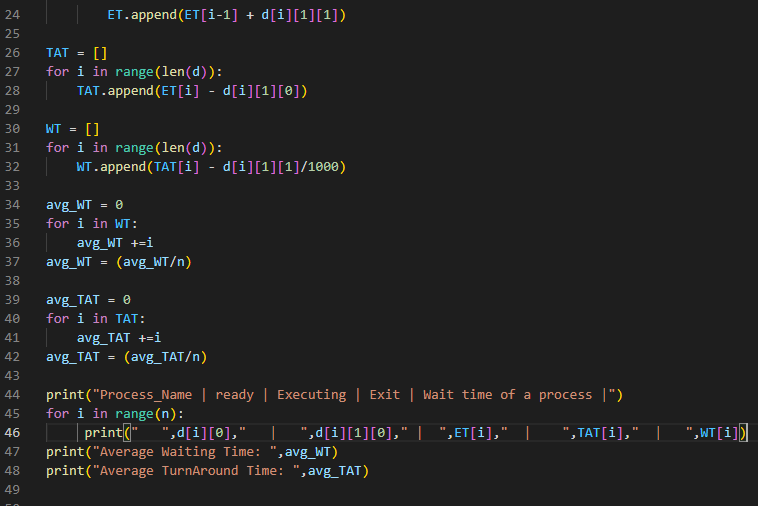
If the CPU scheduling policy is FCFS, write a program to calculate the average waiting time and average turnaround time.

* Create an FCFS scheduling program in which the user enters the number of presses, arrival and burst times at run time.

ANSWER Of Both Tasks:

Code:





Output:

