

**WIA2004 OPERATING SYSTEM**

**LAB 1 PROJECT REPORT**

**CPU**

**GROUP MEMBERS**

| **No.** | **Member Name** | **Matric Number** |
| --- | --- | --- |
| **1.** | **MUHAMMAD IZZAT HAKEEM BIN ROSDI** | **17204856** |
| **2.** | **WAN MUHAMMAD AZIB BIN WAN ABDUL RAHIM** | **17207678** |
| **3.** | **MUHAMMAD AMIRUL AMIN BIN ARMAN** | **17202969** |
| **4.** | **JONE WAN GAN** | **S2031153** |
| **5.** | **LAWRENCE LEROY CHIENG TZE YAO** | **S2018935** |

### **QUESTION**

Write a program to simulate the following non-preemptive CPU 1 scheduling algorithms to find turnaround time and waiting time.

### **OBJECTIVE**

Write a program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time for the above problem.

### **DESCRIPTION**

Assume all the processes arrive at the same time.

### **FCFS CPU SCHEDULING ALGORITHM**

For FCFS scheduling algorithm, read the number of processes/jobs in the system, their CPU burst times. The scheduling is performed on the basis of arrival time of the processes irrespective of their other parameters. Each process will be executed according to its arrival time. Calculate the waiting time and turnaround time of each of the processes accordingly.

### **FCFS**

* + Is a ***non-preemptive*** scheduling algorithm that handles jobs according to their arrival time.
  + lVery simple; implemented using FIFO queue.
  + lGood for batch systems, but unacceptable for interactive systems because users expect a quick response time.
  + lWhen a job enters the system, its PCB is linked to the end of the READY queue and is removed when the processor becomes available.

### **CODE**

def turn\_arnd\_time(process):

tasks = len(process) #number of tasks

tot = 0

for i in range(tasks):

for j in range(i+1): #fcfs so cumulative service time INCLUDING SELF = finish time

tot += process[j][1]

tot -= process[i][0] #minus arrival time

return tot

def wait\_time(process):

tasks = len(process) #number of tasks

tot = 0

for i in range(tasks):

for j in range(i): #fcfs so cumulative service time of PREDECESSORS = start time

tot += process[j][1]

tot -= process[i][0] #minus arrival time

return tot

#extra step: ensure process array is valid (ie equal column length)

def validate(process):

for i in range(len(process)-1):

if len(process[i]) != len(process[i+1]):

return False

return True

# ========== runner code ==========

if \_\_name\_\_ == "\_\_main\_\_":

process = [[0,3], [2,6], [4,4], [6,5], [8,2]]

if validate(process):

tasks = len(process)

turn\_arnd = turn\_arnd\_time(process)

wait = wait\_time(process)

print(f'Turn around time: {turn\_arnd}')

print(f'Average turn around time: {turn\_arnd/tasks}')

print(f'Wait time: {wait}')

print(f'Average wait time: {wait/tasks}')

### **OUTPUT**

