**Exp No.** 8 **Name:** PONNURI ANIRUDDHA

**Date**: 26/09/2022 **Reg No:** RA2112704010015

**FCFS and SJF Scheduling**

**Aim: --**

To implement FCFS (First Come First Serve) and SJF (shortest job first) in CPU scheduling.

**Procedure: --**

FCFS stands for First Come First Serve. In the FCFS scheduling algorithm, the job that arrived first in the ready queue is allocated to the CPU and then the job that came second, and so on. We can say that the ready queue acts as a FIFO (First In First Out) queue thus the arriving jobs/processes are placed at the end of the queue. FCFS is a non-preemptive scheduling algorithm as a process holds the CPU until it either terminates or performs I/O. Thus, if a longer job has been assigned to the CPU then many shorter jobs after it will have to wait. This algorithm is used in most batch operating systems.

The waiting time is the time which the second process is kept in the waiting state so that the first process is moved to the running state and executed. The burst time is the time for execution on the CPU. The turnaround time is the time in which the request for any request has been fulfilled or a job is completed. The completion time is the total time in which the process completes execution from its ready state to the executed state. The algorithms that the FCFS works are as follows:

Turn Around Time = Completion Time - Arrival Time

Waiting Time = Turnaround time - Burst Time

**Algorithm:**

* The first process arrives at the CPU.
* The completion time is calculated so that the waiting time for the next processes are calculated.
* The next processes enter/arrive at the CPU.
* While the process 1 is completing its execution the rest of the processes are kept for wait.
* The turnaround time is calculated by subtracting completion time of the process with the arrival time at the CPU.
* The total waiting time is calculated by subtracting the turnaround time and the burst time.

**Shortest Job First (SJF):**

Shortest job first (SJF) is a scheduling process that selects the waiting process with the smallest execution time to execute next. This scheduling method may or may not be preemptive. Significantly reduces the average waiting time for other processes waiting to be executed. The full form of SJF is Shortest Job First.

**Algorithm:**

* Sort all the processes according to the arrival time.
* Then select that process that has minimum arrival time and minimum Burst time.
* After completion of the process make a pool of processes that arrives afterward till the completion of the previous process and select that process among the pool which is having minimum Burst time.

**Code (FCFS)**:

*def findWaitingTime(processes, n, bt, wt, at):*

*service\_time = [0] \* n*

*service\_time[0] = 0*

*wt[0] = 0*

*# calculating waiting time*

*for i in range(1, n):*

*# Add burst time of previous processes*

*service\_time[i] = (service\_time[i - 1] +*

*bt[i - 1])*

*# Find waiting time for current*

*# process = sum - at[i]*

*wt[i] = service\_time[i] - at[i]*

*# If waiting time for a process is in*

*# negative that means it is already*

*# in the ready queue before CPU becomes*

*# idle so its waiting time is 0*

*if (wt[i] < 0):*

*wt[i] = 0*

*# Function to calculate turn around time*

*def findTurnAroundTime(processes, n, bt, wt, tat):*

*# Calculating turnaround time by*

*# adding bt[i] + wt[i]*

*for i in range(n):*

*tat[i] = bt[i] + wt[i]*

*# Function to calculate average waiting*

*# and turn-around times.*

*def findavgTime(processes, n, bt, at):*

*wt = [0] \* n*

*tat = [0] \* n*

*# Function to find waiting time*

*# of all processes*

*findWaitingTime(processes, n, bt, wt, at)*

*# Function to find turn around time for*

*# all processes*

*findTurnAroundTime(processes, n, bt, wt, tat)*

*# Display processes along with all details*

*print("Processes Burst Time Arrival Time Waiting",*

*"Time Turn-Around Time Completion Time \n")*

*total\_wt = 0*

*total\_tat = 0*

*for i in range(n):*

*total\_wt = total\_wt + wt[i]*

*total\_tat = total\_tat + tat[i]*

*compl\_time = tat[i] + at[i]*

*print(" ", i + 1, "\t\t", bt[i], "\t\t", at[i],*

*"\t\t", wt[i], "\t\t ", tat[i], "\t\t ", compl\_time)*

*print("Average waiting time = %.5f "%(total\_wt /n))*

*print("\nAverage turn around time = ", total\_tat / n)*

*# Driver code*

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

*# Process id's*

*processes = [1, 2, 3]*

*n = 3*

*# Burst time of all processes*

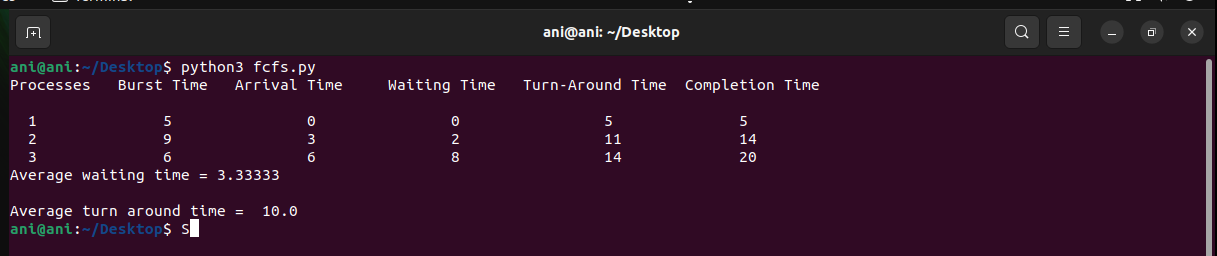
*burst\_time = [5, 9, 6]*

*# Arrival time of all processes*

*arrival\_time = [0, 3, 6]*

*findavgTime(processes, n, burst\_time,arrival\_time)*

**OUTPUT (FCFS): --**



**Code (SJF): --**

*def findWaitingTime(processes, n, wt):*

*rt = [0] \* n*

*# Copy the burst time into rt[]*

*for i in range(n):*

*rt[i] = processes[i][1]*

*complete = 0*

*t = 0*

*minm = 999999999*

*short = 0*

*check = False*

*# Process until all processes gets*

*# completed*

*while (complete != n):*

*# Find process with minimum remaining*

*# time among the processes that*

*# arrives till the current time`*

*for j in range(n):*

*if ((processes[j][2] <= t) and*

*(rt[j] < minm) and rt[j] > 0):*

*minm = rt[j]*

*short = j*

*check = True*

*if (check == False):*

*t += 1*

*continue*

*# Reduce remaining time by one*

*rt[short] -= 1*

*# Update minimum*

*minm = rt[short]*

*if (minm == 0):*

*minm = 999999999*

*# If a process gets completely*

*# executed*

*if (rt[short] == 0):*

*# Increment complete*

*complete += 1*

*check = False*

*# Find finish time of current*

*# process*

*fint = t + 1*

*# Calculate waiting time*

*wt[short] = (fint - proc[short][1] -*

*proc[short][2])*

*if (wt[short] < 0):*

*wt[short] = 0*

*# Increment time*

*t += 1*

*# Function to calculate turn around time*

*def findTurnAroundTime(processes, n, wt, tat):*

*# Calculating turnaround time*

*for i in range(n):*

*tat[i] = processes[i][1] + wt[i]*

*# Function to calculate average waiting*

*# and turn-around times.*

*def findavgTime(processes, n):*

*wt = [0] \* n*

*tat = [0] \* n*

*# Function to find waiting time*

*# of all processes*

*findWaitingTime(processes, n, wt)*

*# Function to find turn around time*

*# for all processes*

*findTurnAroundTime(processes, n, wt, tat)*

*# Display processes along with all details*

*print("Processes Burst Time Waiting",*

*"Time Turn-Around Time")*

*total\_wt = 0*

*total\_tat = 0*

*for i in range(n):*

*total\_wt = total\_wt + wt[i]*

*total\_tat = total\_tat + tat[i]*

*print(" ", processes[i][0], " ",*

*processes[i][1], " ",*

*wt[i], " ", tat[i])*

*print("Average waiting time = %.5f "%(total\_wt /n) )*

*print("Average turn around time = ", total\_tat / n)*

*# Driver code*

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

*# Process id's*

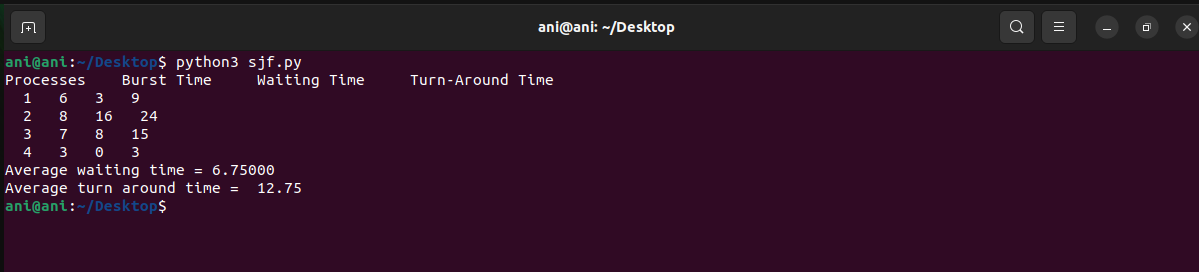
*proc = [[1, 6, 1], [2, 8, 1],*

*[3, 7, 2], [4, 3, 3]]*

*n = 4*

*findavgTime(proc, n)*

**OUTPUT (SJF): --**



**Result:**

The code has been implemented for FCFS ( First come first serve ) type of CPU

Scheduling and SJF ( Shortest Job First ) type of CPU Scheduling.