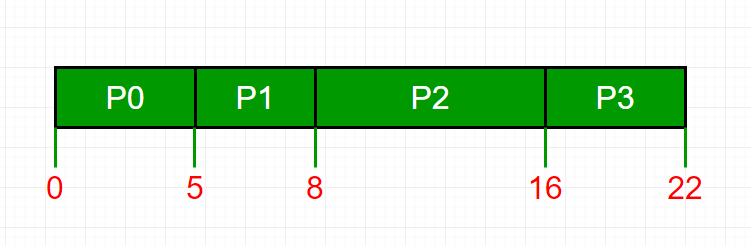
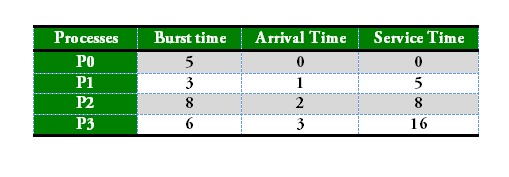
**Program for FCFS CPU Scheduling | Set 2 (Processes with different arrival times)**

We have already discussed [FCFS Scheduling of processes with same arrival time](https://www.geeksforgeeks.org/program-fcfs-scheduling-set-1/). In this post, scenarios, when processes have different arrival times, are discussed. Given n processes with their burst times and arrival times, the task is to find average waiting time and an average turn around time using FCFS scheduling algorithm.   
FIFO simply queues processes in the order they arrive in the ready queue. Here, the process that comes first will be executed first and next process will start only after the previous gets fully executed.

1. Completion Time: Time at which the process completes its execution.
2. Turn Around Time: Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival Time
3. Waiting Time(W.T): Time Difference between turn around time and burst time.   
   Waiting Time = Turn Around Time – Burst Time.



Process Wait Time : Service Time - Arrival Time

P0 0 - 0 = 0

P1 5 - 1 = 4

P2 8 - 2 = 6

P3 16 - 3 = 13

Average Wait Time: (0 + 4 + 6 + 13) / 4 = 5.75

**Service Time:** Service time means amount of time after which a process can start execution. It is summation of burst time of previous processes (Processes that came before)

**Changes in code as compare to code of FCFS with same arrival time:**   
To find waiting time: Time taken by all processes before the current process to be started (i.e. burst time of all previous processes) – arrival time of current process   
**wait\_time[i] = (bt[0] + bt[1] +…… bt[i-1] ) – arrival\_time[i]**

**Implementation:**

1- Input the processes along with their burst time(bt)

and arrival time(at)

2- Find waiting time for all other processes i.e. for

a given process i:

wt[i] = (bt[0] + bt[1] +...... bt[i-1]) - at[i]

3- Now find **turn around time**

= waiting\_time + burst\_time for all processes

4- **Average waiting time** =

total\_waiting\_time / no\_of\_processes

5- **Average turn around time** =

total\_turn\_around\_time / no\_of\_processes

|  |
| --- |
| // Java program for implementation of FCFS  // scheduling with different arrival time  **public** **class** GFG{      // Function to find the waiting time for all  // processes  **static** **void** findWaitingTime(**int** processes[], **int** n, **int** bt[], **int** wt[], **int** at[])  {  **int** service\_time[] = **new** **int**[n];      service\_time[0] = at[0];      wt[0] = 0;        // calculating waiting time  **for** (**int** i = 1; i < n ; i++)      {          //representing wasted time in queue  **int** wasted=0;          // 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          // wasted time is basically time for process to wait after a process is over  **if** (wt[i] < 0) {              wasted = Math.abs(wt[i]);              wt[i] = 0;          }          //Add wasted time          service\_time[i] = service\_time[i] + wasted;      }  }    // Function to calculate turn around time  **static** **void** findTurnAroundTime(**int** processes[], **int** n, **int** bt[],  **int** wt[], **int** tat[])  {      // Calculating turnaround time by adding bt[i] + wt[i]  **for** (**int** i = 0; i < n ; i++)          tat[i] = bt[i] + wt[i];  }    // Function to calculate average waiting and turn-around  // times.  **static** **void** findavgTime(**int** processes[], **int** n, **int** bt[], **int** at[])  {  **int** wt[] = **new** **int**[n], tat[] = **new** **int**[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      System.out.print("Processes " + " Burst Time " + " Arrival Time "          + " Waiting Time " + " Turn-Around Time "          + " Completion Time \n");  **int** total\_wt = 0, total\_tat = 0;  **for** (**int** i = 0 ; i < n ; i++)      {          total\_wt = total\_wt + wt[i];          total\_tat = total\_tat + tat[i];  **int** compl\_time = tat[i] + at[i];          System.out.println(i+1 + "\t\t" + bt[i] + "\t\t"              + at[i] + "\t\t" + wt[i] + "\t\t "              + tat[i] + "\t\t " + compl\_time);      }        System.out.print("Average waiting time = "          + (**float**)total\_wt / (**float**)n);      System.out.print("\nAverage turn around time = "          + (**float**)total\_tat / (**float**)n);  }    // Driver code    **public** **static** **void** main(String args[]) {          // Process id's  **int** processes[] = {1, 2, 3};  **int** n = processes.length;        // Burst time of all processes  **int** burst\_time[] = {5, 9, 6};        // Arrival time of all processes  **int** arrival\_time[] = {0, 3, 6};        findavgTime(processes, n, burst\_time, arrival\_time);        }  }    /\*This code is contributed by PrinciRaj1992\*/ |

**Output:**

Processes Burst Time Arrival Time Waiting Time Turn-Around Time Completion Time

1 5 0 0 5 5

2 9 3 2 11 14

3 6 6 8 14 20

Average waiting time = 3.33333

Average turn around time = 10.0