First Come First Serve Scheduling

In the "First come first serve" scheduling algorithm, as the name suggests, the process which arrives first, gets executed first, or we can say that the process which requests the CPU first, gets the CPU allocated first.

- First Come First Serve, is just like **FIFO**(First in First out) Queue data structure, where the data element which is added to the queue first, is the one who leaves the queue first.
- This is used in Batch Systems.
- It's **easy to understand and implement** programmatically, using a Queue data structure, where a new process enters through the **tail** of the queue, and the scheduler selects process from the **head** of the queue.
- A perfect real life example of FCFS scheduling is **buying tickets at ticket counter**.

Calculating Average Waiting Time

For every scheduling algorithm, **Average waiting time** is a crucial parameter to judge it's performance.

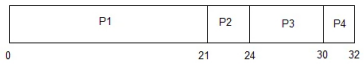
AWT or Average waiting time is the average of the waiting times of the processes in the queue, waiting for the scheduler to pick them for execution.

Lower the Average Waiting Time, better the scheduling algorithm.

Consider the processes P1, P2, P3, P4 given in the below table, arrives for execution in the same order, with **Arrival Time** `0`, and given **Burst Time**, let's find the average waiting time using the FCFS scheduling algorithm.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

The average waiting time will be = (0 + 21 + 24 + 30)/4 = 18.75 ms



This is the GANTT chart for the above processes

The average waiting time will be `18.75 ms`

For the above given proccesses, first **P1** will be provided with the CPU resources,

- Hence, waiting time for **P1** will be `0`
- **P1** requires `21 ms` for completion, hence waiting time for **P2** will be `21 ms`
- Similarly, waiting time for process **P3** will be execution time of **P1** + execution time for **P2**, which will be `(21 + 3) ms` = `24 ms`.
- For process **P4** it will be the sum of execution times of **P1**, **P2** and **P3**.

The **GANTT chart** above perfectly represents the waiting time for each process.

Problems with FCFS Scheduling

Below we have a few shortcomings or problems with the FCFS scheduling algorithm:

1. It is **Non Pre-emptive** algorithm, which means the **process priority** doesn't matter.
   If a process with very least priority is being executed, more like **daily routine backup** process, which takes more time, and all of a sudden some other high priority process arrives, like **interrupt to avoid system crash**, the high priority process will have to wait, and hence in this case, the system will crash, just because of improper process scheduling.
2. Not optimal Average Waiting Time.
3. Resources utilization in parallel is not possible, which leads to **Convoy Effect**, and hence poor resource(CPU, I/O etc) utilization.

What is Convoy Effect?

Convoy Effect is a situation where many processes, who need to use a resource for short time are blocked by one process holding that resource for a long time.

This essentially leads to poort utilization of resources and hence poor performance.

Program for FCFS Scheduling

Here we have a simple C++ program for processes with **arrival time** as `0`.

If you are not familiar with C++ language, we would recommend you to first Learn C++ language.

In the program, we will be calculating the **Average waiting time** and **Average turn around time** for a given `array` of **Burst times** for the list of processes.

```
    // calculating waiting time
    for (int i = 1; i < n ; i++)
    {
        wt[i] =  bt[i-1] + wt[i-1];
    }
}

// function to calculate turn around time
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 time
```

Output:

```
Processes   Burst time  Waiting time  Turn around time
    1          21          0              21
    2              3          21             24
    3              6          24             30
    4              2          30             32
Average waiting time = 18.75
Average turn around time = 26.75
```

Here we have simple formulae for calculating various times for given processes:

Completion Time: Time taken for the execution to complete, starting from arrival time.

Turn Around Time: Time taken to complete after arrival. In simple words, it is the difference between the Completion time and the Arrival time.

Waiting Time: Total time the process has to wait before it's execution begins. It is the difference between the Turn Around time and the Burst time of the process.

For the program above, we have considered the **arrival time** to be `0` for all the processes, try to implement a program with variable arrival times.

← Prev

Next →

Operating System
Computer Network
Computer Architecture
More...