

PROCESS SCHEDULER

A DESKTOP APPLICATION TO VISUALIZE OS JOB/PROCESS SCHEDULING ALGORITHMS.

Developed By

Parth Prajapati [18BCP076]

Ritik Patel [18BCP092]

Nisarg Kapkar [18BCP069]

Purv Patel [18BCP088]

Pandit Deendayal Petroleum University

First Come First Serve (FCFS)

Introduction

Simplest scheduling algorithm that schedules according to arrival times of processes. FCFS scheduling algorithm states that the process that requests the CPU first is allocated the CPU first. It is implemented by using the FIFO queue. When a process enters the ready queue, its PCB is linked to

the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue. The running process is then removed from the queue. FCFS is a non-preemptive scheduling algorithm.

Advantages

- It is simple and easy to understand.

Disadvantages

- The process with less execution time suffers i.e. waiting time is often quite long and favors CPU Bound process then I/O bound process.
- Now, suppose the first process has a large burst time, and other processes have less burst time, then the processes will have to wait more unnecessarily, this will result in more average waiting time (Convoy Effect), which results in lower CPU and device utilization.

Snapshots



Shortest Job First (SJF)

Introduction

It is an algorithm in which the process having the smallest execution time is chosen for the next execution. This scheduling method can be

preemptive or non-preemptive. It significantly reduces the average waiting time for other processes awaiting execution.

Advantages

- Shortest jobs are favored.
- It is probably optimal, in that it gives the minimum average waiting time for a given set of processes.

Disadvantages

- SJF may cause starvation if shorter processes keep coming. This problem is solved by aging.
- It cannot be implemented at the level of short term CPU scheduling.

Snapshots



Shortest Remaining Time First (SRTF)

Introduction

This Algorithm is the preemptive version of SJF scheduling. In SRTF, the execution of the process can be stopped after a certain amount of time. At the arrival of every process, the short term scheduler schedules the process with the least remaining burst time among the list of available processes and the running process. Once all the processes are available in the ready queue.

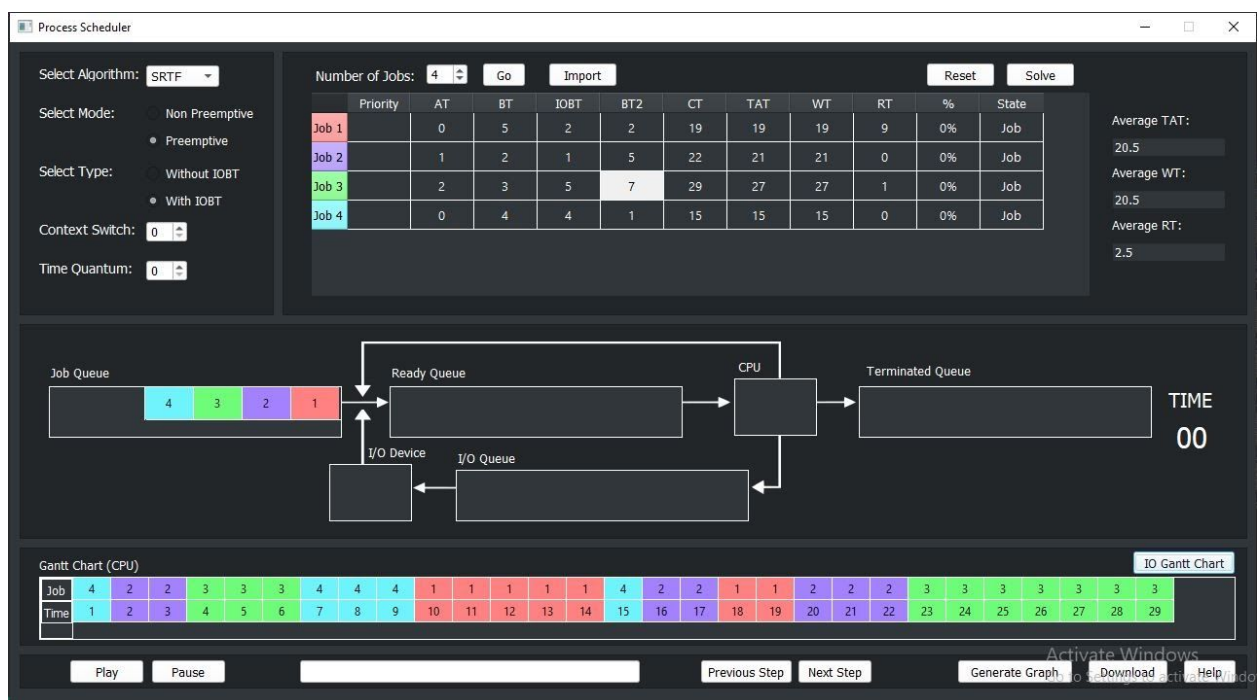
Advantages

- Shortest jobs are favored.
- It is probably optimal, in that it gives the minimum average waiting time for a given set of processes.

Disadvantages

- SRTF may cause starvation if shorter processes keep coming. This problem is solved by aging.
- It cannot be implemented at the level of short term CPU scheduling.

Snapshots



Round Robin (RR)

Introduction

It is the preemptive process scheduling algorithm. Each process is provided a fixed time to execute, it is called a quantum.

Once a process is executed for a given time period, it is preempted and another process executes for a given time period. Context switching is used to save states of preempted processes

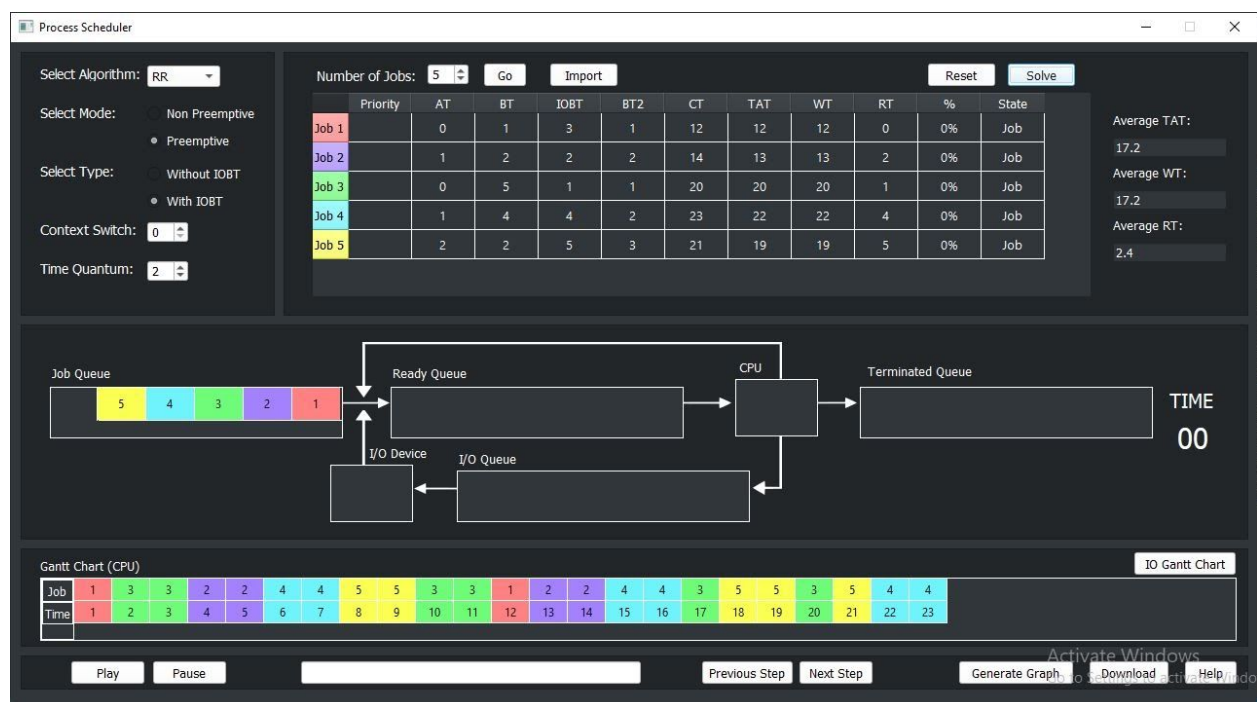
Advantages

- Every process gets an equal share of the CPU.
- RR is cyclic in nature, so there is no starvation.

Disadvantages

- Setting the quantum too short increases the overhead and lowers the CPU efficiency, but setting it too long leads to poor responses to short processes.
- The average waiting time under the RR policy is often long.

Snapshots



Longest Job First (LJF)

Introduction

It is similar to the SJF scheduling algorithm. But, in this scheduling algorithm, we give priority to the process having the longest burst

time. This is non-preemptive in nature i.e., when any process starts executing, can't be interrupted before complete execution.

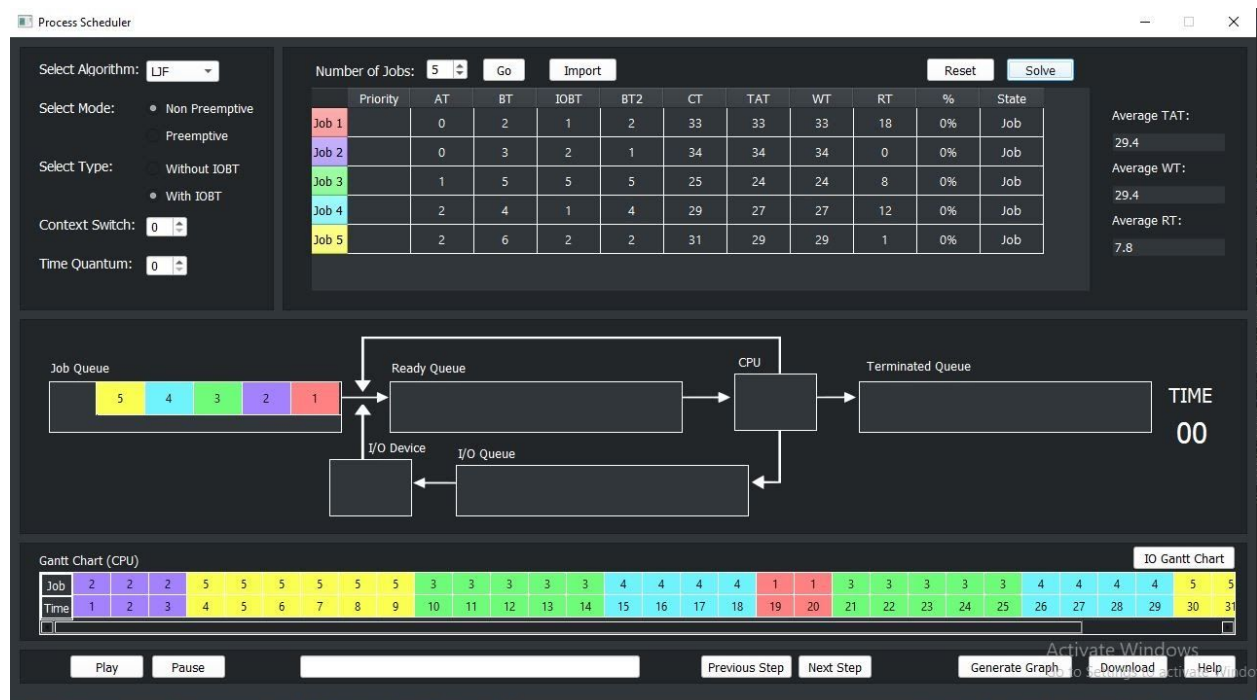
Advantages

- No process can complete until the longest job also reaches its completion.
- All the processes approximately finish at the same time.

Disadvantages

- The waiting time is high and processes with smaller burst time may starve for the CPU.

Snapshots



Longest Remaining Time First (LRTF)

Introduction

This Algorithm is the preemptive version of LJF scheduling. In this scheduling algorithm, we find the process with the maximum remaining time and then process it. We check for the maximum remaining time after some interval of time(say 1 unit each) to check if another process having more Burst Time arrived up to that time.

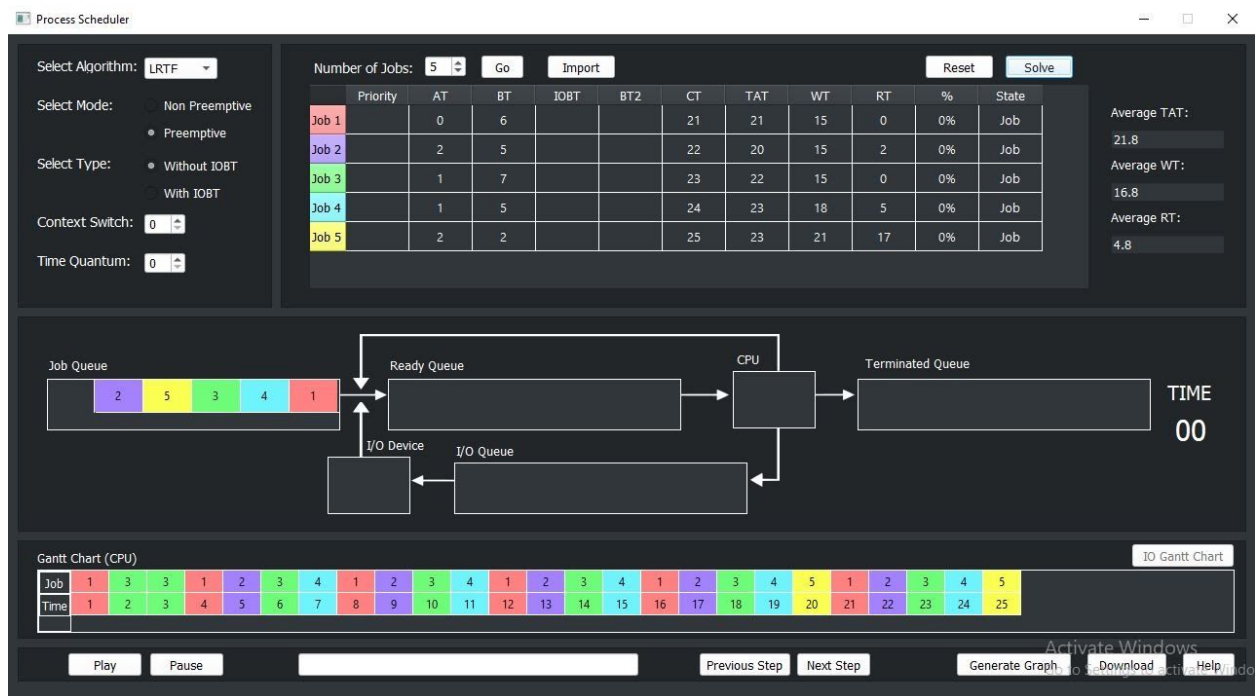
Advantages

- No process can complete until the longest job also reaches its completion.
- All the processes approximately finish at the same time.

Disadvantages

- The waiting time is high and processes with smaller burst time may starve for the CPU.

Snapshots



Priority Based Scheduling

Introduction

In this scheduling, processes are scheduled according to their priorities, i.e., the highest priority process is scheduled first. If the priorities of two processes match, then schedule according to arrival time. Here starvation of process is possible.

Advantages

- This provides a good mechanism where the relative importance of each process may be precisely defined.

Disadvantages

- If high priority processes use up a lot of CPU time, lower priority processes may starve and be postponed indefinitely. The situation where a process never gets scheduled to run is called Starvation.
- Another problem is deciding which process gets which priority level assigned to it.

Snapshots

