**Operating System Design – Project 1**

***Title of your project - CPU Scheduling***

**Team**

1. *Rishwanth Reddy Baddipadaga, 11515020*
2. *Amit Reddy Baddam, 11507588*
3. *Roja Kamble, 11454258*
4. *Pranavanath Reddy Jaggari, 11503083*

**Introduction**

**[a]** ***Overview of the project***

Our project CPU SCHEDULING primarily focuses on demonstrating a simulator for process scheduling and the way different scheduling algorithms behave.

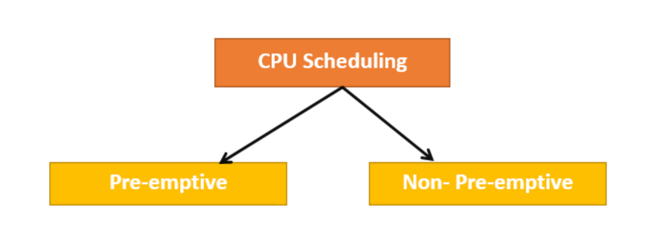
Initially, computers were able to allocate resources only one task at a time and the possibility of allocating resources to multiple processes at a time is done by the introduction of the CPU Scheduling.

It is a crucial job in accomplishing the operating system design as it facilitates by distributing the CPU time among different processes at the same time. The primary objective of an efficient operating system is to allocate the maximum number of processes to run at the same time to extract the best use of the processor’s time. It is quite dependent on managing the CPU by efficiently managing it’s time with the use of suitable algorithms. The scheduling algorithms should satisfy the goals of process scheduling.

**Why CPU Scheduling?**

* The process of choosing which process will use the CPU while another is waiting is known as CPU scheduling.
* CPU scheduling's primary responsibility is to make sure that anytime the CPU is idle, the OS at least chooses one of the tasks in the ready queue to be executed.

**Types of CPU Scheduling:**



**Preemptive Scheduling:** The tasks are often given according to their priority in preemptive scheduling.

**Non-Preemptive Scheduling:** The CPU is assigned to a certain process in this type of scheduling method.

**Factors determining Preemptive or Non-Preemptive:**

* A process switches from the running to the waiting state.
* Specific process switches from the running state to the ready state.
* Specific process switches from the waiting state to the ready state.
* Process finished its execution and terminated.

The scheduling is said to as non-preemptive when only criteria 1 and 4 are present.

The rest of the schedule is preemptive.

**In achieving the goals of this project, we have the following objectives:**

* To demonstrate CPU SCHEDULING by using different scheduling algorithms.
* To be able to simulate the behavior of the scheduling algorithms.

**[b]** ***What problem being solved?***

We are solving CPU Scheduling for the given processes using different scheduling algorithms.

**[c]** ***Why is it important?***

To organize and manage the tasks and to do multitasking with minimum throughput and minimum response time based on the scheduling criteria.

Diagram

Description automatically generated

***Note:***

**Important terms used in CPU scheduling:**

* **Burst Time/Execution Time:** This is the amount of time needed for the process to finish processing. It's also referred to as running time.
* **Arrival Time:** the time a process enters a condition of readiness.
* **Finish Time:** the time a process is finished and leaves a system.
* **Multiprogramming:** The ability to run many programs simultaneously in memory.
* **Jobs:** This sort of application does not involve any user interaction.
* **User:** It is a program that allows for user involvement.
* **Process:** It serves as reference to both the user's and the jobs.
* **CPU/IO burst cycle:** Process execution that alternates between CPU and I/O activity is known as a CPU/IO burst cycle. Typically, CPU times are faster than I/O times.
* **Context Switch:** Technique to move a process between states, so that CPUs can carry out their operations. It is a procedure for saving the context (state) of the previous process (suspend) and loading it into the new process (resume).
* **Alpha:** The relative weight of recent data versus past data is determined by alpha.
* **Tau:** Predicts length of next burst, based on past measurement of burst times for the process.

**This project deals with 5 Algorithms for scheduling which are:**

* First-Come, First-Served FCFS Scheduling
* Shortest-Job-First SJF Scheduling
* Shortest Remaining Time
* Priority Scheduling
* Round Robin Scheduling

**Background**

**[a]** ***What does one need to know to understand the problem?***

In general, one need to possess the knowledge of processes, operating systems, terminologies, and scheduling algorithm’s mechanism. Also, the importance of scheduling algorithm in optimizing the operating systems performance and the results based on choosing a specific algorithm.

**Implementation**

**[a] *What are the solutions to your problem?***

* We are not visualizing or building any UI for this project.
* The queue is represented as time units at a time.
* Understanding the problem, inputs, and expected outputs.

The user is free to select any input variables for the job creation like:

1. The quantity of jobs.
2. The degree of multi-programming.
3. A job's total number of bursts.
4. ranges for CPU and IO bursts.
5. Priority.
6. Range of arrival times.
7. The change in context i.e., context switch.
8. Alpha and Tau. (If required)

**[b] *How do you implement?***

* Programming language: Java
* Operating system: Mac OS/Windows
* Editor: VS Code
* Logic Hook: Algorithms

Understanding the algorithms and applying to build logic with proper analyzed inputs and verifying the expected output.

***i. Programming language for implementation.***

* Programming language: Java

***ii. Operating system to test the project.***

* Operating system: Mac OS/Windows/Linux

**[c] *Test cases.***

* We tested on different input size, different values for determining factors alpha, tau and noted the response time for the algorithm. [Say 5 captures]
* We tried on different range of arrival times.
* Almost covered the input variables defined above for testing.
* Learnt the behavior of algorithms by testing for each one of them.
* Output varies, we are trying to predict prime algorithm with maximum throughput and minimum response time.

**Experimental Results**

**[a] *Any results or output in graphs or tables or figures that shows results of   
improvements/implementations.***

**Test case - 1**

***Text

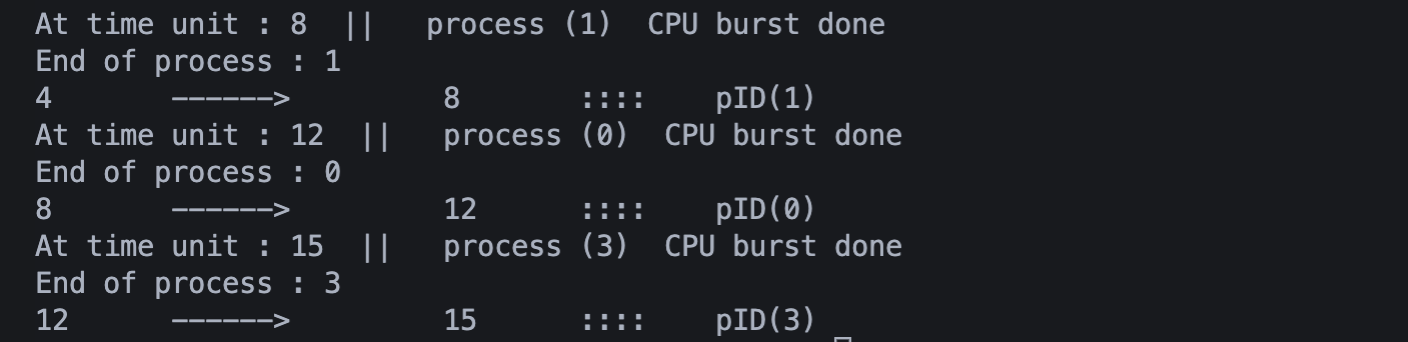
Description automatically generated***

**Text

Description automatically generated**

***Text

Description automatically generated***

******

**Test case – 2**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**[b] *Interpretation of the results.***

The program asks for the context switch, number of processes, Degree of Multiprogramming, number of bursts, range of CPU bursts, range of I/O bursts, range of priority, range of initial arrival times, initial Tau, and alpha. After giving the inputs, we get status of the processes before performing the algorithm on the processes. We have given an option to enter again so that the all the CPU scheduling algorithms are performed on the given data and at the end we get the optimal solution for all the processes we have given in the initial step.

**Conclusion**

**[a] *What have been accomplished and what is still left to be done?***

The primary requirement for any efficient operating system is employing a best suitable scheduling algorithm to finish the job but it’s a fact that there’s no single algorithm that always does the job. Our project gives an insight on how the process scheduling works and this helps us on understanding the pros & cons in using them. The results shows that a clever management is the key here and the simulation shows the requirement of specific scheduling algorithms in each different scenario. We knew that there’s always room for improvement and we firmly believe that for a faster and better optimized operating system, new scheduling algorithms must come into place where it would minimize the bad outcomes that we are experiencing and derive the best use.  
  
**References**

* <https://www.guru99.com/cpu-scheduling-algorithms.html#:~:text=What%20is%20CPU%20Scheduling%3F,the%20ready%20queue%20for%20execution>.
* <https://www.geeksforgeeks.org/difference-between-swapping-and-context-switching/>
* <https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/5_CPU_Scheduling.html>