# TATYASAHEB KORE INSTITUTE OF ENGINEERING AND TECHNOLOGY WARANANAGAR- 416 113

Department of Computer Science & Engineering Academic Year (2021-22)



# A

**Mini Project Synopsis On**

**“PROCESS SCHEDULING”**

**Submitted By,** Group Number: G3

Group Members:

|  |  |  |  |
| --- | --- | --- | --- |
| Roll No | Name | Email ID | Mobile No |
| 105 | Namrata . M . Medidar | medidarnamrata2002@gmail.com | 9067440447 |
| 116 | Sanyogita . M . Patil | sanyogitapatil1508@gmail.com | 9359148165 |
| 119 | Susmita . M . Janawade | suishijanawade@gmail.com | 8308658313 |

Under the Guidance of,

Guide Name: - Prof.Mrs.B.A.Chougule

# Introduction

# CPU scheduling is the basis of multi programmed operating systems. By switching the CPU among processes, the operating system can make the computer more productive. This project introduce basic CPU-scheduling concepts and present several CPU-scheduling algorithms and also consider the problem of selecting an algorithm for a particular system.

# The success of CPU scheduling depends on an observed property of processes. Process execution consists of a cycle of CPU execution and I/0 wait. Processes alternate between these two states. Process execution begins with a CPU burst. That is followed by an I/O burst, which is followed by another CPU burst, then another I/0 burst, and so on. Eventually, the final CPU burst ends with a system request to terminate execution.

# Whenever the CPU becomes idle, the operating system must select one of the processes in the ready queue to be executed. The selection process is carried out by the short-term scheduler (or CPU scheduler). The scheduler selects a process from the processes in memory that are ready to execute and allocates the CPU to that process.

CPU scheduling decision may takes place under the following circumstances:

1.When a process switches from the running state to waiting state.

2.When a process switches from the running state to the ready state.

3. When a process switches from the waiting state to the ready state.

4. When a process terminates.

# Problem Statement

# In a single-processor system, only one process can run at a time; any others must wait until the CPU is free and can be rescheduled. The objective of multiprogramming is to have some process running at all times, to maximize CPU utilization. The idea is relatively simple. A process is executed until it must wait, typically for the completion of some I/O request. In a simple computer system, the CPU then just sits idle. All this waiting time is wasted; no useful work is accomplished. With multiprogramming, here try to use this time productively.

# Thus the project overcomes this problem using scheduling algorithms by scheduling the processes. Several processes are kept in memory at one time. When one process has to wait, the operating system takes the CPU away from that process and gives the CPU to another process. This pattern continues. Every time one process has to wait, another process can take over use of the CPU. Scheduling of this kind is a fundamental operating-system function.

# Requirement Analysis

# A] Software Requirement:

# 1. Operating System – Windows

# 2. Software - Codeblocks

# 3. Language C

# Hardware Requirement:

# 

# Personal Computer/Laptop

# Methodology to Solve the Problem

**Input:**

Input is process ID, Arrival Time, Burst Time, Quantum/Time slice.

Then select the type of algorithm for scheduling.

# Output:

# When user gives the input that is process details and after selecting the type of algorithm for scheduling, then the output contains the waiting time, turnaround time for each process and also the average waiting and turnaround time for selected type of scheduling algorithm.

.

# 

# Flowchart

**Start**

# 

**List of scheduling** **algorithms**

**1 1**1 **1**

**First Come First Serve**

**Shortest Job First**

**Round Robin**

**2** **2**

**3**

# 

**select any one algorithm**

# 

**Input**

**1.Number of processes**

**2.Process id**

**3.Arrival Time**

**4.Burst Time**

**Input**

**1.Number of processes**

**2.Process id**

**3.Arrival Time**

**4.Burst Time**

**Input**

**1.Number of processes**

**2.Process id**

**3.Arrival Time**

**4.Burst Time**

**5.Time slice**

# 

**Output**

**1.Processes**

**2.Burst Time**

**3.Waiting Time**

**4.Turnarround Time**

**5.Avg.Turnarround Time**

**6.Avg.Waiting Time**

# Data Structure

# Linear Queue

# Circular Queue

1. **References**
2. Operating Systems Concepts

8th Edition by – Silberschatz , Galvin, Gagne

1. www.geeksforgeeks.org

Prof.Mrs.B.A.Chougule

# (Project Guide)

**Signature and recommendation by** Dr.G.V.Patil

**Synopsis Approval Committee (HOD CSE Department)**