VISION 2030 رؤيـة

جامعة شقراء
Shaqra University

PROJECT OF:

CPU SCHEDULING ALGORITHMS

Prepared For :

Done By : Rahaf Al-swailem

Supervisor: Dr.Fahad Al-shammery

Abstract:-

Scheduling is a fundamental concept in multitasking and operating system design in computer, and in real-time operating system design. CPU scheduler is the basis of multi-programming operating systems by switching the CPU between processes; an operating system can make a computer more productive, and scheduling algorithms are widely used in communication networks and operating systems to allocate resources to competing tasks. [4]
In this paper, visual interfaces for CPU scheduling algorithms are designed using Visual Stiedio code language and a webpage is presented for solving CPU algorithms.It may be used to educate users about these algorithms and how they work .

Keywords: scheduling, process, cpu.

Contents:

List of figures:

Chapter 1

1.Introduction:

1.1 Domain spacefic:

The Central Processing Unit (CPU) is an important component of a computer system; Hence it must be used efficiently. This can be achieved through something called CPU scheduling. CPU scheduling can be defined as the art of determining which processes are running on a CPU when there are several running processes. Also, the problem with which computer process is determining which queue is ready (in other words, which specific programs need some processing and are ready and waiting for that) that CPU will be allocated for processing. It is a fundamental problem with operating systems (OS) in terms of reducing user waiting when they simply want to perform a certain set of tasks. It is important because it has a significant impact on resource usage and overall system performance. [4]

1.2 Objective of the project :

To demstrate keep cpu always occupied .

1.3 Problem definition:

Scheduling: Refers to the way processes are set to run on available CPUs, since there are usually many more processes running than there are available CPUs. This task is performed by a program known as a scheduler and dispatcher. There are many types of scheduling tools such as short term scheduler, long term scheduler, and mid term scheduler. [3]

1.4 Project feature:

The goal of the scheduler is to achieve:

1. High processor usage
2. High productivity: that is, the number of operations performed per unit of time
3. Low response time: no time lapse from application submission to initiation the answer.[4]

Chapter 2

2. Related Works:

The CPU plays a major role in managing the processes that arrive in the form of multiple queues.All of this requires scheduling algorithms to operate in a real-time environment with special reference to task, control, and efficiency. Many researchers have introduced various CPU scheduling algorithms from time to time. Some of the research that is relevant to our work are:

Jain et al. (2015) presented a linear data model-based study of an improved Round Robin CPU scheduling algorithm with shortest-significant-first scheduling features with a variable amount of time. Banda et al. (2013) considered different temporal quantities of a set of operations and reduced context switches as well as improving the performance of the RR algorithm, computed the quantitative time using a measure of minimum dispersion and experimental analysis showed that the performance of the group-based quantitative RR algorithm (GBTQ) is better than the existing RR algorithm.

Suranauwarat (2007) used the emulator in the operating system to learn CPU scheduling algorithms in an easier and more efficient way.

Sindhu et al. (2010) proposed an algorithm that can handle all types of operations with optimal scheduling parameters.[3]

Chapter 3

3. CPU scheduling algorithms :

3.1 CPU Scheduling Concepts:

3.1.1- Bursts:

An executing process alternates between two cycles: CPU execution and I/O wait. The amount of time it takes for each CPU execution to be completed is called the CPU burst, and the time the process spends waiting for an I/O request to be fulfilled is the I/O burst. Historical data for the burst times of a process are taken into account when selecting a CPU scheduling algorithm. The point of this is to prevent a process with a very long burst time from taking control of the CPU before a process with a short burst time is able to run.[4]

3.1.2- Dispatcher:

The dispatcher is responsible for performing context switches. This means that it gives control of the CPU to the process that is to be run next. First, it saves the state of the currently running process. Next, the state of the selected process is loaded into the registers. Finally, the process begins executing at the location specified by the program counter.[4]

3.2 Scheduling algorithms categories:

The Scheduling algorithms can be divided into two categories with respect to how they deal with clock interrupts.

3.2.1-Preemptive Scheduling:

A scheduling discipline is preemptive if, once a process has been given the CPU can take away. The strategy of allowing processes that are logically run able to be temporarily suspended is called Preemptive Scheduling and it is contrast to the "run to completion" method.[1]

3.2.2-Non preemptive Scheduling:

A scheduling discipline is non-preemptive if, once a process has been given the CPU, the CPU cannot be taken away from that process. Following are some characteristics of non- preemptive scheduling:

1. In non-preemptive system, short jobs are made to wait by longer jobs but the overall treatment of all processes is fair.
2. In non-preemptive scheduling, a scheduler executes jobs in the following two situations.
   a. When a process switches from running state to the waiting state.
   b. When a process terminates.[1]

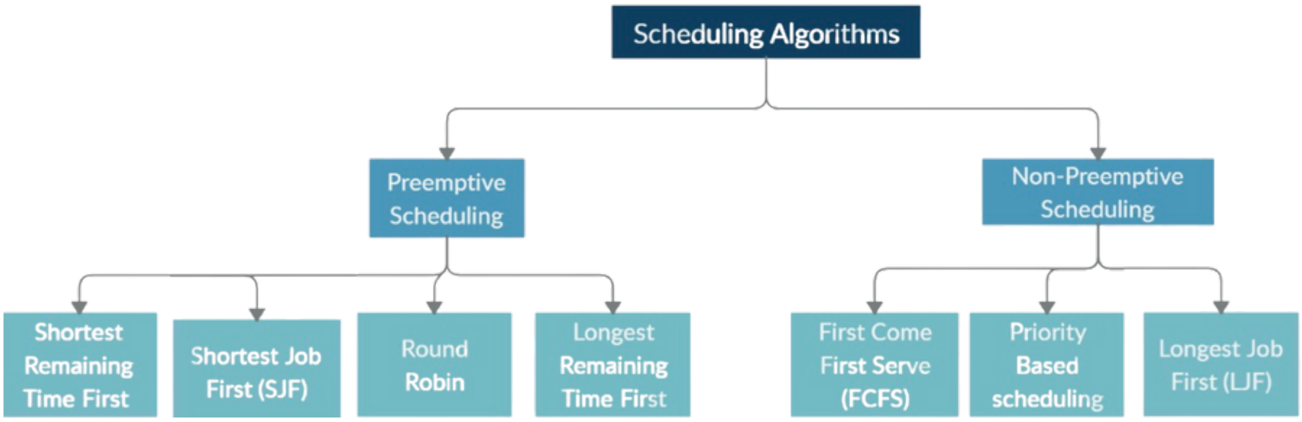

fig 1: The conceptual diagram shows which algorithms in the CPU are preventive or non-preemptive.

3.3 Scheduling Algorithms:

Sensible scheduling strategies might be:

• Maximize throughput or CPU utilization

• Minimize average turnaround time, waiting time or response time.

CPU Scheduling deals with the problem of deciding which of the processes in the ready queue is to be allocated the CPU; the following are some scheduling algorithms

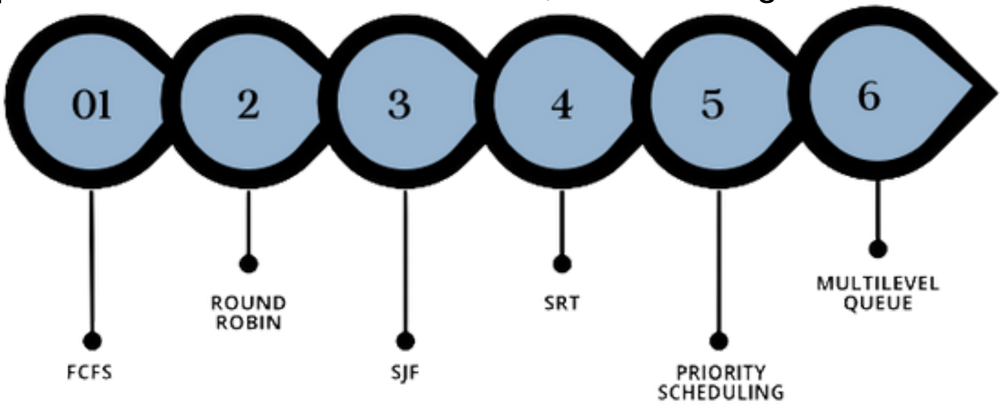

fig 2: CPU Scheduling algorithm .

3.3.1 First-come-first-served scheduling (FCFS):

Other names for this algorithm are:

- First in first out (FIFO)
- From commissioning to completion
- Run to completion

The first-in-first-out algorithm is probably the simplest scheduling algorithm. Processes are sent according to their arrival time in the ready queue. Once a process has a CPU, it runs to completion. FCFS scheduling is fair in the formal sense or human sense of fairness but unfair in the sense that long jobs make short jobs wait and unimportant jobs make important jobs wait. FCFS is more predictable than most other schemes because it saves time. FCFS schema is not useful for scheduling interactive users because it cannot guarantee good response time. FCFS scheduling code is easy to write and understand. First-Come-First-Served algorithm is rarely used as a master scheme in modern operating systems but is often built into other schemes.[1]

3.3.2 Schedule a Round robin:

Round robin (RR) is one of the oldest, simplest, fairest and most widely used algorithms. In round robin scheduling, processes are dispatched in a FIFO fashion but are given a limited amount of CPU time called a time slice or quantum. If a process does not complete before its CPU time expires, the CPU is preempted and given to the next process waiting in the queue. The pre-process is then placed at the back of the ready-made slate. Round Robin scheduling is proactive and therefore effective in time-sharing environments where the system needs to ensure reasonable response times to interactive users. The only interesting issue with the circular Robin scheme is the sleeve length. Setting quantum too short leads to too many context switches and reduces CPU efficiency. On the other hand, quantum tuning too long may result in poor response time and FCFS approximation. In any case, the average wait time in a round-robin scheduling window is often very long.[4]

3.3.3 Schedule Shortest Job First (SJF):

Another name for this algorithm is Shortest-Process-Next (SPN). Shortest job first is a scheduling algorithm in which the process with the smallest execution time is selected for execution next. Owing to its simple nature, shortest job first is considered optimal. It also reduces the average waiting time for other processes awaiting execution.[1]

3.3.4 Schedule Shortest Remaining Time (SRT):

In (SRT) scheduling algorithm, the process with the smallest amount of time remaining until completion is selected to execute. Since the currently executing process is the one with the shortest amount of time remaining by definition, and since that time should only reduce as execution progresses, processes will always run until they complete or a new process is added that requires a smaller amount of time.[1]

3.3.5 Priority scheduling:

Processes with equal priority are scheduled in FCFS order. The Shortest Job First (SJF)algorithm is a special case of the generic priority scheduling algorithm. The SJF algorithm is simply a priority algorithm where the priority is the opposite of the next (expected) CPU burst. That is, the longer the CPU burst, the lower the priority, and vice versa. Priority can be set either internally or externally. Internally defined priorities use some measurable quantity or attribute to calculate the priority of a process.[1]

3.3.6 Multilevel Queue Scheduling:

The multi-level queue scheduling algorithm divides the ready queue into several separate queues, in the operations of scheduling a multi-level queue permanently into a single queue. Processes are permanently assigned to each other, based on some process properties, such as:

- Visualize the CPU table
- Memory size
- Process priority
- Type of operation

Algorithms choose the process from the busy queue that has the highest priority, and run that process either:

- Proactive
- Not proactive

Each queue has its own scheduling algorithm.[4]

3.4 Objectives of a Scheduling Algorithm:

1. **Maximum utilization of CPU:** we can keep the CPU as busy as possible.
2. **Throughput:** means the number of processes which are completing their execution in per unit time. There must be maximum throughput.
3. **Turnaround time:** means that the time taken by the processes to finish their implementation. It must be a minimum.
4. **Waiting time:** is that time for which the process remains in the ready queue. It must be a minimum.
5. **Response time:** is the time when the process gives its first response. It must be a minimum.[1]

Chapter 4

4. Methodology :

In my project two methods were implemented to implement cpu scheduling algorithms to get the most efficiency:

1- cpu schedule visualization.

2- cpu scheduling algorithms.

4.1 cpu schedule visualization [5]:

Operating systems scheduling algorithms visualization. It is required to implement an OS scheduler using different scheduling algorithms. The work is divided into two modules:

- "Process Generator": generates the processes to bescheduled.
- "Scheduler": produces the schedules based on the chosen algorithm and demonstrates these schedules by visualgraphs.

scheduler module is responsible for generating a schedule for the current processes in the system to specify the CPU usage by these processes. implemented 4 scheduling algorithms:

1. Highest Priority First.(HPF)
2. First Come First Served. (FCFS)
3. Round Robin.(RR)
4. Shortest Remaining Time .(SRT)

4.1.1 Requirements:

- matplotlib
- tkinter
- numpy

Library used for GUI : Tkinter
Library used for plots : matplotlib
Startup window for selecting (Input file) and (Selecting an algorithm to draw its corresponding scheduling
The following output graphs is for inputfile at (testcases/SheetQuestion.txt)

4.1.2 How to use?

1. Start the program
2. Choose an input file
3. Select the algorithm using the scrolling box .
4. Click on Show/Update Graph .

4.2 cpu scheduling algorithms [6] :

In CPU scheduling algorithms, a page is designed for the four algorithms:
1- FCFS
2- SJF
3- Priority Scheduling
4- Round Robin
to apply how each algorithm works in the processor based on the number of processors and their priorities and to implement them according to the algorithm used.

4.2.1 Requirements:

- HTML
- CSS
- JavaScript

HTML: to build the initial website,
CSS: to design and format the web page,
JavaScript: to perform operations of CPU scheduler

4.2.2 How to use?

1- Choose the algorithm to be implemented,
2- Fill in the table with the numbers to be applied in the cpu
3- A column can be added or a column can be reduced according to the requirements
4- Pressing the go button to start executing the program

Chapter 5

5.Results :

5.1 cpu schedule visualization .

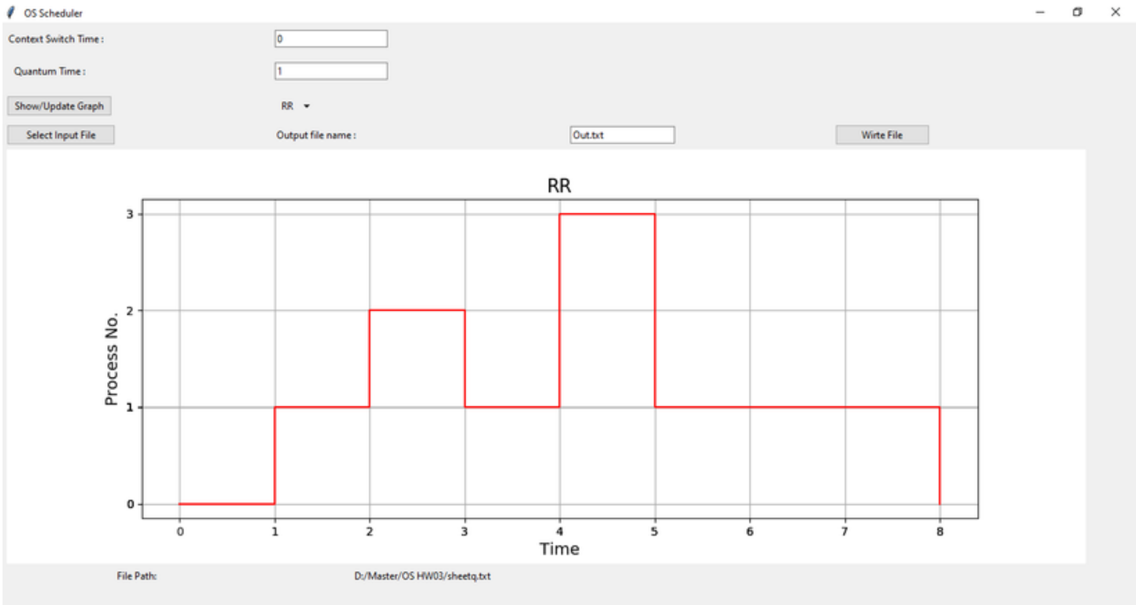

fig 3: The RR algorithm outputs in the program .

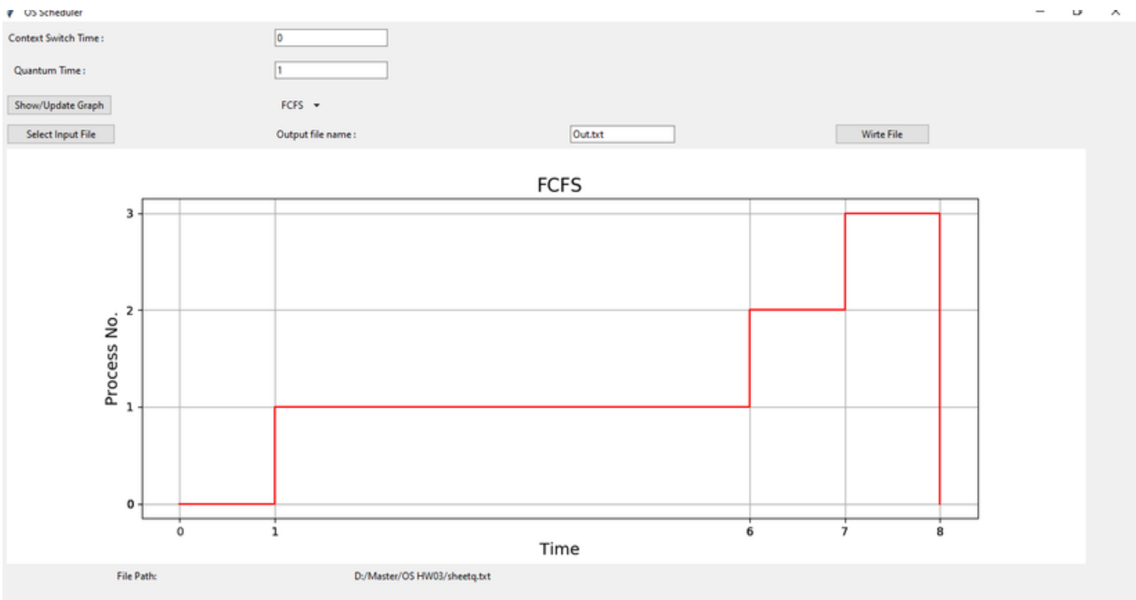

fig 4: The FCFS algorithm outputs in the program .

5.2 cpu scheduling algorithms.

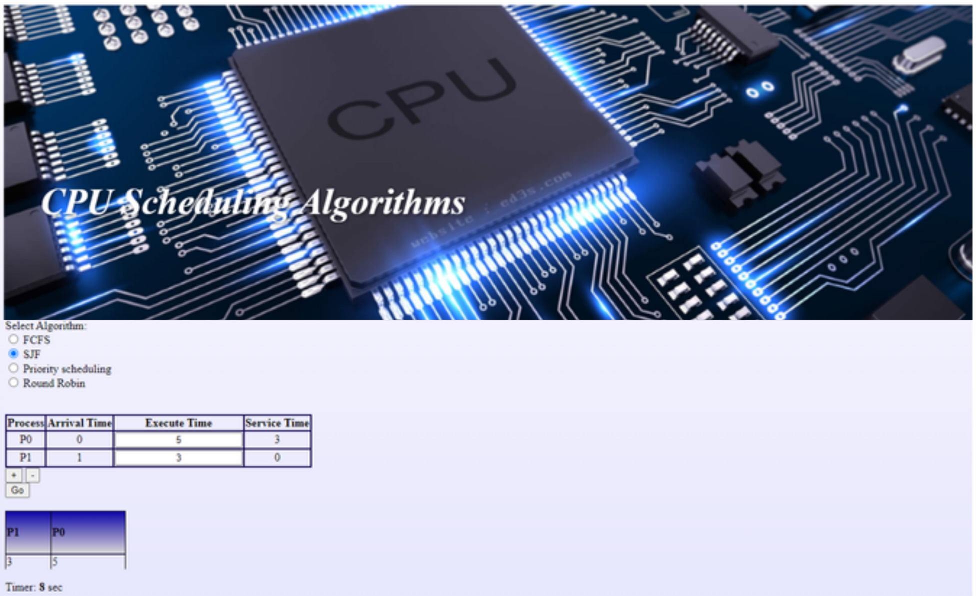

fig 5: The SJF algorithm outputs in the program .

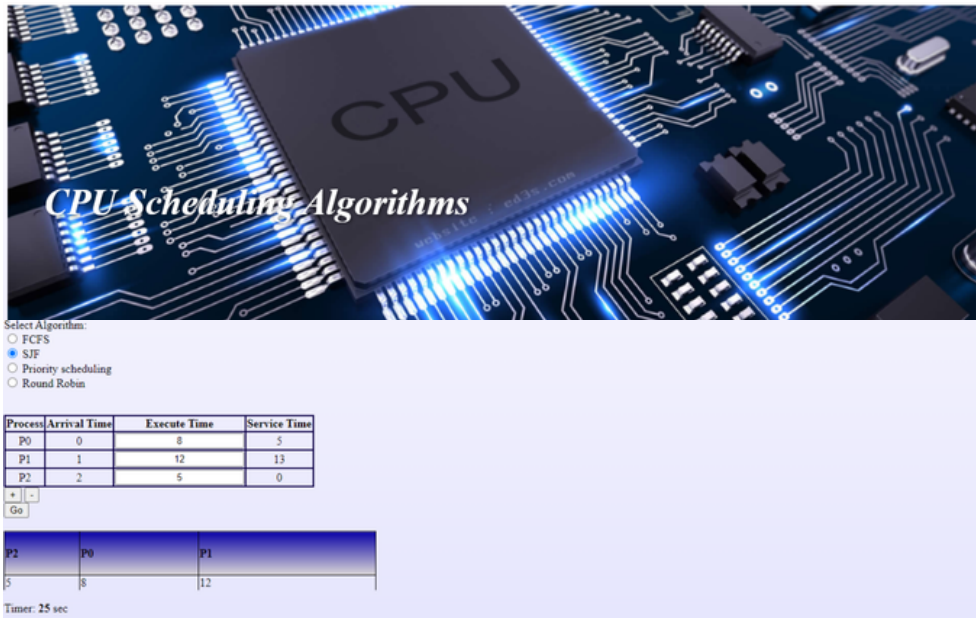

fig 6: The SJF algorithm outputs with add processes in the program .

Chapter 6

<u>6. Conclusions :</u>

At the end of my research on CPU scheduling I recommend several things:
1- Based on performance,the shortest job first(SJF)algorithm is recommended for the CPU scheduling problems of minimizing either the average waiting time or average turnaround time.
2- The first come first serve (FCFS) algorithm is recommended for the CPU scheduling problems of minimizing either the average CPU utilization or average throughput.
3- In RR algorithm the major problem is the selection of time quantum.Setting the quantum too short causes too many context switches and lower the CPU efficiency; setting the quantum too long may cause poor response time and approximates FCFS. In any event, the average waiting time under round robin scheduling is often quite long.
4- The SRT is the preemptive counterpart of SJF, and it has higher overhead than its counterpart SJF.
5- Multilevel Queue algorithm allow different algorithms to be used for various classes of processes.

Chapter 7

7. References :

[1] Abraham Silberschatz, A. and Galvin, P.B., (1994), “Operating System Concepts “, 4th ed .Addison-Wesley, MA.

[2] Steven Hand, (2010) "Operating Systems", Michaelmas Term, 12 lectures for CST

[3] Kevin Mellott ,(February 3, 2003)“CPU Scheduling,Withan Emphasison Algorithms".

[4] Taqwa Flayyih Hasan“CPU scheduling visualization“ (29 March 2014) .

[5] Operating systems scheduling algorithms visualization.github (6 May 2021).

[6] cpu schedule algorithms.codepen (13 April 2016).