ISSN 2278-3091
Volume 9 No.2, March -April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse75922020.pdf
https://doi.org/10.30534/ijatcse/2020/75922020

A Comparison of Scheduling parallel program tasks based on Java Applet

Haleema Essa Solayman
University of Mosul, Iraq, haleema_essa@uomosul.edu.iq

ABSTRACT

Process scheduling eminent to hold every treat that needs a resource in parallel processing, which is called processor or CPU. For different situations on a specific problem not every algorithm works best. Occasionally the algorithm(FCFS) is alter than other separate algorithms in small break abstraction when for multiple processes in each unit of time Round Robin is better. The Average of Turn Around Time and Average of Waiting Time is the touchstone measures for giving title to the planning rule to the algorithm used in scheduling. A lot of methods hold be practical for compound a CPU execution. The lens of this thesis is to hit study between algorithms, FCFS, Round Robin, SJF and Priority scheduling algorithms. The design of this compare to influence which formula is many fit for the processes that exist in the ready queue. The objective has shown what is the more suitable algorithm for the a specific processes. All Algorithms are programmed using JAVA APPLETS under ECLIPSE environment.

Key words: process, multiple process, scheduling, scheduling algorithm, burst time, waiting time, turnaround time, average waiting time, average turnaround time, priority.

1. INTRODUCTION

An operating system can be defined as a special program which manages software and instrumentality of computer resources. Operating system is the archetypal piece which is undischarged in the memory as the computer is supercharged on. If the operating system doesn't exist in the system, each human is responsible for creating a way to channelize accumulation to a machine, inform it how to indicate a file, and how to wad with new programs. In the first programmers required a way to carry out input operations/output operations. Withal, an indication needful for touch off the equipment between disparate programs became limpid Operating System. So, a needed period for dealing with the tools between various software begin to be clear and applying this guide to produce software such as DOS program. The DOS shortcut symbolize the(Disk Operating System). Which emphasizes the specific operating systems are primarily prefabricated for dealing with specified compounded input operations /output operations such as interacting with the variety of disk drives[1].

Older machines are not so powerful like today's computers. For the systems with young computer, programmers could not run more than one program at a time. For instance, programmer could not be browsing the internet and writing a paper all at the same time. However, today's operating systems are capable of handling enormous applications at the same instant of time. In order to make a computer able to deal with multiple applications at the same instance of time. In ordering for a computer to able fit to lot with many applications at the identical case, an efficient method must be exist for an efficient using of the CPU[1][2]. Lot of processes may be operate in identical instance of time, for this reason, some type of request must be adoptive to let apiece operation to get its quantum of CPU minute. The operating system should reserve the resources of computer between the potentially requirements that compete of multiple processes, the resources that are reserved to the process is processor execution time and the allocation of the system resources is called scheduling. The responsibility of scheduling should determine number of sensate, which consists of: efficient, fairness use of the time of processor, shortage of starvation of any specific process, and small overhead[1]. Furthermore, the scheduling may require to satisfy different grades of real time deadlines or priority for the beginning or finalization of particular process. In previous years, the focusing of the research, and various algorithms has been implemented. An affirmation of the planning investigate for using the systems with multiprocessors, especially in real-time planning and multithreaded applications. In a system with multiple programs, numerous jobs arrest concurrently inside the main memory. Every job pass between suspending for some occurrence of the events, like the Input/output operation completion and executing on a cpu. A processor remains busy for running single process whereas other processes waiting for execution, for this reason, scheduling is the essential key to the multiprogramming.

2. PROCESSORSCHEDULING

Processor scheduling is the rudimentary of operating systems with multiple programs. By swapping the processor through several processes, an operating system has the ability to raise the machine productivity. CPU scheduling permit single process for executing on the CPU whereas the other processes execution is waiting for execution, through the inaccessibility of a resource in the system, such as I/O, by this way, the complete using of the CPU is making. An objective of processor scheduling is for building the system in rapid, fair, and in an efficient manner. When a state of a processor be idle, an operating system required to choose one processes from the other processes that exist in the ready queue for execution. A chosen process be executed using short-term scheduler (what is called a CPU scheduler). A scheduler chooses the processes exist inside the memory and be ready for execution, assigning the processor to one from these processes. When, the CPU begin to be idle, an OS should choose one process from the ready-queue for execution. There is no requirement that the ready queue to be a first-in, first-out implementation (FIFO queue). An implementation can be a FIFO queue with specific priority assignment. Or like linked list implementation, or tree implementation. essentially, all processes that wait in the queue(ready queue) are wait to has a chance to execute on the processor. an information in a ready queues typically are(PCB)process control blocks for the system processes[1][2][3].

Scheduler can be defined as a special system program that treat process scheduling in different ways. the essential task of scheduler is specifying the jobs to be producing for the system and determining the process to be executed. The main Scheduler types are listed following:[3]

1-Long term scheduler – Because of a minimal capacity of the memory, all jobs at first are kept in secondary storage. Whenever they were stored in the main storage they were to be named as a process. The decision made is by long term(job)scheduler which determine the number of processes that should wait in the ready queue. so, this scheduler is responsible for determining the degree of system multi-programming.

2-Medium term scheduler – usually, an executing process requires Input /Output operations and doesn’t needs the processor computations. Through the running state of a process when the process require an I/O operation, an OS passes that process to the blocked queue from the running queue. As a process accomplishes its Input /Output operation; the process would reserve in the ready queue again. The complete decisions have to be making by the medium_term_scheduler. Which is a part of **swapping operation**.

3-Short Term Scheduler (or CPU Scheduler) – Whenever many processes exist in the primary memory basically, the whole processes are existent in the ready queue. Only one process is chosen to execute on the CPU. short term scheduler make this decision. clear sight is considered on the figure given below[3][4].

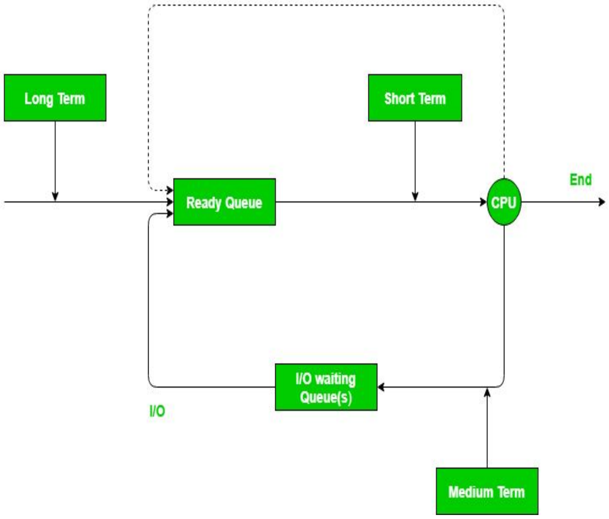


Figure 1: scheduler types

3. DISPATCHER

Another important part compromised in the function of the CPU scheduling is the **Dispatcher**. Which can be defined as a special program which his role in work comes next to the scheduler work. As a scheduler finished its task in determining the specific process, after this, it gets the process into the queue according to the required situation. A dispatcher is a part of system that grants the control of the processor to selected process by the CPU scheduler. That mission consists the listed steps[5]:

-Context Switch

-Switch to the mode of user

-Going to a desired program location of the user to start again from the location of the user program where it left in the final period of time. A dispatcher must be so quick, this operation done for each switching of the process. A required time used by the dispatcher for pausing one process, and begin with another is called **Dispatch Latency**. Which can be displayed by the following image[6]:

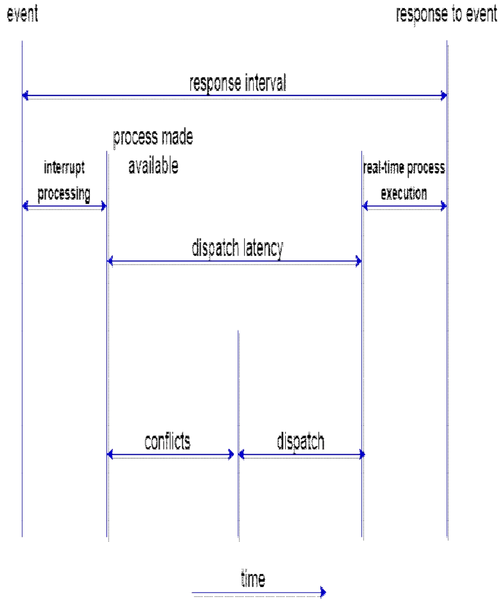


Figure 2: Dispatcher Latency

4. DIFFERENCES BETWEEN THE SCHEDULER AND DISPATCHER

Meditate a state, when there are several processes existing in the queue waiting to execute on the processor. But CPU unable to execute all the processes exist in the queue at that identical instant of time, operating system must determine a process depending on specific scheduling algorithm that used. The responsibility of choosing a process between enormous processes made by **cpu scheduler**, when the scheduler mission is finished. The **dispatcher** now has the responsibility. After a scheduler determine a specific process to execute, the dispatcher gets that process from the waiting state and a state of the process becomes the running state, in other word supplying the processor to the specific process be the responsibility of the dispatcher[7].

Considering the following model –assume that the number of processes in the ready queue are four processes from P1_to_ P4; and these processes are reached at t0_to_t3 respectively. The algorithm that used is First come First served scheduling algorithm. scheduler will decided P1 as the first process has come, so, it's the first process to executed, so dispatcher gets P1 to the running state[7][8].

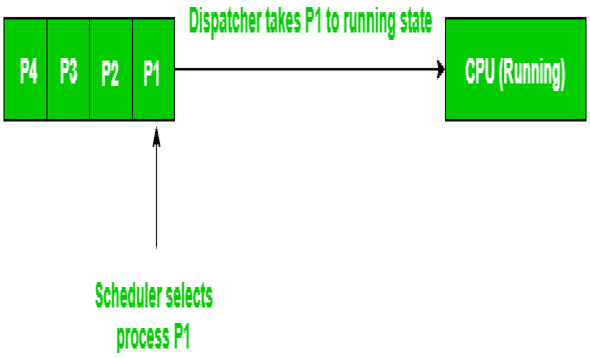

Dispatcher takes P1 to running state
P4
P3
P2
P1
CPU (Running)
Scheduler selects process P1

Figure 3: System processes

Table 1: Difference between the Scheduler and Dispatcher

ASSETS	DISPATCHER	SCHEDULER
Description:	Dispatcher is a special program which his role in work comes after the scheduler	Scheduler is a software which determines one process between many processes
Kinds:	The Dispatcher can be defined as a special piece of code, so it is not diversified	There are three kinds of scheduler: *LTS *STS *MTS
Dependence :	The dispatcher depends on scheduler, this denote that it must be waiting till the process is chosen by the scheduler.	The scheduler work separately, immediately when its work is required
Algorithm:	There is no deterministic method for the dispatcher working	There are different algorithm for scheduler working like Round Robin, priority, etc.
Time required:	Required period by the dispatcher is called dispatch latency.	Required period by scheduler, commonly small, Hence it disregard.
Functions:	Dispatcher is accountable for: Switching Context, Switch to user mode, switch to appropriate location when process work when returned	Scheduler is responsible of selection of processes.

5-THE SCHEDULING CRITERIA OF THE SCHEDULING ALGORITHMS

There are several criteria to be satisfied when determining the "best" scheduling algorithm for a specific situation and environment ,which including[9]:

The utilization of the processor

CPU utilization means the CPU remains busy as conceivable. The utilization may be between 0% till 100%. it must be ranged from 40% for a system load is lightly to 90% for a system with heavily used, In a real system.

Throughput

One of the most significant measurements of work is the number of processes executed at a unit of time, this known as **throughput**. In the system with high number of processes, the throughput should become 1 process for each hr.; in little cooperation, throughput should become ten processes in the sec.

Turnaround Time

One of most significant criterion for the process is what is the amount of time taken for execution. Total amount of time since presenting the process till the execution accomplished, called the turnaround time. It can be defined as the sum of the intervals time consumed by the process for waiting to the time when loaded into main storage, holding inside the waiting queue, processing using the processor, applying Input operations/Output operations. For reacting environment, this criterion may not become preferable.

- T.A.T is the Time differs from the time at which process reach the system until termination of that process.

Waiting Time

The amount of time taken by the process in the ready queue with the waiting state.

•Waiting time is the different between the turnaround time and Burst time

Response Time

usually, some outputs can be constructed earlier by the process, this process still computes newly output results as the earlier computation outputs have been presented. So, one of the most important criterion is an instant between presenting the demand till the primary producing response.

Such criterion is known as response time, which can be defined as the total time taken by the process to begin responding, but it is not the time taken to produce that required response[9][10].

6- IMPORTANT TIME TERMS OF SCHEDULING OPERATIONS

Reached Time or Arrival Time

An instant at which the particular process enters the queue, and enter the waiting state.

Termination Time or Completion Time

An instant at which process execution is accomplished.

Execution Time or Burst Time

The required period for executing the process on CPU.

CPU Scheduling Time or CPU Time

The required period for choosing the process to execute on CPU[10][11].

7. CONDITIONS FORPROCESSOR SCHEDULING

The decision taken by the processor should satisfies the listed below four conditions:

1. The transformation of the state of the process from running state to the waiting state, when this process waiting When a process transforms for I/O operations, or for completion of other executed process.
2. The transformation of the state of the process from running state to the waiting state, when an interrupt is occurred.
3. The transformation of the state of the process from running state to the waiting state, when I/O termination.
4. When a process **completes its execution[12]**.

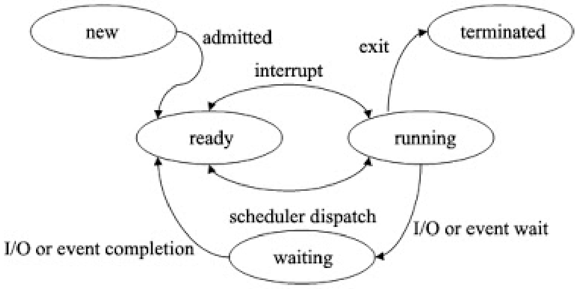


Figure 4: Process state

In conditions one and four, the process cannot determines the used scheduling(there is no option for the process), because, when the process is newly enter the system and enter the ready queue, this process must be determined to execute on the processor. Unlike states two and three, in these states an option is exist.

For the states one and four, the scheduling scheme can be called **non-preemptive**; if not, the scheduling scheme can be called **preemptive[1][13].**

7.1 Non-Preemptive Scheduling

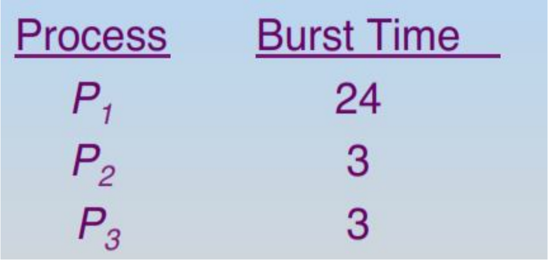
In the scheduling of type non-preemptive, when a CPU has been reserved for a particular process, the process yet uses the CPU until it freeing the CPU either by terminating the process or switching to the wait state.
This scheduling procedure is chosen by the Microsoft.
This type of scheduling does not need the a specific hardware like timer that is must be exist in the preemptive scheduling type[14][11].

7.2 Preemptive Scheduling

In scheduling of type preemptive, usually there are priorities allocated for each process. It is essentially to executes the task with a higher priority before that task that has the lower priority, even if the lower priority task is in the running state. So, it is interrupted for a specific amount of time, then it can continue later as the higher priority task terminate the execution on the CPU[14][11].

8. SCHEDULING ALGORITHMS

8.1 First-Come First-Served Scheduling Algorithm(FCFS)



Process	Burst Time
P_1	24
P_2	3
P_3	3

Figure 5: System processes with burst time

Suppose that the processes arrive in the order:
The Gantt Chart for scheduling these processes is:

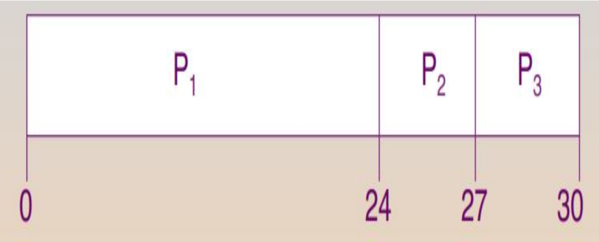


Figure 6: Gantt chart

If waiting time is symbolled as W.T and average waiting time as A.W.T

W.T of P1 = 0,W.T of P2 = 24; W.T of P3 = 27
A.W.T = (0 + 24 + 27)/3 = 17

If the process P2 is reach to the system ,then P3,finally P1 is reach.

■ The following figure show the scheduling Gantt chart of these processes:

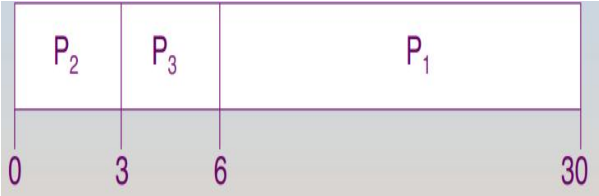


Figure 7: Gantt chart

W.T of P1 = 6;W.T of P2 = 0; W.T ofP3 = 3.
A.W.T = (6 + 0 + 3)/3 = 3.
This case is better than preceding case.

8.2- SJF Scheduling Algorithm

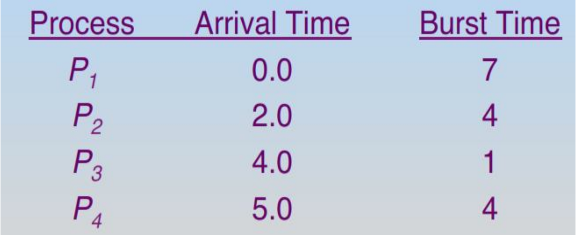
Joined with every system process the length of next processor burst associated with it. By using these lengths to schedule the process using the shortest time, this happened using one of the following two schemes[15]:

✦ no preemption – once the CPU allocated to the process should not be preempted till it terminates its CPU burst time.

✦ preemption – when a new process reaches with CPU burst time less than residual time for executing process, it preempt the present executing process. This method is defined as the Shortest-Remaining-Time-First (SRTF).

SJF is favorable– because it grants less average waiting time for a particular set of processes.

Suppose that the processes that reached to the system are P1 , P2 , P3 , P4.



Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

Figure 8: System processes with burst time

The Gantt chart used for scheduling these processes using SJF (non-preemptive) is:

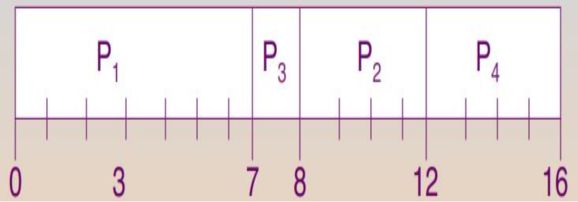
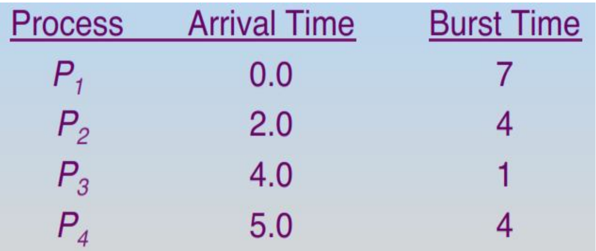


Figure 9: Gantt chart

A.W.T = (0 + 6 + 3 + 7)/4 = 4
If the used the scheme is Preemptive SJF



Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

Figure 10: System processes with burst time

The Gantt chart used for scheduling these processes using SJF (preemptive) is:

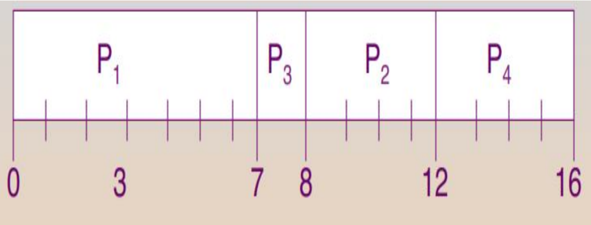


Figure 11: Gantt chart

A.W.T = (9 + 1 + 0 +2)/4 = 3.
Choosing the Next processor Burst Length
■ length evaluation is can be only done.
■ by using exponential averaging, using preceding CPU bursts length the length evaluation can be done.

1. tm =real nth processor burst length.
2. tm+1 = evaluated next processor burst value
3. q , 0≤ q ≤1
4. τ m+1 = qtm + (1− q) τm.

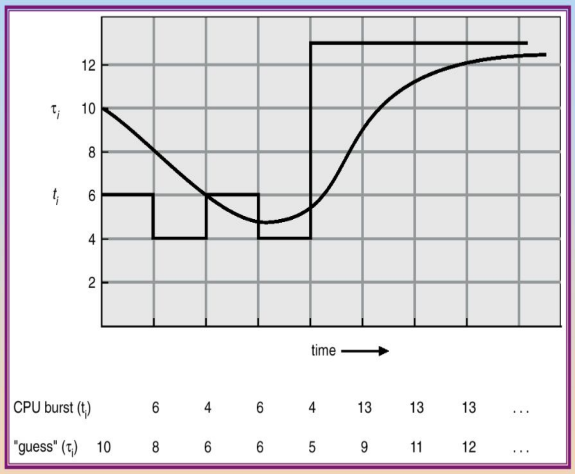


Figure12: Evaluation Next CPU BurstLength

The following model give an examples for Exponential Averaging

■ q =0

✦ $\tau m+1 = \tau*m$

✦ Current value does not evaluate.

■ q =1

✦ $\tau m+1 = tm$

✦the presenting value of the last CPU burst is count only.

■ the expansion of this the formula is:

$\tau m+1 = q* tm+(1 - q)*q\ tm -1$, which also continue

$+(1 - q)*j*q* tm -1$,which also continue

$+(1 - a)*m=1 *tm*\tau 0$

■ the value of(q) and (1 - q) must be equal to 1or less than 1, each predecessor term has greater weight than its successive.

8.3Scheduling Algorithm with Priority Assignment

Each process using this type of scheduling is assigned with an integer number, this number known as apriority of this process. The process higher priority is the one that will allocates the CPU, usually the number with smallest value means highest priority, priority scheduling algorithm may be:

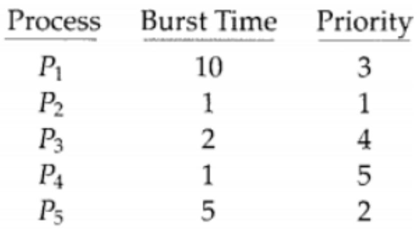
✦ Preemptive scheduling algorithm

✦ Non preemptive scheduling algorithm

SJF is categorized under priority scheduling algorithm, in this case the priority is the next CPU burst time that expected.

This conditions cannot be executed by the process with low priority and can leads to Starvation, the solution is to upgrade the priority of the process as time is progresses[1][15][16].

supposed to have the following set of processes(P1, l2, ..., P5),and all of these processes reached at time 0, in the order. and the CPU burst given in milliseconds[1]:



Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

Figure 13: System processes with burst time and priority

By using priority scheduling, the scheduling of these processes be visible in the following Gantt chart:

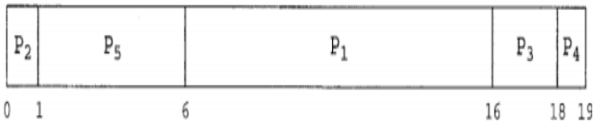


Figure 14: scheduling Gantt chart

The A.W.T = 8.2 Ms.

The specification of the Priorities can be done either externally or internally. some measurements are used to consider Internally priorities for the process. like, the time limits, how many files are opened, the requirements of memory, and arate between the average burst of Input/ Output to average processor burst can be applied when priorities is being computed. Standards measurements which lie externally of the operating system used to assign external priorities to the processes, such the importance of the process, the department guarantee the work, the amount and the type of property being paid for the use of the computer, and other, usually constitutional, worker.

With Priority scheduling algorithm also there are the two for scheduling preemptive and non-preemptive scheduling[7][9]. The priority of the process entered the ready queue is compared with the priority of the process that is running in the present time. if the result of comparison show that the reached process has the higher priority, then this scheme of scheduling(preemptive) will preempt the CPU. In a non-preemptive priority scheduling algorithm will plainly place the new process at the head of the ready queue.

Starvation or indefinite blocking A significant complications when using priority scheduling algorithms. A process that is exist in the queue is ready for execution, but waiting long time for the CPU, and could be detected as a blocked process, this type of scheduling algorithm can produce some processes with low- priority, that wait for unlimited amount of time. A process with low-priority In a system with heavy load can be refused from allocating the CPU, by the processes with higher priority. often, one of the following two situations can happened, the first, is the low priority process will be late in running, or collapsing the low priority processes(which are not completed yet), and the system finally will break [17][5].

To solve the previous situation of unlimited block of processes by using an aging technique. Which can be defined as a method for progressively rising the priority of waiting processes that is waits for a long time.

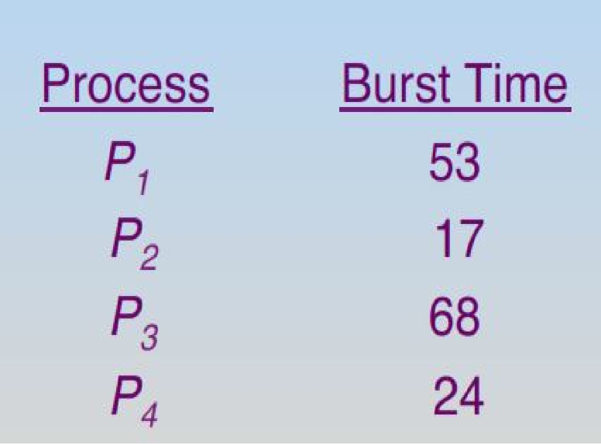
8.4 Round Robin Scheduling Algorithm (RR)

In the Round Robin scheduling Algorithm each process take a small amount of CPU time (called), usually 10-100 milliseconds. When the time quantum has finished, the presently executing process is preempted and combined to the ready queue end.

If the number of waiting processes that exist in the ready queue is m, and t is the time quantum consumed by every process for executing on the CPU, so a single process possess 1/m of CPU time of slice q time units at instant. There is no process waits for more than (n-1)q time [2][10].

With regard of this algorithm performance, if q is great amount of time then the algorithm function like FIFO, else when q is little(q must be large with regard to context switch), overhead is too high.

If the processes which arrived to the system are P1 , P2 , P3 , P4.



Process	Burst Time
P_1	53
P_2	17
P_3	68
P_4	24

Figure 15: Processes with burst time

The Gantt chart for scheduling those processes using RR with Time Quantum = 20 shown below:

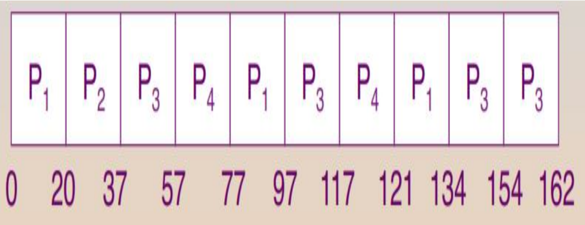


Figure 16: Gantt chart

Usually, higher average turnaround time as compared with SJF, but larger response.

The following figure show the Time Quantum and the Context Switch Time

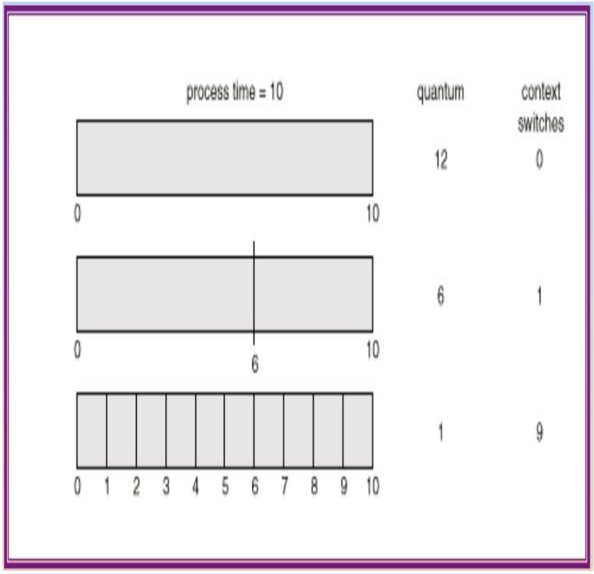


Figure17: Quantum and Context Switch Time

8.5 Multilevel Queue Scheduling:

In this type of scheduling algorithm, processes are organized into different categories. A common example may be foreground (or what so called Interactive processes) or background (or what so called batch processes). Ready queue is divided into separate queues: the first queue is called foreground queue or interactive queue, and the second is called background queue or batch queue, each queue possess its private scheduling algorithm, foreground process use RR scheduling algorithm, and background process use First Come First Serve algorithm, Scheduling must be done between the queues. constant priority scheduling; this denote works on all from foreground processes then from background processes. Let investigate an example of a multilevel queue scheduling with five queues, listed below in the order of the priority[2][16].

A. The processes of the system
B. Interactive processes
C. Interactive rewriting processes
D. Batch processes
E. Student processes Each of which has indefinite priority over lower priority queues.

There are no processes in the batch queue, for example can execute except if the queue used for processes in the system processes, interactive processes, all are vacuous. If a process of an interactive type arrive in the ready queue and a process of a batch type is running, then the batch process must be preempted. Another solution that could be taken is splitting the process or time among the queues, this mean using time slice among queues. As an example, given a 80% of the processor time for foreground queue, based on Round Robin algorithm between these processes, where given a 20% of the processor time for background [1][5][13].

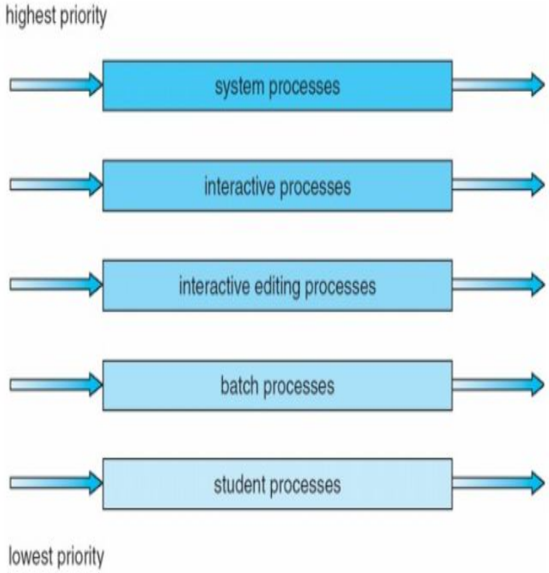


Figure 18: Multilevel Queue Scheduling

8-9- Multilevel Feedback Queue Scheduling

In this type of scheduling algorithm, there are a number of queues and the process may progress between them; this method can be completed, by using aging technique. The parameters that can be used to describe the scheduling of the Multilevel-feedback-queue are:

A. The type of algorithm used for scheduling the set of processes that exist in each queue.
B. The number of available queues
C. The used method for describing the desired time for process improvement.
D. The used method for describing the desired time for process demotion.

E. The used method for describing the desired queue for the process, and the time at which the process requires a specific service.

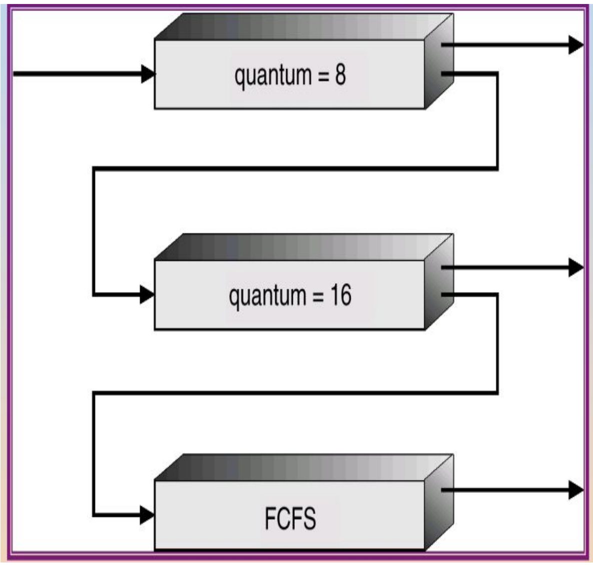
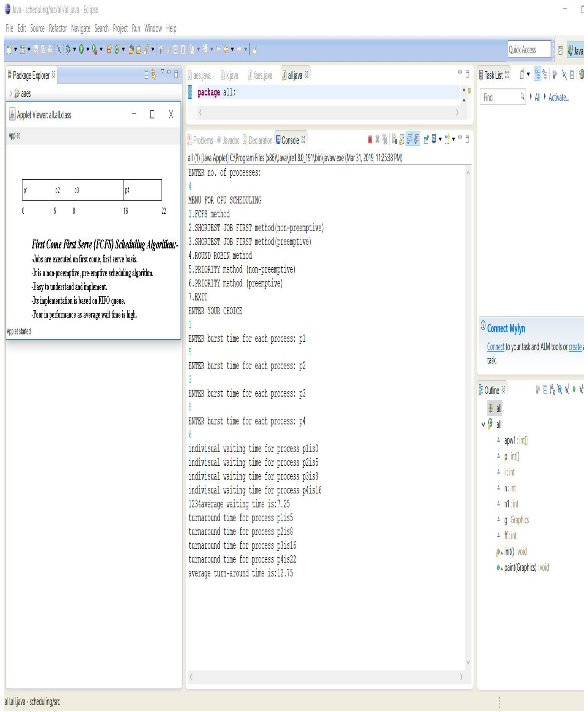


Figure 19: Scheduling of Multilevel Feedback Queue

If there are Three queues in the system: Q0 with time 8 MS, Q1 with time 16 MS, and Q2 work with First Come First Serve method. The first job arrives in Q0 that depends on FCFS algorithm. When the first process gets the processor, it has 8 Ms. If this time is not enough, it is presented to Q1. At Q1 job is scheduled using FCFS and gains 16 Ms. If the process yet does not finish all it works, it progress to next queue Q2, after termination in previous queue Q1[5][12[8].

9- PRACTICALPARTUSING JAVA APPLET


Java - scheduling/src/all/all.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
aaes
aes.java
k.java
faes.java
all.java
Task List
Find
Applet Viewer: all.all.class
Applet
Problems
Javadoc
Declaration
Console
Outline
all.all.java - scheduling/src

```
package all;
```

p1	p2	p3	p4

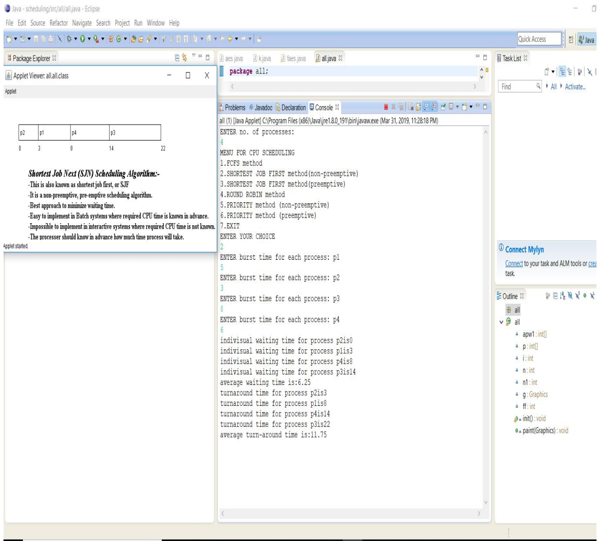
0 5 8 16 22

First Come First Serve (FCFS) Scheduling Algorithm:-
-Jobs are executed on first come, first serve basis.
-It is a non-preemptive, pre-emptive scheduling algorithm.
-Easy to understand and implement.
-Its implementation is based on FIFO queue.
-Poor in performance as average wait time is high.
Applet started.

all (1) [Java Applet] C:\Program Files (x86)\Java\jre1.8.0_191\bin\javaw.exe (Mar 31, 2019, 11:25:38 PM)

```
ENTER no. of processes:
4
MENU FOR CPU SCHEDULING
1.FCFS method
2.SHORTEST JOB FIRST method(non-preemptive)
3.SHORTEST JOB FIRST method(preemptive)
4.ROUND ROBIN method
5.PRIORITY method (non-preemptive)
6.PRIORITY method (preemptive)
7.EXIT
ENTER YOUR CHOICE
1
ENTER burst time for each process: p1
5
ENTER burst time for each process: p2
3
ENTER burst time for each process: p3
8
ENTER burst time for each process: p4
6
indivisual waiting time for process p1is0
indivisual waiting time for process p2is5
indivisual waiting time for process p3is8
indivisual waiting time for process p4is16
1234average waiting time is:7.25
turnaround time for process p1is5
turnaround time for process p2is8
turnaround time for process p3is16
turnaround time for process p4is22
average turn-around time is:12.75
```

Figure 20: Applying FCFS Scheduling Algorithm


Java - scheduling/src/all/all.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
aes.java
k.java
faes.java
all.java
Task List
Find
Applet Viewer: all.all.class
Applet
Problems
Javadoc
Declaration
Console
Connect Mylyn
Outline

```
package all;
```

p2	p1	p4	p3

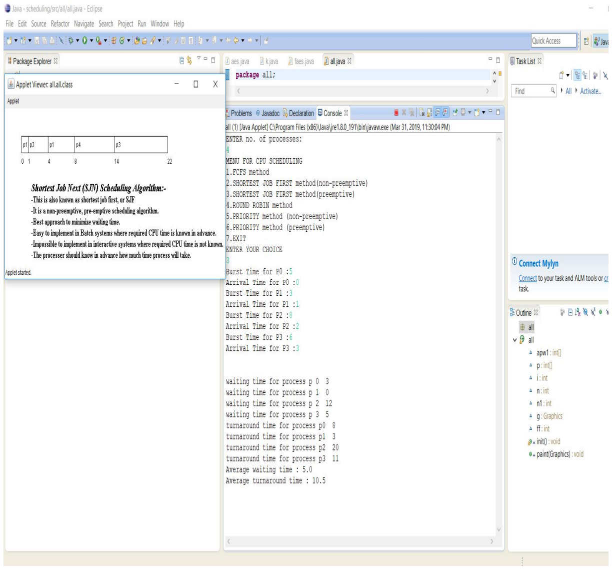
0 3 8 14 22

Shortest Job Next (SJN) Scheduling Algorithm:-
-This is also known as shortest job first, or SJF
-It is a non-preemptive, pre-emptive scheduling algorithm.
-Best approach to minimize waiting time.
-Easy to implement in Batch systems where required CPU time is known in advance.
-Impossible to implement in interactive systems where required CPU time is not known.
-The processer should know in advance how much time process will take.
Applet started.

all (1) [Java Applet] C:\Program Files (x86)\Java\jre1.8.0_191\bin\javaw.exe (Mar 31, 2019, 11:28:18 PM)

```
ENTER no. of processes:
4
MENU FOR CPU SCHEDULING
1.FCFS method
2.SHORTEST JOB FIRST method(non-preemptive)
3.SHORTEST JOB FIRST method(preemptive)
4.ROUND ROBIN method
5.PRIORITY method (non-preemptive)
6.PRIORITY method (preemptive)
7.EXIT
ENTER YOUR CHOICE
2
ENTER burst time for each process: p1
5
ENTER burst time for each process: p2
3
ENTER burst time for each process: p3
8
ENTER burst time for each process: p4
6
indivisual waiting time for process p2is0
indivisual waiting time for process p1is3
indivisual waiting time for process p4is8
indivisual waiting time for process p3is14
average waiting time is:6.25
turnaround time for process p2is3
turnaround time for process p1is8
turnaround time for process p4is14
turnaround time for process p3is22
average turn-around time is:11.75
```

Figure 21: Applying SJN Scheduling Algorithm no preemption


Java - scheduling/src/all/all.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
aes.java
k.java
faes.java
all.java
Task List
Find
Applet Viewer: all.all.class
Applet
Problems
Javadoc
Declaration
Console
Connect Mylyn
Outline

```
package all;
```

p1	p2	p1	p4	p3

0 1 4 8 14 22

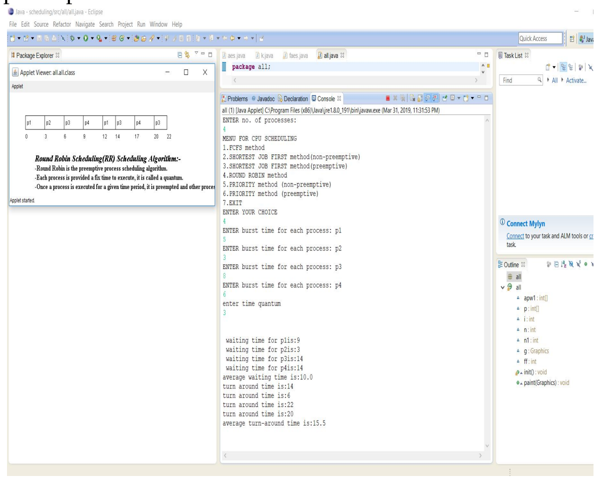
Shortest Job Next (SJN) Scheduling Algorithm:-
-This is also known as shortest job first, or SJF
-It is a non-preemptive, pre-emptive scheduling algorithm.
-Best approach to minimize waiting time.
-Easy to implement in Batch systems where required CPU time is known in advance.
-Impossible to implement in interactive systems where required CPU time is not known.
-The processer should know in advance how much time process will take.
Applet started.

all (1) [Java Applet] C:\Program Files (x86)\Java\jre1.8.0_191\bin\javaw.exe (Mar 31, 2019, 11:30:04 PM)

```
ENTER no. of processes:
4
MENU FOR CPU SCHEDULING
1.FCFS method
2.SHORTEST JOB FIRST method(non-preemptive)
3.SHORTEST JOB FIRST method(preemptive)
4.ROUND ROBIN method
5.PRIORITY method (non-preemptive)
6.PRIORITY method (preemptive)
7.EXIT
ENTER YOUR CHOICE
3
Burst Time for P0 :5
Arrival Time for P0 :0
Burst Time for P1 :3
Arrival Time for P1 :1
Burst Time for P2 :8
Arrival Time for P2 :2
Burst Time for P3 :6
Arrival Time for P3 :3


waiting time for process p 0  3
waiting time for process p 1  0
waiting time for process p 2  12
waiting time for process p 3  5
turnaround time for process p0  8
turnaround time for process p1  3
turnaround time for process p2  20
turnaround time for process p3  11
Average waiting time : 5.0
Average turnaround time : 10.5
```

Figure 22: Applying FCFS Scheduling Algorithm preemption


Java - scheduling/src/all/all.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
aes.java
k.java
faes.java
all.java
Task List
Find
Applet Viewer: all.all.class
Applet
Problems
Javadoc
Declaration
Console
Connect Mylyn
Outline

```
package all;
```

p1	p2	p3	p4	p1	p3	p4	p3

0 3 6 9 12 14 17 20 22

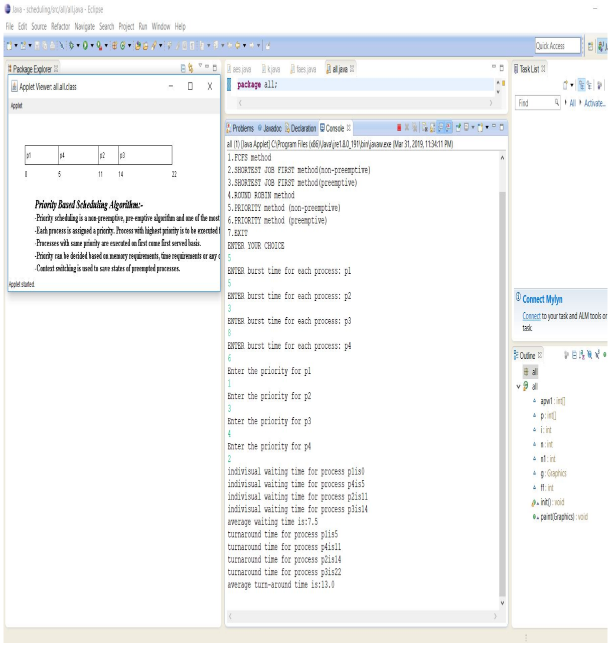
Round Robin Scheduling(RR) Scheduling Algorithm:-
-Round Robin is the preemptive process scheduling algorithm.
-Each process is provided a fix time to execute, it is called a quantum.
-Once a process is executed for a given time period, it is preempted and other proces
Applet started.

all (1) [Java Applet] C:\Program Files (x86)\Java\jre1.8.0_191\bin\javaw.exe (Mar 31, 2019, 11:31:53 PM)

```
ENTER no. of processes:
4
MENU FOR CPU SCHEDULING
1.FCFS method
2.SHORTEST JOB FIRST method(non-preemptive)
3.SHORTEST JOB FIRST method(preemptive)
4.ROUND ROBIN method
5.PRIORITY method (non-preemptive)
6.PRIORITY method (preemptive)
7.EXIT
ENTER YOUR CHOICE
4
ENTER burst time for each process: p1
5
ENTER burst time for each process: p2
3
ENTER burst time for each process: p3
8
ENTER burst time for each process: p4
6
enter time quantum
3


 waiting time for p1is:9
 waiting time for p2is:3
 waiting time for p3is:14
 waiting time for p4is:14
average waiting time is:10.0
turn around time is:14
turn around time is:6
turn around time is:22
turn around time is:20
average turn-around time is:15.5
```

Figure 23: Applying RR Scheduling Algorithm



Priority Based Scheduling Algorithm:-

-Priority scheduling is a non-preemptive, pre-emptive algorithm and one of the most

-Each process is assigned a priority. Process with highest priority is to be executed f

-Processes with same priority are executed on first come first served basis.

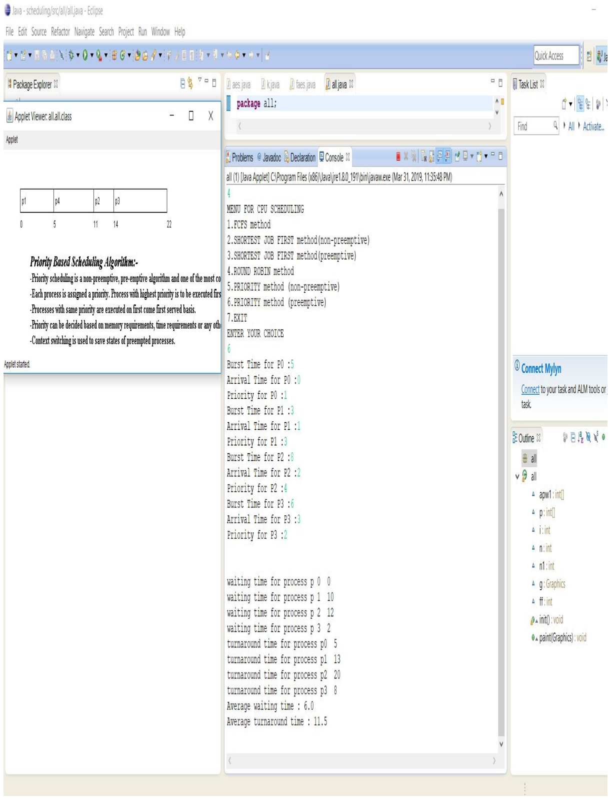
-Priority can be decided based on memory requirements, time requirements or any c

-Context switching is used to save states of preempted processes.

```
package all;
```

```
all (1) [Java Applet] C:\Program Files (x86)\Java\jre1.8.0_191\bin\javaw.exe (Mar 31, 2019, 11:34:11 PM)
1.FCFS method
2.SHORTEST JOB FIRST method(non-preemptive)
3.SHORTEST JOB FIRST method(preemptive)
4.ROUND ROBIN method
5.PRIORITY method (non-preemptive)
6.PRIORITY method (preemptive)
7.EXIT
ENTER YOUR CHOICE
5
ENTER burst time for each process: p1
5
ENTER burst time for each process: p2
3
ENTER burst time for each process: p3
8
ENTER burst time for each process: p4
6
Enter the priority for p1
1
Enter the priority for p2
3
Enter the priority for p3
4
Enter the priority for p4
2
indivisual waiting time for process p1is0
indivisual waiting time for process p4is5
indivisual waiting time for process p2is11
indivisual waiting time for process p3is14
average waiting time is:7.5
turnaround time for process p1is5
turnaround time for process p4is11
turnaround time for process p2is14
turnaround time for process p3is22
average turn-around time is:13.0
```

Figure 24: Applying Priority Scheduling Algorithm no preemption



Priority Based Scheduling Algorithm:-

-Priority scheduling is a non-preemptive, pre-emptive algorithm and one of the most co

-Each process is assigned a priority. Process with highest priority is to be executed firs

-Processes with same priority are executed on first come first served basis.

-Priority can be decided based on memory requirements, time requirements or any oth

-Context switching is used to save states of preempted processes.

```
package all;
```

```
all (1) [Java Applet] C:\Program Files (x86)\Java\jre1.8.0_191\bin\javaw.exe (Mar 31, 2019, 11:35:48 PM)
4
MENU FOR CPU SCHEDULING
1.FCFS method
2.SHORTEST JOB FIRST method(non-preemptive)
3.SHORTEST JOB FIRST method(preemptive)
4.ROUND ROBIN method
5.PRIORITY method (non-preemptive)
6.PRIORITY method (preemptive)
7.EXIT
ENTER YOUR CHOICE
6
Burst Time for P0 :5
Arrival Time for P0 :0
Priority for P0 :1
Burst Time for P1 :3
Arrival Time for P1 :1
Priority for P1 :3
Burst Time for P2 :8
Arrival Time for P2 :2
Priority for P2 :4
Burst Time for P3 :6
Arrival Time for P3 :3
Priority for P3 :2


waiting time for process p 0  0
waiting time for process p 1  10
waiting time for process p 2  12
waiting time for process p 3  2
turnaround time for process p0  5
turnaround time for process p1  13
turnaround time for process p2  20
turnaround time for process p3  8
Average waiting time : 6.0
Average turnaround time : 11.5
```

Figure 25: Applying FCFS Scheduling Algorithm Preemption

10. CONCLUSION

In this paper, various values of average waiting time and average turnaround time were displayed, based on different scheduling algorithms used, such as First Come First Serve algorithm(FCFS), Shortest Job First algorithm(SJF) ,Round Robin algorithm(RR), and Priority Scheduling algorithm, all of these algorithms are applied in the practical part of this work. After gathering the needed information, the following results were reached: the FCFS algorithm is the better when the burst time is small, the SJF algorithm is preferred when the processes running in concurrent manner. The RR algorithm should takes the value, so the result become as proximity as the result produced by the FCFS or SJF algorithms.

For the foreground and background processes, the best algorithms to used are Multi level Queue scheduling algorithm and Multi level feedback Queue scheduling algorithm, which are used to schedule the foreground and background processes.

Finally form the gathered results, one can be conclude that each algorithm is good for specific situation, based on the processor that exist in the system, which play the important role, and the speed of it rely on the load or burden falls on it.

REFERENCES

1-MARWAN A.OMAR and Mustafa J. Zwaid, CPU SCHEDULING, PhD thesis, university of Baghdad, may2012

2- Avid. Silberschatz, Peter B.Galvin, Greg Gagne, Operating System Concepts, Tenth Edition, John Wiley & Sons, Inc. ISBN 978-1-118-06333-0, April2018

3-Andysah PuteraUtamaSiahaan,Comparison Analysis of CPU Scheduling : FCFS, SJF and Round Robin, Article 2016 IJEDR | Volume 4, Issue 3 | ISSN: 2321-9939,July 2016

4_https://www.tutorialspoint.com/operating_system/os_process_scheduling_algorithms

5- Song B. et al. (2018) A Multidimensional Workload Assessment Method for Power Grid Dispatcher. In: Harris D. (eds) Engineering Psychology and Cognitive Ergonomics. EPCE 2018. Lecture Notes in Computer Science, vol 10906. Springer, Cham https://doi.org/10.1007/978-3-319-91122-9_5

6- https://www.studytonight.com/operating-system/cpu-scheduling(handout)

7_https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/5_CPU_Scheduling

8-https://www.geeksforgeeks.org/operating-system-difference-dispatcher-scheduler

9-Dr.Amit Agarwal, Saloni Jain, Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment, International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 7– Mar 2014 https://doi.org/10.14445/22312803/IJCTT-V9P163

10-Sonia., Rajesh Khanna, Neeraj Kumar, Load Balancing Approach for Proposed Hybrid Scheduling Algorithm in Heterogeneous Networks ,Load Balancing Approach for Proposed Hybrid Scheduling Algorithm in Heterogeneous Networks, Vol 8, No 1 (2018)

11-Chowdhury, N., M. Aslam Uddin, K., Afrin, S., Adhikary, A., & Rabbi, F. (2018). Performance Evaluation of Various Scheduling Algorithm Based on Cloud Computing System. Asian Journal of Research in Computer Science, 2(1), 1-6.
https://doi.org/10.9734/ajrcos/2018/v2i124768

12-Sujith Kumar Reddy N., Santhi H., Gayathri P., Jaisankar N. (2018) A New CPU Scheduling Algorithm Using Round-robin and Mean of the Processes. In: Muttoo S. (eds) System and Architecture. Advances in Intelligent Systems and Computing, vol 732. Springer, Singapore

13-Stallings, W.: Operating Systems Internals and Design Principles, 7th ed. Prentice Hall, Pearson Publications 2015

14-Somani, M.J.S., Chhatwani, M.P.K.: Comparative study of different CPU scheduling algorithms. IJCSMC, 310–318 (2013)

15-Shyam, R., Nandal, S.K.: Improved mean round robin with short job first scheduling. Int. J. Adv. Res. Comput. Sci. Software Eng. (2014). ISSN: 2277 128X

16-Jezreel Ian C. Manuel1, Rey Benjamin M. Baquirin and other, Fittest Job First Dynamic Round Robin (FJFDRR) scheduling algorithm using dual queue and arrival time factor: a comparison Jezreel Ian C. Manuel et al 2019 IOP Conf. Ser.: Mater. Sci. Eng. 482 012046
https://doi.org/10.1088/1757-899X/482/1/012046

17-Somula R., Nalluri S., Nalla Karuppan M.K., Ashok S., Kannayaram G. (2019) Analysis of CPU Scheduling Algorithms for Cloud Computing. In: Satapathy S., Bhateja V., Das S. (eds) Smart Intelligent Computing and Applications. Smart Innovation, Systems and Technologies, vol 105. Springer, Singapore
https://doi.org/10.1007/978-981-13-1927-3_40