Diyala Journal of Engineering Sciences

ISSN 1999-8716

Vol. 02, No. 02, pp. 39-52, December 2009

CPU Scheduling Simulation

Huda M. Salih, Yousra A. Fadil

College of Engineering, University of Diyala

(Received:16/5/2008; Accepted:6/9/2009)

ABSTRACT - CPU scheduling is the basis of multiprogrammed operating system. By switching the CPU among processes, the operating system can make the computer more productive. The objective of multiprogramming is to have some process running at all times, in order to maximize CPU utilization. This paper presents a simulator that uses graphical representation to convey the concepts of various scheduling algorithms for a single CPU. It allows the user to test and increase his understanding of the concepts studied by making his own scheduling decisions, through the very easy graphical user interface of the simulator. It graphically depicts each process in terms of what the process is currently doing against time. Using this representation, it becomes much easier to understand what is going on inside the system and why different set of processes are candidate for allocation of the CPU at different time.In this paper, we introduce the basic scheduling concepts and present several different CPU – scheduling algorithms in visual way.

Keywords: CPU Scheduling, Multiprogramming system, Process.

---

1. INTRODUCTION

The process management is one of the most important principles in operating system, without management multiprograms can't execute at the same time because each process should finish before another process starts execution. But with process management multiple process can execute at the same time synchronously. So, a process is a program under execution, which competes for the CPU time and other resources(1). We provide a description about the context switching which plays a major role in providing multiple process at the same time. The context switch is an essential feature of a multitasking operating system(2).

The paper illustrate the concept of process states through the figure(1) that show the life time of processes(3). Finally, we show the important of process control block through the figure(2) that contains many pieces of information associated with a specific process, (i.e.;) the PCB represents a record or a data structure that holds information about each process(4).

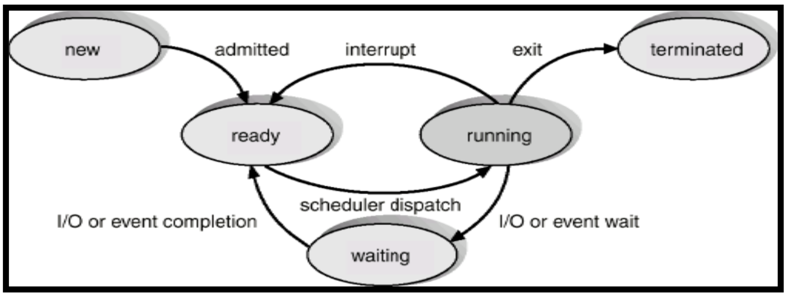

new
admitted
interrupt
exit
terminated
ready
running
scheduler dispatch
I/O or event completion
I/O or event wait
waiting

Fig.(1):- Diagram of Process State (4)

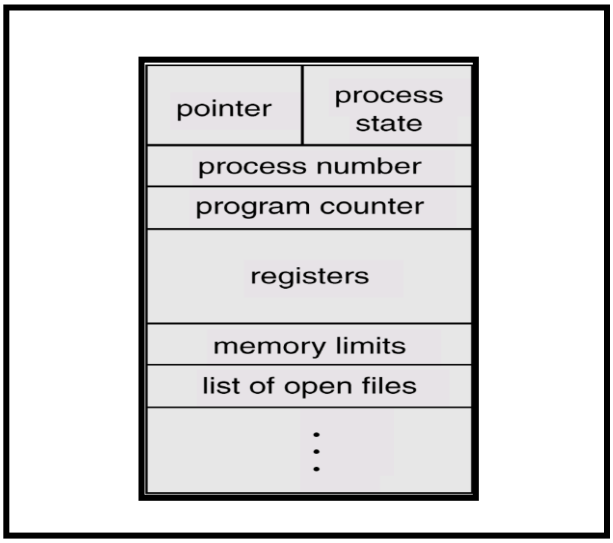

pointer
process state
process number
program counter
registers
memory limits
list of open files
•
•
•

Fig.(2):- Process Control Block (PCB)(4)

2- DESIGN SYSTEM FOR CPU SCHEDULING

Building system that will implement the concepts of process schedules in visual way and the system must able to show to the user how the process will be execute and what is the highest priority and what is the first process to execute. The system allow user to choose the type of schedule (FCFS, Round Robin, priority, short job first, shortest remaining time). On most multitasking systems, only one process can truly be active at a time - the system must therefore share its time between the execution of many processes. This sharing is called scheduling(5). Different methods of scheduling are appropriate for different kinds of execution. There are many type of scheduling

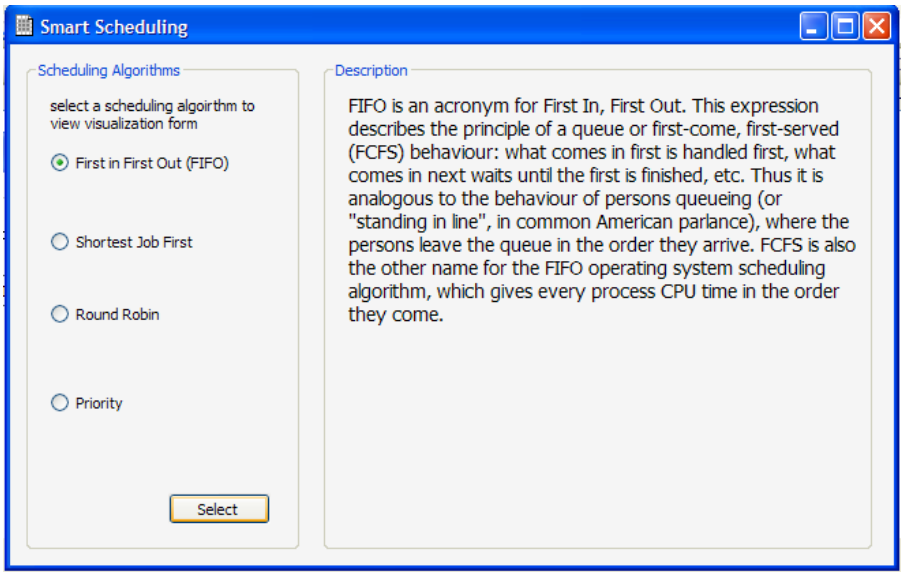
- Queuing. This is appropriate for serial or batch jobs like print spooling and requests from a server. There are two main ways of giving priority to the jobs in a queue. One is a first-come first-served (FCFS) basis, also referred to as first-in first-out (FIFO); the other is Sorted queue, in which the elements are regularly ordered according to some rule. The most prevalent example of this is the *shortest job first* (SJF) rule.

- Round-robin. This is the time-sharing approach in which several tasks can coexist. The scheduler gives a short time-slice to each job, before moving on to the next job, polling each task round and round. This way, all the tasks advance, little by little, on a controlled basis(6)(7).

The system should be able to allow user to add process ,delete process and show how will execute in visual way. Using C# language, where the C# is a modern object- oriented language.

2.1 The system form and entering form

The graphical user interface will include a main window figure(3) that holds scheduling algorithm selection from an option list and a command for launching the sub window which holds specific data about the selected algorithm where the user can manipulate with the scheduling execution.



select a scheduling algorithm to view visualization form

● First in First Out (FIFO)

○ Shortest Job First

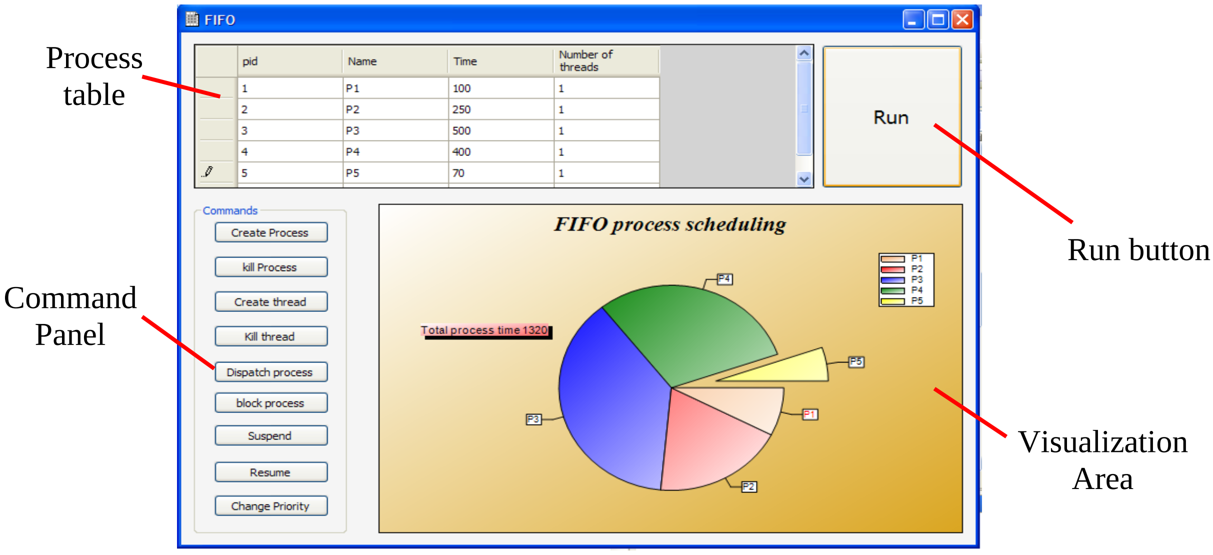
○ Round Robin

○ Priority

FIFO is an acronym for First In, First Out. This expression describes the principle of a queue or first-come, first-served (FCFS) behaviour: what comes in first is handled first, what comes in next waits until the first is finished, etc. Thus it is analogous to the behaviour of persons queueing (or "standing in line", in common American parlance), where the persons leave the queue in the order they arrive. FCFS is also the other name for the FIFO operating system scheduling algorithm, which gives every process CPU time in the order they come.

Fig.(3):- The Main Window

When the user presses the select button a new window should be viewed to him containing data fields for the selected scheduling algorithm and some commands for managing the execution of processes and threads.


Process table
FIFO
Run
Run button
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
Command Panel
FIFO process scheduling
Total process time 1320
P1
P2
P3
P4
P5
Visualization Area

pid	Name	Time	Number of threads
1	P1	100	1
2	P2	250	1
3	P3	500	1
4	P4	400	1
5	P5	70	1

Fig.(4):- Visualization Window

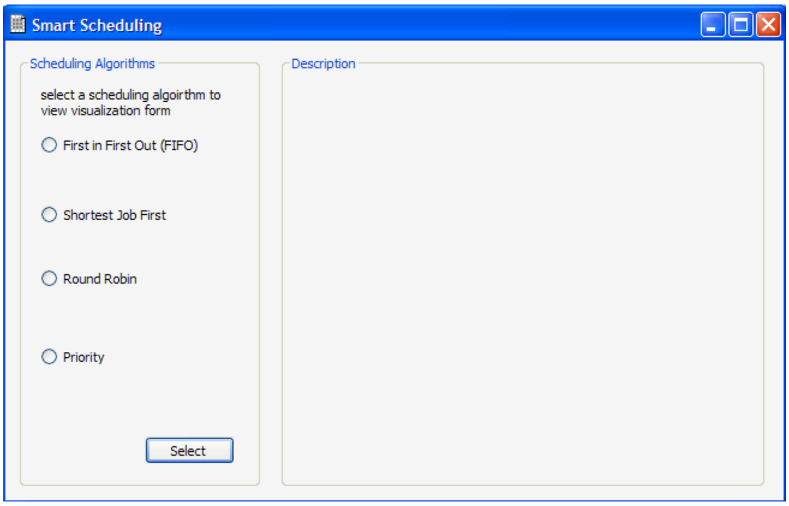
Each scheduling algorithms will have its unique representation, in the FIFO algorithm for example as shown in the above figure(4), the running process is the process that is sliced out from the pie chart.

The figure (4) represents all processes and their information and it allows the filling of an initial data, this data can be change while run time with the help of the commands in the command panel.

2.1.1. Interaction scenario

This scenario represents the interaction between the user and the system.

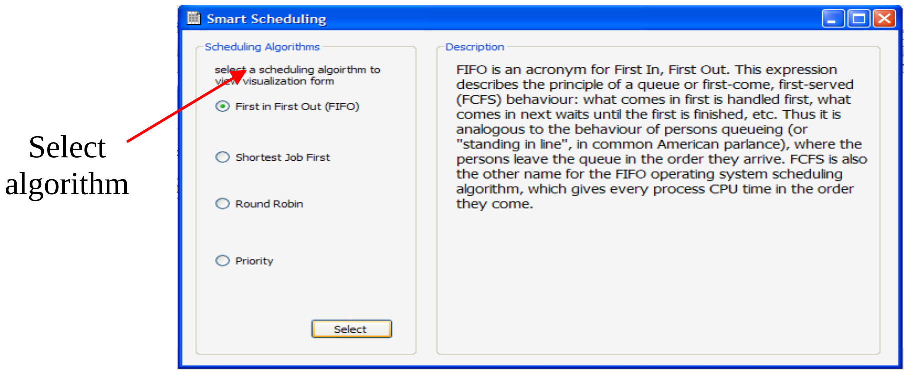
- The user enters the system.
- The system views the main window.


Smart Scheduling
Scheduling Algorithms
Description
○ First in First Out (FIFO)
○ Shortest Job First
○ Round Robin
○ Priority
Select

select a scheduling algorithm to view visualization form

Fig.(5):- The Window Viewed When The User Enters The System

- In the main window the user selects the type of scheduling algorithm he wants to visualize.


Select algorithm
Smart Scheduling
Scheduling Algorithms
● First in First Out (FIFO)
○ Shortest Job First
○ Round Robin
○ Priority
Select
Description

select a scheduling algoirthm to view visualization form

FIFO is an acronym for First In, First Out. This expression describes the principle of a queue or first-come, first-served (FCFS) behaviour: what comes in first is handled first, what comes in next waits until the first is finished, etc. Thus it is analogous to the behaviour of persons queueing (or "standing in line", in common American parlance), where the persons leave the queue in the order they arrive. FCFS is also the other name for the FIFO operating system scheduling algorithm, which gives every process CPU time in the order they come.

Fig.(6):- The Selection of Algorithm

- The system views a sub window which is the scheduling window and contains initial fields and specific data for each algorithm.

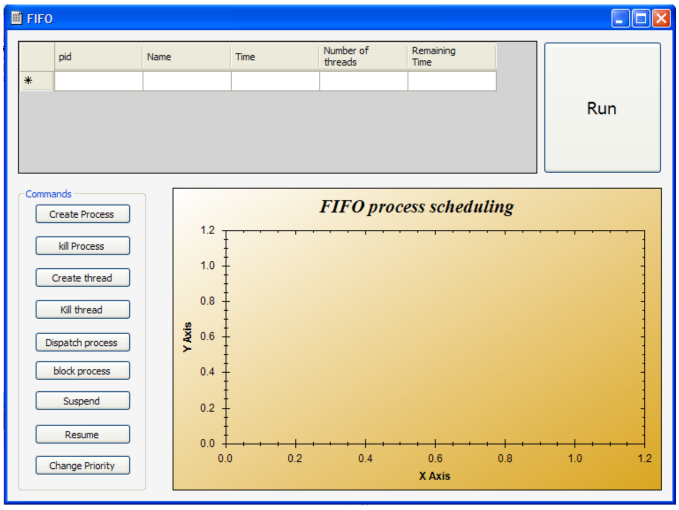
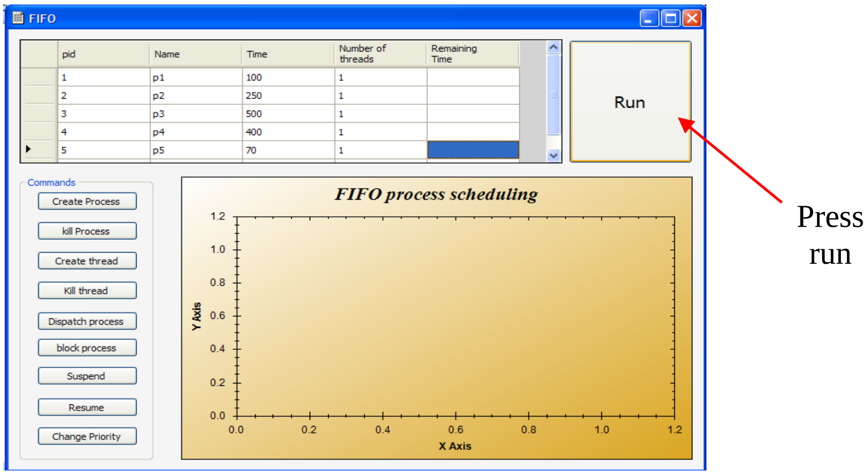

FIFO
pid
Name
Time
Number of threads
Remaining Time
*
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
Y Axis
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
X Axis

Fig.(7):- After Selecting the Algorithm

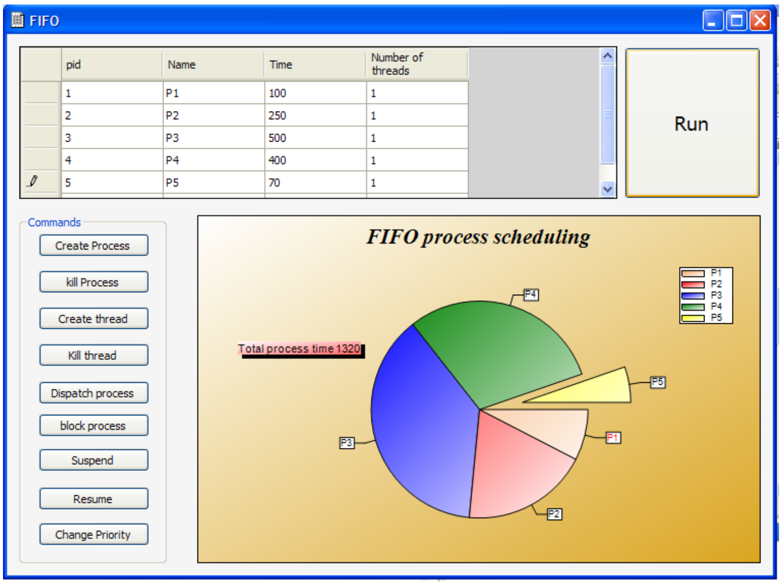
- The user sets the initial data and presses run.


FIFO
Run
Press run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
Y Axis
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
X Axis

pid	Name	Time	Number of threads	Remaining Time
1	p1	100	1	
2	p2	250	1	
3	p3	500	1	
4	p4	400	1	
5	p5	70	1	

Fig.(8):- Before Starting the Visualize

The system starts the scheduling and views it in a visual manner.



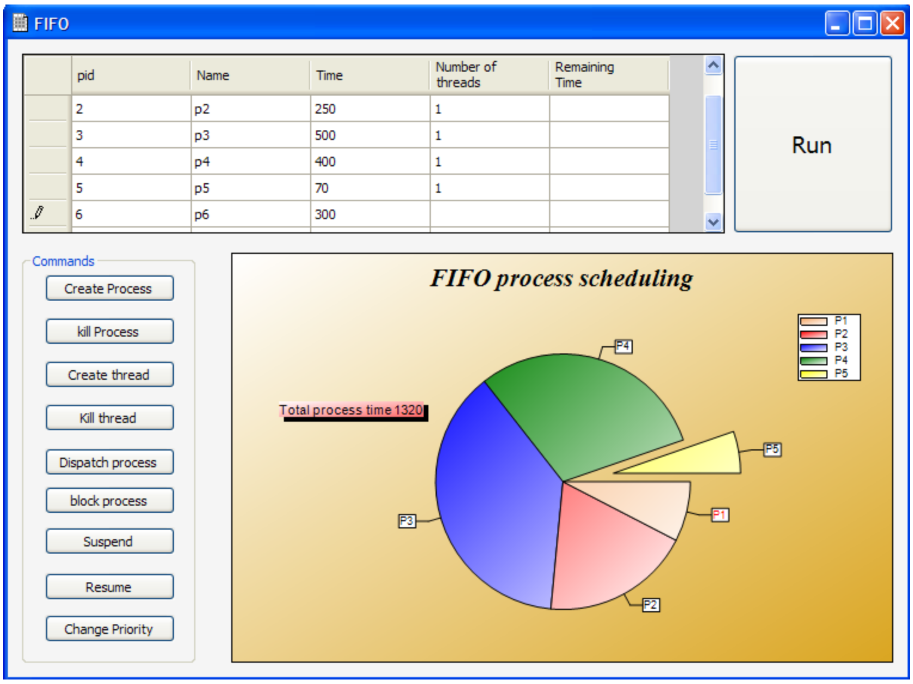
pid	Name	Time	Number of threads
1	P1	100	1
2	P2	250	1
3	P3	500	1
4	P4	400	1
5	P5	70	1

Total process time 1320

FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
P1
P2
P3
P4
P5

Fig.(9):- Starting the Visualize

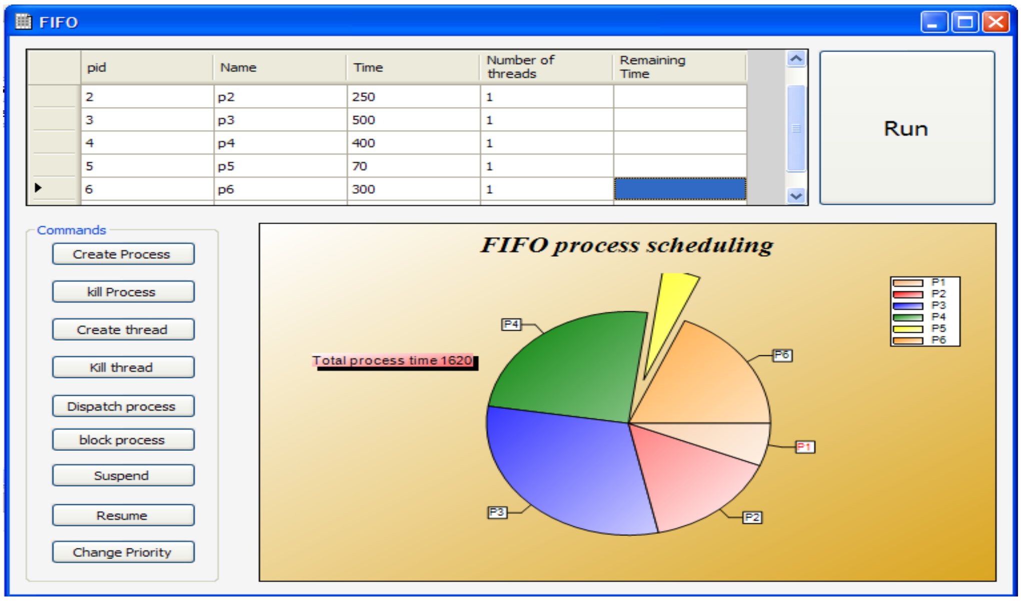
- The user can then do some commands like creating processes or threads and killing them.
- When the user calls a command, the system will affect the scheduling with the command and the visualization will change according to it.
- When the user wants to restart the scheduling, he has only to press the restart command then the system will end the current scheduling and set it to the beginning of a new scheduling,
- When a user wants to change the scheduling, he may press the change scheduling and the current window will release resources and let him start a new window with a new scheduling process.
- When a user presses the exit command all the processes and threads will terminate and the system will release all resources.
- When the user wants to create a new process, he fills down a new row in the table with the process values then press's create process button.


FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
Total process time 1320
P1
P2
P3
P4
P5

pid	Name	Time	Number of threads	Remaining Time
2	p2	250	1	
3	p3	500	1	
4	p4	400	1	
5	p5	70	1	
6	p6	300		

Fig.(10):- Creating Process

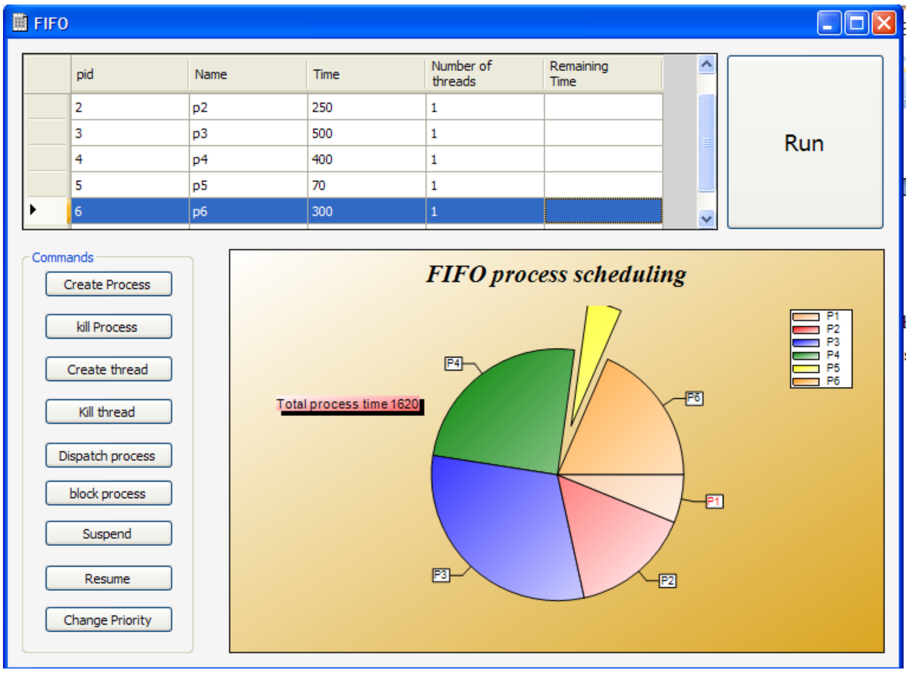
The result for the new process creation will affect the visualization by adding a new slice for that process.


FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
Total process time 1620
P1
P2
P3
P4
P5
P6

pid	Name	Time	Number of threads	Remaining Time
2	p2	250	1	
3	p3	500	1	
4	p4	400	1	
5	p5	70	1	
6	p6	300	1	

Fig.(11):- Creating Process Slice

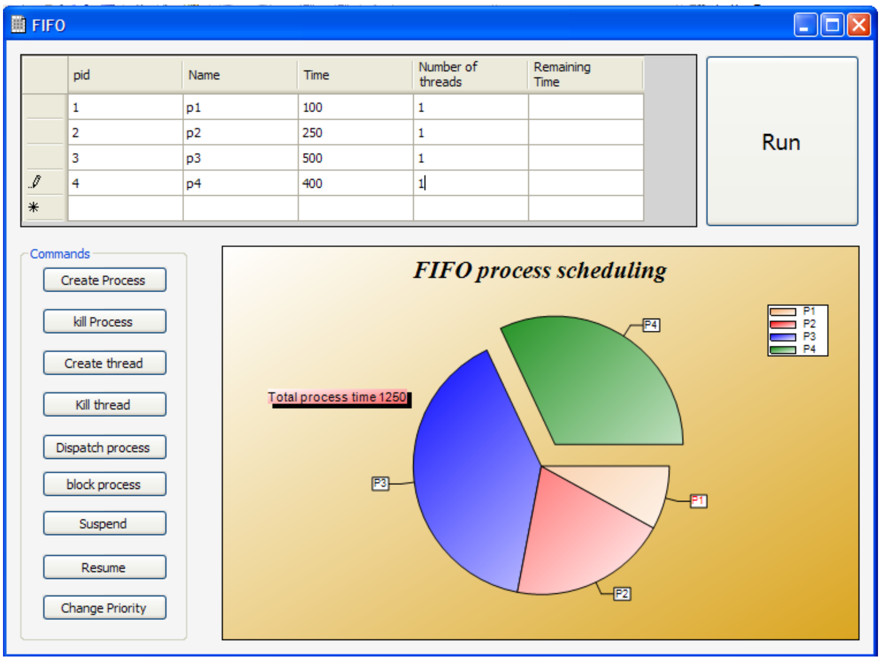
- As the system allows the creation of a process the system will allow the killing of them by simply selecting the process to kill from the table and pressing the kill process button.


FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
Total process time 1620
P1
P2
P3
P4
P5
P6

pid	Name	Time	Number of threads	Remaining Time
2	p2	250	1	
3	p3	500	1	
4	p4	400	1	
5	p5	70	1	
6	p6	300	1	

Fig.(12):- Killing Process

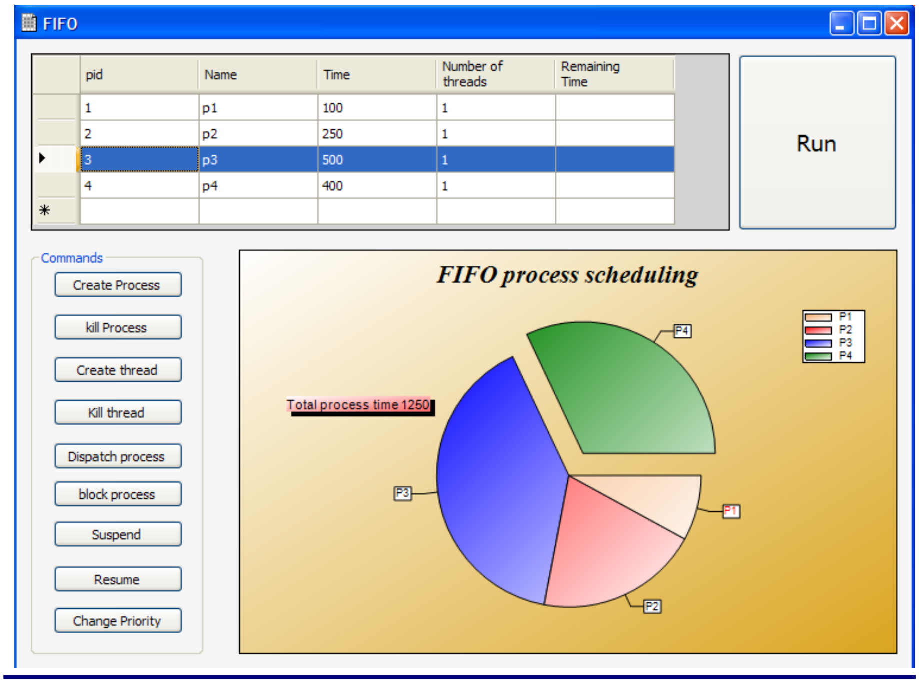
The result for process killing is shown in figure(13)


FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
Total process time 1250
P1
P2
P3
P4

pid	Name	Time	Number of threads	Remaining Time
1	p1	100	1	
2	p2	250	1	
3	p3	500	1	
4	p4	400	1	

Fig.(13):- Removing Process Slice

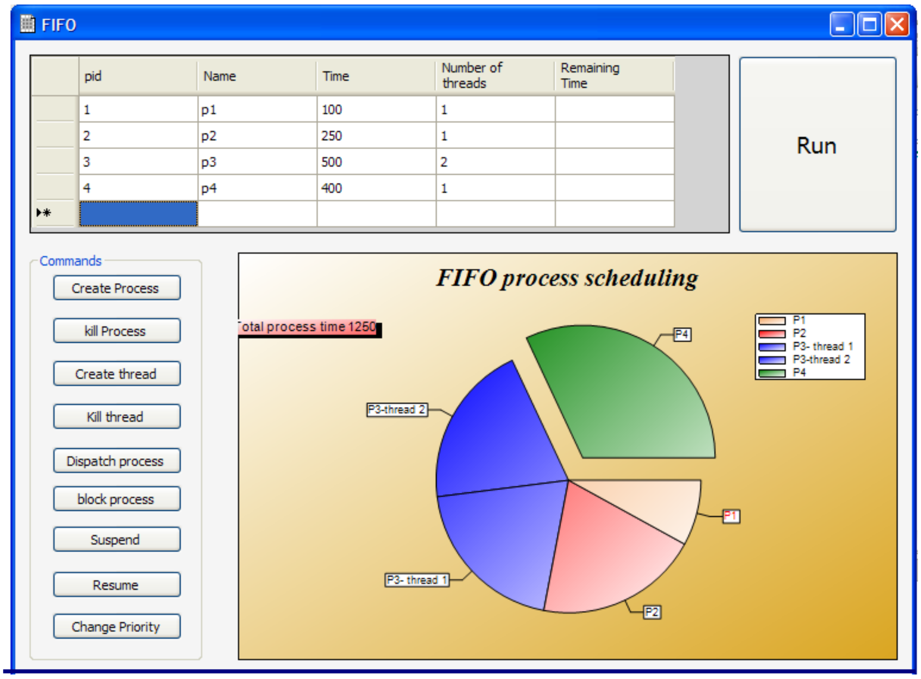
- The user may even create a thread for a certain process by selecting the process and pressing create thread.



pid	Name	Time	Number of threads	Remaining Time
1	p1	100	1	
2	p2	250	1	
3	p3	500	1	
4	p4	400	1	

FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
Total process time 1250
P1
P2
P3
P4

Fig.(14):- Creating Thread

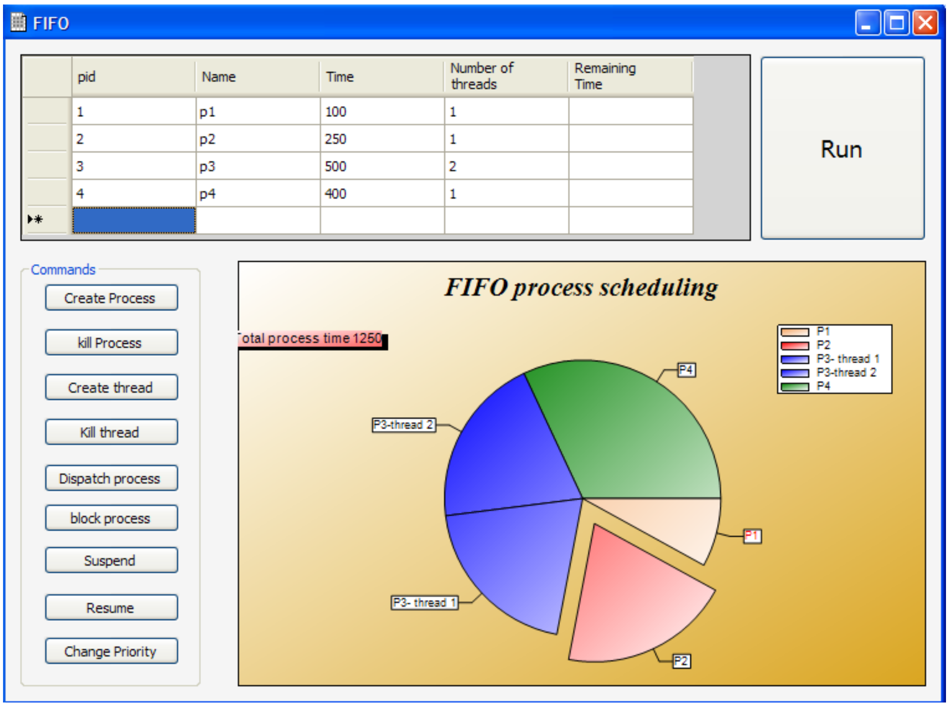


pid	Name	Time	Number of threads	Remaining Time
1	p1	100	1	
2	p2	250	1	
3	p3	500	2	
4	p4	400	1	

FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
otal process time 1250
P1
P2
P3- thread 1
P3-thread 2
P4

Fig.(15):- Cutting the Slice to two Pieces that Represents the Threads

- To dispatch process the user selects a process that is not running from the table and make it run by pressing the dispatch process button.

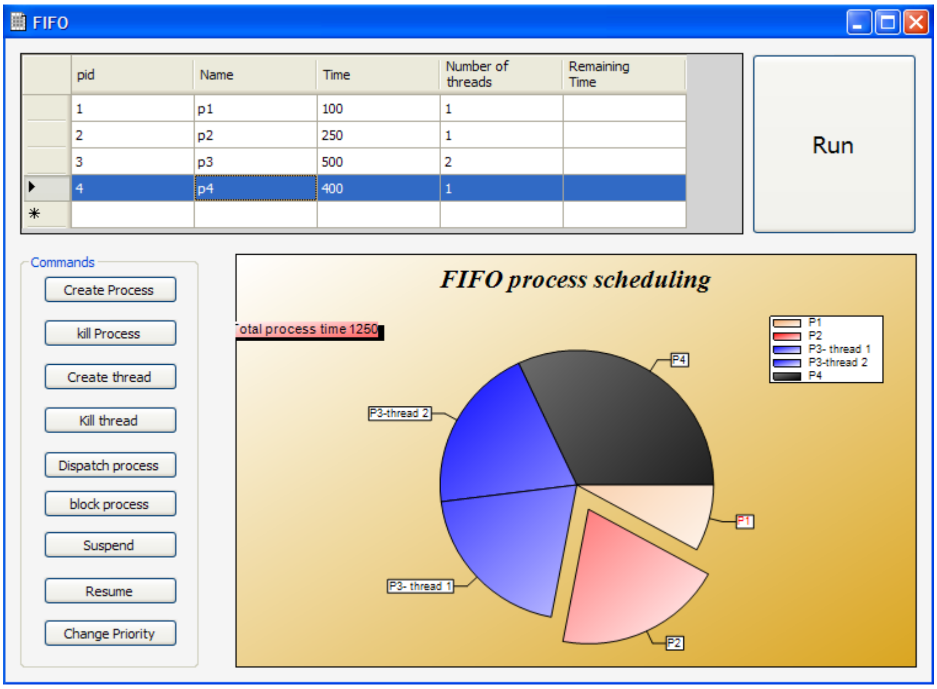


pid	Name	Time	Number of threads	Remaining Time
1	p1	100	1	
2	p2	250	1	
3	p3	500	2	
4	p4	400	1	

FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
otal process time 1250
P1
P2
P3- thread 1
P3-thread 2
P4
P4
P3-thread 2
P1
P3- thread 1
P2

Fig.(16):- Dispatching A Process

- To block a process the user selects the process from the table and presses the block button the system will turn its visual slice to black.

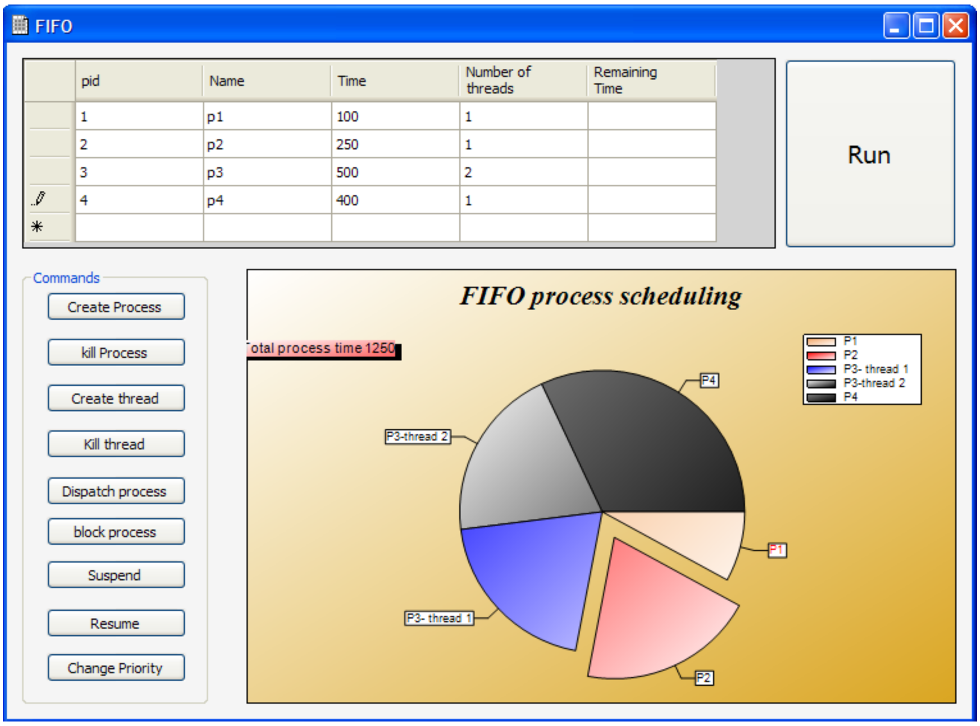


pid	Name	Time	Number of threads	Remaining Time
1	p1	100	1	
2	p2	250	1	
3	p3	500	2	
4	p4	400	1	

FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
otal process time 1250
P1
P2
P3- thread 1
P3-thread 2
P4
P4
P3-thread 2
P1
P3- thread 1
P2

Fig.(17):- Blocking A Process

- To suspend a process the user just selects the process and presses suspend and the system turns it to gray.



pid	Name	Time	Number of threads	Remaining Time
1	p1	100	1	
2	p2	250	1	
3	p3	500	2	
4	p4	400	1	

FIFO
Run
Commands
Create Process
kill Process
Create thread
Kill thread
Dispatch process
block process
Suspend
Resume
Change Priority
FIFO process scheduling
otal process time 1250
P1
P2
P3- thread 1
P3-thread 2
P4
P4
P3-thread 2
P1
P3- thread 1
P2

Fig.(18) Suspending A Process

- To resume a process the user selects a process that is already suspended and the system resume it to execution.
- To change the priority of a process in any other scheduling algorithm other than FIFO the user just set the new priority of the process in the table and updates it.

3- TESTING

System testing involves integration components that implement system functions or features and then testing this integrated system.

3.1. Black Box Testing:

Black Box Testing is testing technique having no knowledge of the internal functionality/structure of the system. This testing technique treats the system as black box or closed box. Tester will only know the formal inputs and projected results. Tester does not know how the program actually arrives at those results. Hence tester tests the system based on the functional specifications given to him. That is the reason black box testing is also considered as functional testing. This testing technique is also called as behavioral testing or opaque box testing or simply closed box testing. Although black box testing is a behavioral testing, Behavioral test design is slightly different from black-box test design because the use of internal knowledge is not illegal in behavioral testing (8).

3.1.1 Run

Function Being Tested	Initial System State	Input	Expected Output	Post Condition
User will choose to start running algorithm.	Click run Button	Data for algorithm	Display diagram	Starting execution of the algorithm with the selected data
User will choose to start running algorithm.	Click run Button	Wrong data	Error Message	Error Message

3.1.2. Fill Scheduling data

Function Being Tested	Initial System State	Input	Expected Output	Post Condition
User will choose to fill the initial algorithm data	Click fill Button	Data for algorithm	Visualization form is visible	Storing data and enabling run button
User will choose to start running algorithm.	Click fill Button	Wrong data	Error Message	Error Message

4- CONCLUSIONS

1. This system is built to show how scheduling algorithms are work in visual way and show how process are executing.
2. This system show process acts in different process states and show the difference between schedule algorithm.
3. From this system we obtain easy tools for the user to easy manages process and easy tools to the user to add or remove process and show the effect.

4. Easy tools to show how different process scheduling are working and the user have better understand the process schedule.
5. The system more realistic process model that can be configured easily by the user.

REFERENCES

1. Silberschatz, Galivin, Gagne, 2002, ”Operating System Concepts” ,Sixth Edition, John Wiley & Sons , Inc.
2. H.M. Deitel, 1990, ”Introduction to operating Systems”, Second edition, Addison-Wesley
3. J.Zahorjan and C.McCann, 1990, ” Processor Scheduling” in Shared–Memory Multi-processors” , proceedings of the Conference on Measurement and Modeling of Computer System.
4. F. Zabatta and K.Young, 1998, "A thread performance comparison: Windows NT and Solaris on a symmetric multiprocessor", proceedings of the 2nd USENIX Windows NT symposium.
5. Jones and Schwarz,1989,"Discussed thread scheduling Discussion concerning multiprocessor scheduling", Anderson et al.
6. E.W. Dijkstra,1968 ”The structure ofthe multiprogramming system” comm-unications of the ACM, volume number 5 ,page 341-346
7. Haldar and Subramanian, 1991, "Discuss fairness in processor sechduling in time sharing system".

محاكاة لجدولة وحدة المعالجة المركزية

يسرى احمد فاضل
مدرس مساعد
كلية الهندسة ـ جامعة ديالى

هدى محمد صالح
مدرس مساعد
كلية الهندسة ـ جامعة ديالى

الخلاصة

جدولة وحدة المعالجة المركزية هي أساس عمل نظام التشغيل متعدد البرامج. إذ عن طريق قيام وحدة المعالجة المركزية بالتحويل بين البرامج ، يصبح من الممكن أن نجعل نظام تشغيل الكمبيوتر أكثر إنتاجية بسبب هذا التحويل. الهدف من نظام التشغيل متعدد البرامج هو أن تكون هناك برامج تتنفذ على التوالي في جميع الأوقات ، من أجل تعظيم الاستفادة القصوى من عمل وحدة المعالجة المركزية.

هذا البحث يقدم محاكاة لعمل وحدة المعالجة المركزية باستخدام التمثيل الرسومي لتمثيل مفاهيم خوارزميات جدولة وحدة المعالجة المركزية وهو يتيح للمستخدم اختبار وزيادة فهمه للمفاهيم المدروسة عن طريق جدولة القرارات ، من خلال واجهة مستخدم رسومية سهله. إذ انه رسومياً يصور كل برنامج من حيث ما يقوم به حاليا في عمله ضمن زمن محدد. وباستخدام هذا التمثيل يصبح أسهل بكثير فهم ما يجري داخل النظام ،وما هو سبب ترشيح مجموعه مختلفة من البرامج وتخصيصها لوحدة المعالجة المركزية في أوقات مختلفة .

في هذا البحث ، تم تقديم مفاهيم أساسية للجدولة، وعرض العديد من خوارزميات جدولة وحدة المعالجة المركزية المختلفة بطريقة مرئية.

الكلمة المفتاحية : جدولة وحدة المعالجة المركزية، ونظام متعدد البرامج، البرنامج بصيغته التنفيذية.