جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

ENCS3390

Operating Systems

PROJECT

Memory Management Simulation

Names:

Sadeel Malassa	1191153	1
Shahd Abu-Daghash	1191448	3
Tala Al-Sweiti	1191068	4

Instructors:

Dr. Ahmad Afaneh

Dr. Ayman Hroub

Date: 10-01-2022

Abstract:

Memory management is defined as “the process of controlling and coordinating computer memory, assigning portions called blocks to various running programs to optimise overall system performance”[1]. Memory management relays in the middle between software and hardware as it combines both of them in its operations. This report will go through a simple simulation system that prescribes the virtual memory processing besides several page replacement algorithms.

Table of Contents

1 Theory

1.1 Scheduling

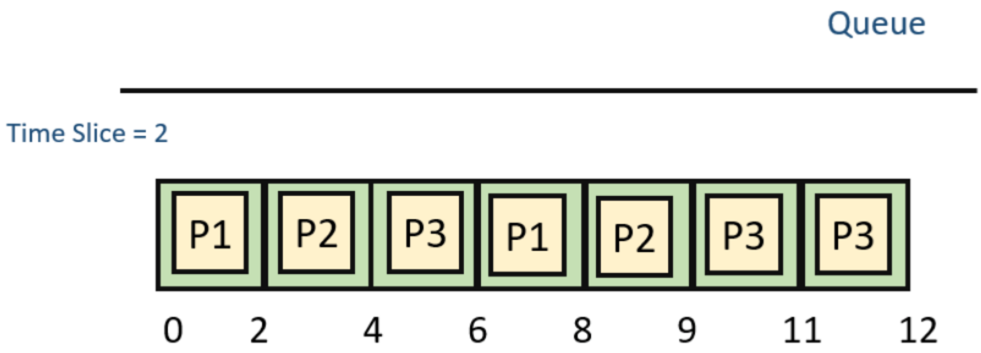
Many actions are taken on the computer per unit time,as a result many processes are sent to the Computer Processing unit; the CPU needed a tool to control such a large number of processes. And that's where the scheduler and its policies interfere. The scheduler at its simpler definition is used to arrange the process entering the cpu and the ones waiting per time unit, process priority and other multiple factors.

One of the main scheduling policies that will be handled in this report is the Round Robin (RR)

1.2 Round Robin scheduling

As the name suggests round robin scheduling policy came from a famous old principle called Round robin principle. Oldest most simple way to arrange any physical, mental, or even an every day operation. That suggests to give every element involved its equal amount of the shared pic.

In the operating System the CPU operates only one process at a time. Thus, to guarantee that all processes analysed in the CPU we use RR. It is a preemptive algorithm that shifts the CPU from one process to another. The round robin algorithm guarantees that the cpu is shifted after a fixed amount of time called quantum. Then after shifting the next process that is preempted will be added to the end of the queue and so on.



1.3 Paging

The main memory is a very essential component in the computer system. As it saves the data and so on for all its advantages. To know how paging works we first need to take a close look at the memory scheme. Memory is made out of blocks that have the same fixed size (frames). Pages are saved inside the frames. In order to have faster access to the pages and more organised memory, the following paging logical algorithms are used to decide which pages are to be replaced when needed.

The next two algorithms explained are the one displayed in the project presented later on.

- **Second chance FIFO**

  FIFO is shortcut for “First in First out” which means that the first page added is the first one to be replaced when needed. The typical way is to replace the first page. However, Second Chance FIFO is a more advanced way. It works using a circular linked list. When a page is found with its reference bit set to 1, it’s given another chance. On the other hand, if it’s found with a bit reference of 0, it’s removed and replaced with the new page.

- **Clock**

  The Clock algorithm implemented is close to the LRU. When using a certain page, a variable associated with the page is set to the time the page was used. The algorithm then checks for the page that has the lowest time because it would be the one that was not used recently and removes it.

2 Procedure and Discussion

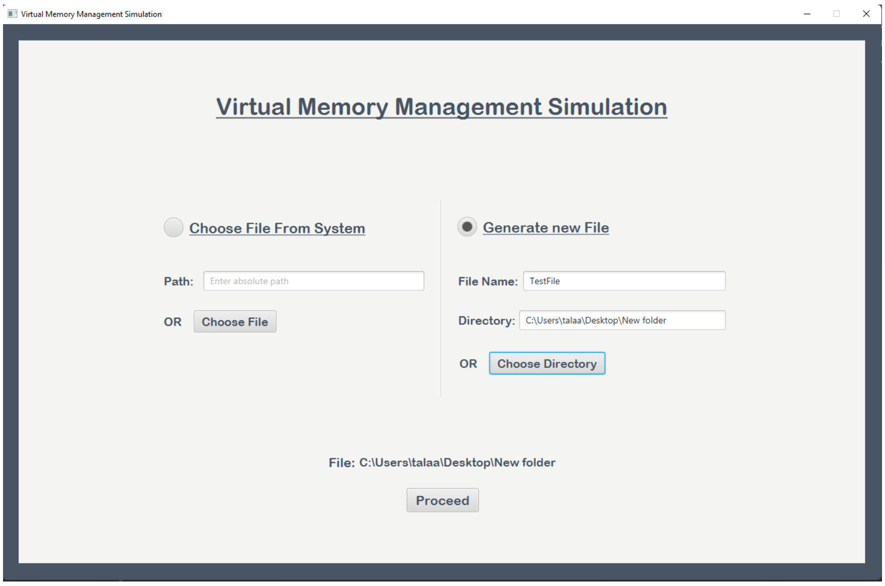
2.1 Graphical User Interface (GUI)

The program is shown to the user using a friendly user interface. It consists of two pages. The first page is used to select the file and the other one is to perform the simulation and view the results. The Main class is the one responsible for viewing the user interface.

To run the simulator a file is needed. The interface provided introduces two ways: the first one is running an already existing file, the second is giving the program the order of generating a file with the requested information.

In order to run the simulator you need to follow the following steps :

- ➔ selecting a file either by having an existing one or by generating a new one using the system(you need to the file name and a path to be saved on)
- ➔ Click “Proceed”



Virtual Memory Management Simulation

○ Choose File From System

● Generate new File

Path: Enter absolute path

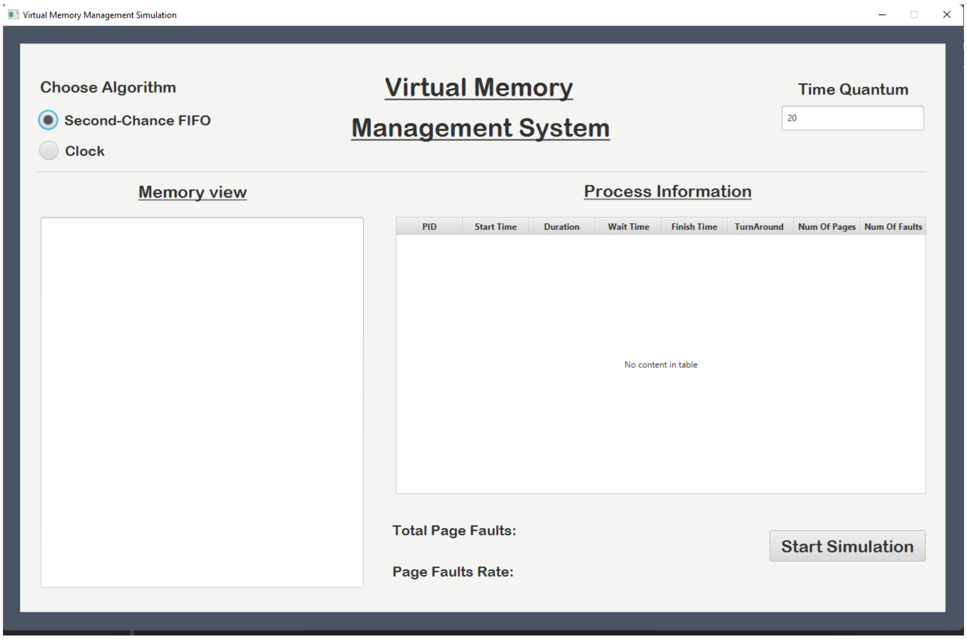
File Name: TestFile

Directory: C:\Users\talaa\Desktop\New folder

File: C:\Users\talaa\Desktop\New folder

A new Scene is then displayed, as shown below

- ➔ select what page replacement algorithm to follow (by default second choice FIFO)
- ➔ set quantum amount (by default quantum = 20)
- ➔ Click “Start Simulation”.



Virtual Memory Management Simulation

Choose Algorithm

● Second-Chance FIFO

○ Clock

Virtual Memory Management System

Time Quantum: 20

Memory view

Process Information

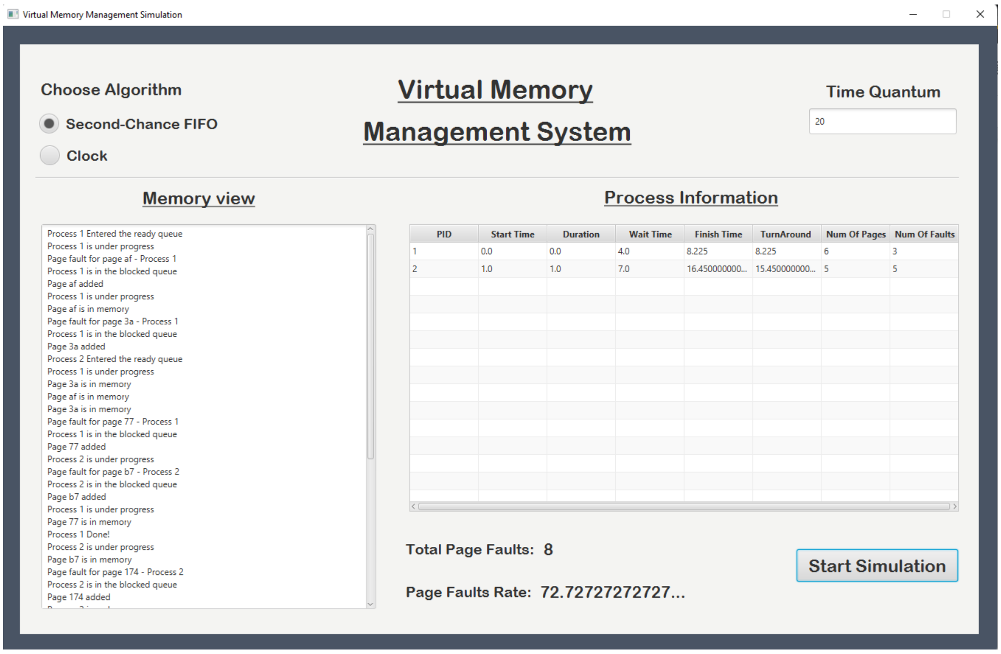
PID	Start Time	Duration	Wait Time	Finish Time	TurnAround	Num Of Pages	Num Of Faults

No content in table

Total Page Faults:

Page Faults Rate:

After that, the program presents the result in two views; memory view which shows the memory tracing of each process, next to it is the process view that displays all the processes information in an organised table.



Virtual Memory Management Simulation

Choose Algorithm

● Second-Chance FIFO

○ Clock

Virtual Memory Management System

Time Quantum: 20

Memory view

```
Process 1 Entered the ready queue
Process 1 is under progress
Page fault for page af - Process 1
Process 1 is in the blocked queue
Page af added
Process 1 is under progress
Page af is in memory
Page fault for page 3a - Process 1
Process 1 is in the blocked queue
Page 3a added
Process 2 Entered the ready queue
Process 1 is under progress
Page 3a is in memory
Page af is in memory
Page 3a is in memory
Page fault for page 77 - Process 1
Process 1 is in the blocked queue
Page 77 added
Process 2 is under progress
Page fault for page b7 - Process 2
Process 2 is in the blocked queue
Page b7 added
Process 1 is under progress
Page 77 is in memory
Process 1 Done!
Process 2 is under progress
Page b7 is in memory
Page fault for page 174 - Process 2
Process 2 is in the blocked queue
Page 174 added
```

Process Information

PID	Start Time	Duration	Wait Time	Finish Time	TurnAround	Num Of Pages	Num Of Faults
1	0.0	0.0	4.0	8.225	8.225	6	3
2	1.0	1.0	7.0	16.450000000...	15.450000000...	5	5

Total Page Faults: 8

Page Faults Rate: 72.72727272727...

RESULT Discussion

As seen in the previous figure, process 1 is inserted. thus a page fault will appear considering that when first initiating the memory has zero pages. a page fault is served and the next calling of the same page is not resolved as a page fault. you can notice that after the quantum specified, the process is pushed out the CPU and Process 2 is pushed in.

After that it goes through the algorithm to the end. giving the detailed results .

Results : [The page faults = 8, page fault rate = 72.72727273, Process information (ID, start time, Duration, Wait time, Finish time, Turn around, Number of pages, Number of faults)].

And as mentioned earlier you can notice all the memory tracing moves for the two running processes.

2.2 Simulator Class

The first class that is very important is the Simulator class. It contains all the data variables and data structures that are going to be used, such as the ready queue, blocked queue, and the Arrays that contain the processes and their threads.

Also, the Simulator class contains the methods used to generate a file randomly and read the file. It contains the method that it's going to start the execution of the program. To start the execution, a Scheduler thread is constructed and started.

2.3 Scheduler Class

This class is used through a thread and takes the processes from the ready queue, makes the process start or continue the trading of pages. If the page exists, it just calls a memory management thread to update the bit reference and the time of use of the process. However, if there is a page fault, a page replacement thread is initiated to solve the matter.

Keep in mind that the scheduling algorithm used is Round Robin, and so the process is removed from the CPU after a fixed quantum time.

2.4 Page Replacement Class

This class is used through a thread. It checks if there is a vacancy in the memory to write the new page from disk. If there is no space then it must use two page replacement algorithms. Both are mentioned in the above section and implemented as discussed. For writing the new page to memory and setting up the bit reference and the time of use, a memory management thread is used to serve the purpose.

2.5 Memory Management Class

This class is used by initiating a thread that this class is part of. Its purpose is to write to memory and prevent any other process of accessing the page table while doing so. Synchronisation is used in this part of the program to achieve that.

3 Conclusion

Throughout this project we were able to build a better understanding of the Concepts We have learned throughout the semester. Furthermore we were able to take a close look at a very essential component in our major even in our daily lives which is the computer and its way of operating all the actions starting from opening the desktop to downloading and running programs, even just clicking a button on the keyboard.

4 References

	Memory Management Definition, https://whatis.techtarget.com/definition/memory-management#:~:text=Memory%20management%20is%20the%20process,and%20in%20programs%20and%20applications.
	scheduling https://www.tutorialspoint.com/operating_system/os_process_scheduling.htm
	Round Robin algorithm https://www.guru99.com/round-robin-scheduling-example.html
	Paging https://www.geeksforgeeks.org/paging-in-operating-system/
	Paging in Operating System (OS): What is, Advantages, Example https://www.guru99.com/paging-in-operating-system.html