UTSA CS 3733
Operating Systems

Project 3 – Memory Manager (worth 50pts)

Instructor: Dr. Mohammad Imran Chowdhury

Due: 11/20/2025 at 11:59 pm

Purpose
In this project, you will implement the FIFO virtual memory page replacement policy used by a typical OS memory manager as discussed in class.

Grading
The entire project grade is worth 10% of your class grade (50 points). This is an individual project.

Prerequisites:

- A system running MacOS / Windows OS / Linux OS such as CS UTSA Fox Server or any Linux OS
- Access to the terminal/command line
- GCC Compiler

Project Description / Methodology:
As discussed in class First in, first out (FIFO): toss out the oldest page. Here is the sample example also discussed in class: see the lecture slide "**lecture_project_3_discussion.pptx**" for details.

- A process makes references to 5 pages: 1, 2, 3, 4, and 5
- Reference stream: 232152453252
- Physical memory size: 3 pages

Following is the final output returned by the First in, first out (FIFO) virtual memory page replacement policy used by a typical OS memory manager as discussed in class:

Memory page	2	3	2	1	5	2	4	5	3	2	5	2
1	2		*		5			*	3			
2		3				2				*	5	
3				1			4					2

- Total Page faults: **6 page faults** due to ***FIFO Policy misses***

Memory page	2	3	2	1	5	2	4	5	3	2	5	2
1	2		*		5			*	3			
2		3				2				*	5	
3				1			4					2

In this project, you'll implement this same FIFO virtual memory page replacement policy used by a typical OS memory manager.

Sample Input:
Your program should take the following file data as input. Here file name is inputFIFO.txt and has the following data:

```
3
FIFO
2
3
2
1
5
2
4
5
3
2
5
2
-1
```

Where first line value 3 means the size of the memory (i.e., in number of frames). The second line value FIFO indicates the name of the virtual memory page replacement policy. The remaining lines are the requested frame numbers. That is the program will keep accepting requested frame numbers until entered -1 which indicates the end of the input value i.e., the end of the reference stream.

Sample output:
Your program should provide the following as output:

```
Replacement Policy = FIFO
-------------------------------------
Page   Content of Frames
----   -----------------
02     02
03     02 03
02     02 03
01     02 03 01
05  F  05 03 01
02  F  05 02 01
04  F  05 02 04
05     05 02 04
03  F  03 02 04
02     03 02 04
05  F  03 05 04
02  F  03 05 02
-------------------------------------
Number of page faults = 6
```

You can see that in the program output above when Page No. i.e., frame number 05 was requested, at that time Frame Size was full hence it was marked "F" and the OS Memory manager used the FIFO strategy as discussed in class to replace frame no 02 with the requested frame number 05. The same process occurs later as well. Just check the row with the label "F".

For compilation:

Note: You can **save** the entire **program as FIFO.c file**.

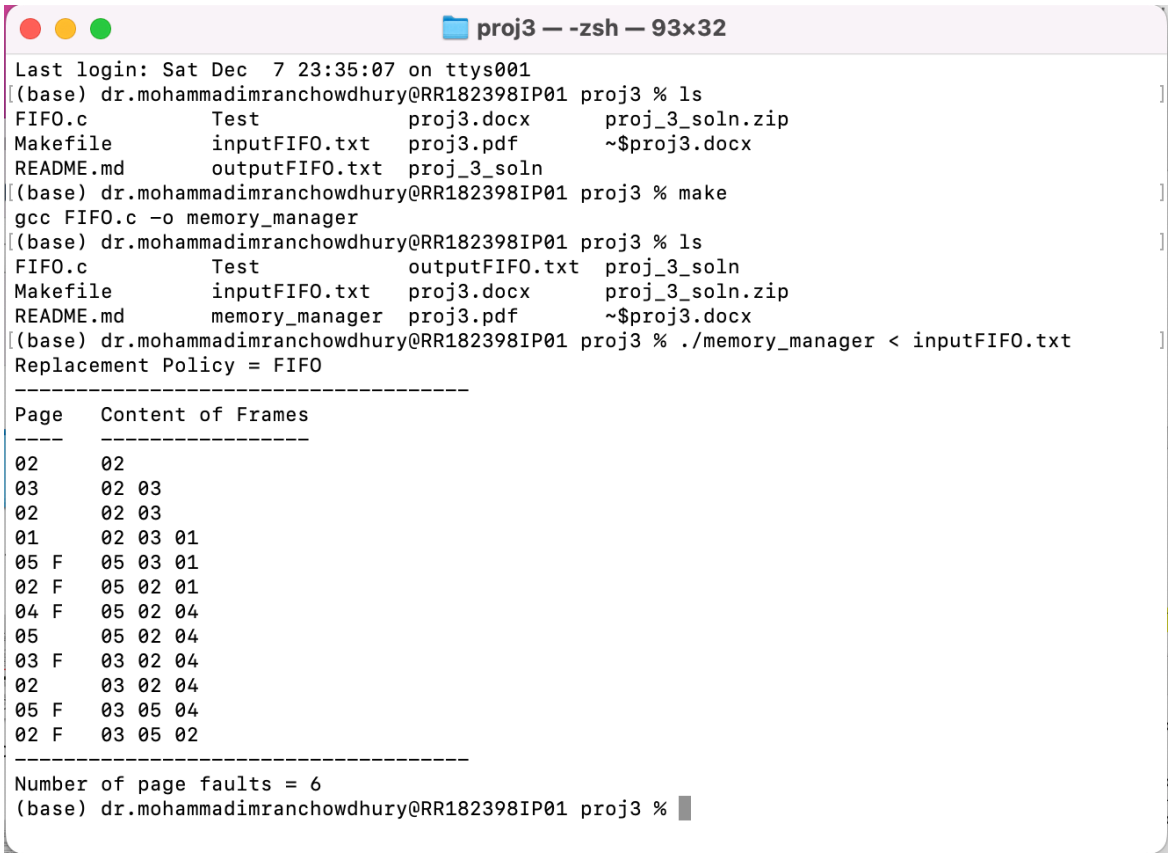
$ make

```
gcc FIFO.c –o memory_manager
```

For execution:

```
$ ./memory_manager < inputFIFO.txt
```

Sample Run: Your program output should be as follows



Project Submission

Please submit four (04) files: README file, Makefile, Screenshot of your program output file, and the completed version of the FIFO.c file i.e., the source code file, containing the implementation of the FIFO virtual memory page replacement policy used by a typical OS memory manager.as per the problem description/methodology mentioned above.

Detailed directions are below.

- The README file (can be a Word document or PDF), should contain:
  - Your CS UTSA ID, Banner ID, Name, and Email address
  - Any challenges you encountered along the way
- The Makefile file (must get compiled with the make utility)
- The completed version of the FIFO.c file i.e., your program file.
- The screenshot of your program output file. Like the one shown as a sample run in the project writeup.

Create a compressed file (.zip or .tar.gz files are accepted). **The README file, the MakeFile, the screenshot of your program output file, and the completed version of the FIFO.c file** should be turned into a compressed file such as a .zip file on or before the final submission due date. Navigate to the class Canvas page and click on the Project Submission link. Follow the links to submit your entire project files as a .zip file on Canvas.

Grading Rubric (out of 50 points)

Item	Points Possible
README file	5
Makefile	5
Screenshot of your program output	10
The completed version of the FIFO.c file i.e., the source code file	30

Late submission policy: As described in the syllabus, any late submission will be penalized with 10% off after each 24 hours late. For example, an assignment worth 100 points turned in 2 days late will receive a 20 point penalty. Assignments turned in 5 or more days after the due date will receive a grade of 0.

Go 'Runners!