



Lab4 Paging and Replacement

Objectives:

1. Implementing several page replacement algorithms.
2. Evaluating several page replacement algorithms.

Problem Statement:

It is required to simulate some of paging replacement algorithms. The required algorithms to be implemented are as follows:

- First In First Out (FIFO)
- Least Recently Used (LRU)
- Clock

Input/Output:

Your input will be from standard input. Your output will be from standard output. The input will contain the number of pages allocated to the process, the simulated algorithm (FIFO, LRU, or CLOCK), and then a sequence of page references like below:

```
3
FIFO
5
12
5
2
4
2
5
.....
-1
```

Note: The last line in the input is -1 (and is ignored)

For each run, you should print the following:

1. A trace recording page faults for each memory reference in the sequence.
2. Counter recording total page faults.

We will have the following results (*notice the two-digit page numbers*):

```
Policy = FIFO
-----
Page    Content of Frames
-----
05      05
12      05 12
05      05 12
02      05 12 02
04 F    04 12 02
02      04 12 02
05 F    04 05 02
-----
Number of page faults = 2
```

Notes:

- You must write your solution in C/C++ .
- Operating System: Linux
- Your input and output **must** follow the described format **strictly**. Your program will be automatically graded.
- Complete source code, commented thoroughly and clearly.
- You should work individually.
- Check the academic integrity policy of the course.

Tutorial:

- [A Simple Makefile Tutorial](#)

Deliverables:

- Source code: lab4.cc
 - A makefile that produced the binary executable lab4.
 - Your source program will be compiled by typing `make`, then will be tested by running the following command, which should produce nothing in case of successful test:
 - `cat input | ./lab4 | diff output -`
 - The turnin command **MUST** be executed with the following syntax
 - `turnin -c cc373 -p lab4 lab4.cc Makefile`
-