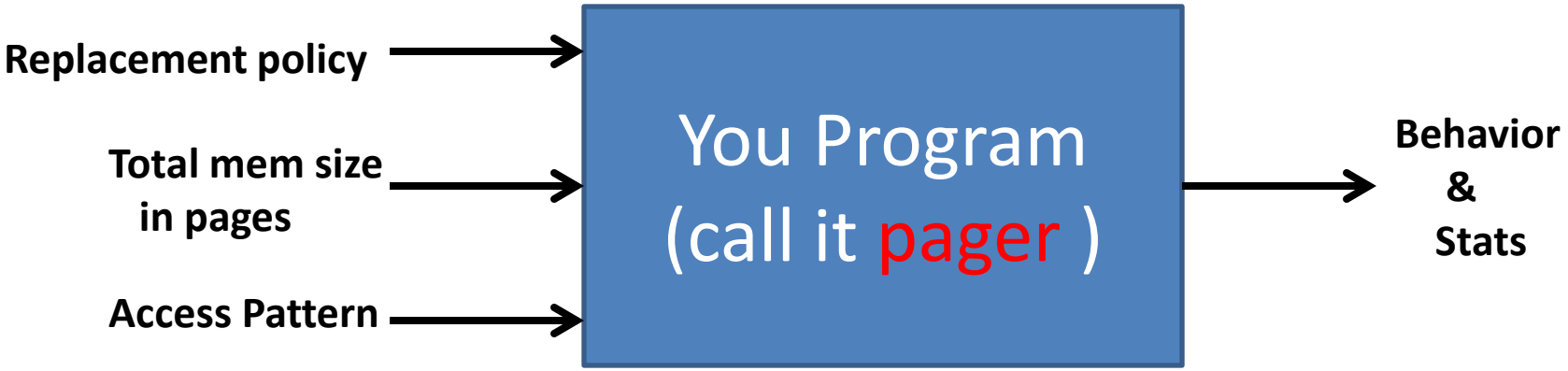
Page Replacement Policy

The Aim Of This Project

Implement several replacement policies and compare among them in terms of performance.



Exact command line: ***pager policy size file***

Replacement Policy

0: FIFO

1: LRU

2: Your own

You need to *invent* your own replacement policy that does the same or better than the best of FIFO or LRU.

For example: If we test with 3 files and FIFO has better results than LRU in file 1 and LRU has better results in files 2 and 3 → then your policy must be as good (or better) than FIFO in file 1 and as good (or better) than LRU in files 2 and 3.

Better = Less page faults

Total Memory Size in Page

- A positive non-zero number
- Indicates the total number of pages that can exist in memory at the same time

Access Pattern

- A text file
- Each line contains a page number, means that this page is accessed
- The file indicates page accesss over time.
- Page number > 0
- Example:

  7

  1

  9

Behavior and Stats

- A text file
- The file is divided into two parts:
  - Memory map
  - Statistics
- The memory map prints in a single line the numbers of the pages available in memory after each access
- Stats:
  - Percentage of page faults (#page faults/# accesses)

Example

pager 0 3 access.txt

Means: apply FIFO to a memory that can hold up to 3 pages and the accesses are in file access.txt

access.txt

7
8
9
7
4
5

The output filename is the input file name with extension dependent on replacement policy
Here for example: access.txt.fifo

Example:
Output

7
8
9
7
4
5

7 0 0
7 8 0
7 8 9
7 8 9
4 8 9
4 5 9

Percentage of Page faults = 0.83

Note: “0” indicates empty memory page slot

To Help You

- You are given a file pager.c that reads the command line arguments and loops over the memory accesses.
- You need to modify fifo(), lru(), and own()
- Feel free to update any data structures and add any extra functions you write as long as input/output of the program does not change.
- Actually, you can totally disregard the above file and implement the whole thing yourself, as long as the input/output format is the same.

Also to help you

We are giving you the input and output of a small example of a memory of 3 pages and a memory of 10 pages with input access patter test3.txt and test10.txt and the corresponding output:

- test3.txt.fifo and test3.txt.lru
- test10.txt.fifo and test10.txt.lru

What To Submit

- pager.c

- For honor students only … In addition to the source code:
  - input file sublru.txt of 20 entries that makes LRU better than FIFO for a memory of 10 pages
  - input file subfifo.txt of 20 entries that makes FIFO better than LRU for a memory of 10 pages