

CSCI 460: Operating Systems—Assignment 1 (8 marks)

This assignment is on memory management. You will implement a small paging system. Assume the following:

- a.** The (toy) memory has 16 page frames, each page is of 4096 bytes and pages are allocated from small address to large ones. Each job request is through your command line and of two types:

`% job_number bytes` — a job arrives with a request of certain memory

`% job_number 0` — delete this job

In addition, we have

`% print` — print the current memory status

`% exit` — exit the memory system

For example, if we have the following request:

`% 1 30000` — job 1 arrives with a request of 30000 bytes of memory

`% 2 30000` — job 2 arrives with a request of 30000 bytes of memory

`% 1 0` — delete job 1

`% 3 4090` — job 3 arrives with a request of 4090 bytes of memory

`% print` — print the current memory status

`% exit` — exit the memory system

Then, the current memory status is `job_3` (with one page) on frame 1, followed with 7 empty page frames, and then followed with the 8 pages for `job_2` in frame 9-16.

- b.** Remember that this is a normal paging system, so a job must either either completely reside in or out of the memory. For the replacement policy, you should use First In First Out (the time could be virtual, e.g., according to the order of the command line input — except `print`).

(1) Use the following data to test your program, print out your output in a text file, say `output-1.txt`.

Job Number	Size
1	35000
2	4096
3	4096
4	4096
5	4096
6	4096
7	4096
8	4096
2	0
4	0
6	0
8	0
9	12000
10	24000
print	
exit	

Date Due: 8:30pm on Wednesday, September 21, 2022 (on or before 8:30pm, Sep 21, 2022). Load your source code and output as two separate files on D2L in the folder **Assignment 1**, preferably in the form of **family_name – 1.c** and **output – 1.txt** (assuming that you are using C).

While discussion is allowed, you **MUST** finish the assignment by yourself and it does not matter which language you choose to use — of course, it does not hurt to put some comments in the source code on how to run it.

The TA (grader), Jared Weiss, will grade the assignment. If you do need to ask him questions regarding implementation, please email him directly for an appointment (the best time is MWF 10-10:50am).