ASSIGNMENT 2

Subject : Page Replacement Algorithms
TAs: Selim Yilmaz, Levent Karacan, Merve Ozdes
Due Date: 05.04.2018 23:59:59

1 Introduction

In this experiment, you will practice using binary search tree, unordered linked list and priority queue as heap tree data structures. You are expected to design and implement a program that will work like a basic page replacement scheme for memory management of operating systems. The algorithm that you are expected to implement will be a simpler version of FIFO(First In First Out) and Second Chance Page Replacement Algorithms.

2 Background

2.1 Virtual Memory

Virtual memory is a memory management technique of operating systems. It maps virtual adresses into physical adresses in computer memory. A memory management unit that is a hardware in CPU translates adresses to physical adresses and a software in OS provide a virtual adress space to use more memory than physically available memory space by using technique of paging.

2.2 Paging

The pages refer to the same-size data blocks in secondary storage and paging is to retrieve these pages to the main memory from secondary storage when a process wants to use specific data blocks i.e. pages. A program tries to to access pages that are not currently mapped to physical memory(RAM). This situation known as a page fault. The operating systems do the following steps to take page faults under the control:

- Determine the location of the data in secondary storage.
- Obtain an empty page frame in RAM.
- Load data to the page frame.
- Update the page table.
- Return control to the program.

If all page frames are non-empty, one of those frames must be chosen to empty for the following memory requirement. It is expected from an efficient page replacement algorithm that chosen frame must be least likely to be needed within a short time. There are various page replacement algorithms that try to do page replacement efficiently.

2.3 Page Replacement Algorithms

When all frames is full and the data you have to read is not in the memory, then you have to replace data in a frame with the data required. The main problem regarding the memory replacement is de- ciding which frame is to be replaced. The FIFO page replacement algorithm use simple first in first out strategy to discard next frame on the queue when a page fault occurs so that the oldest frame is removed from memory to open place for latest data sequence. In order to recognize the FIFO page replacement algorithm, each item has to have a sequence number (seq no).

Imagine we have a memory that can have 3 frames maximum and we have the following order of items that are read from the secondary storage:

Sequence	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
		0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
			1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1
Page Faults	x	x	x	x		x	x	x	x	x	x			x	x			x	x	x

Figure 1: First In First Out(FIFO) Page Replacement Algorithm

The structure of the memory flows as given in Figure 1. Once the all memory frames is full (after having 7, 0, 1), to make empty frame for the new item 2 the oldest used item 7 is removed from the memory and replaced with 2. The same process is repeated.

The Second Chance Page Replacement Algorithm is called as the Clock Replacement algorithm in some books as well is a FIFO(First in First out) replacement algorithm. It works by looking at the front of the queue as FIFO does, but instead of immediately paging out that page, it checks to see if its referenced bit is set. If it is not set, the page is swapped out. Otherwise, the referenced bit is cleared, the page is inserted at the back of the queue (as if it were a new page) and this process is repeated.

Sequence		7		0		1		2		0		3		0		4		2		3
	0	7	0	7	0	7	0	2	0	2	0	2	0	2	0	2	1	2	0	2
			0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	3
					0	1	0	1	0	1	0	3	0	3	0	4	0	4	0	4
Page Faults		x		x		x		x				x				x				x
Sequence		0		3		2		1		2		0		1		7		0		1
	0	2	0	2	1	2	0	2	1	2	0	2	0	2	0	7	0	7	0	7
	0	3	1	3	1	3	0	3	0	3	0	0	0	0	0	0	1	0	1	0
	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	0	1	1	1
Page Faults		x						x				x				x				

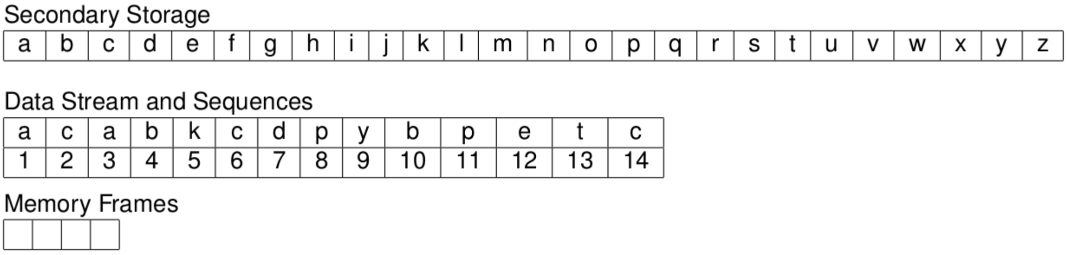
Figure 2: Second Chance Page Replacement Algorithm

For same sequence in Figure 1, replacement procedure is shown in Figure 2 for Second Chance Page Replacement Algorithm. As can be seen on the Table 2, a second chance bit is set for each frame according to the following rules:

- Each time a memory frame is referenced, set the ”second chance” bit to ONE (1) - this will give the frame a second chance...
- A new page read into a memory frame has the second chance bit set to ZERO (0)
- If the second chance bit is ONE, reset its second chance bit (to ZERO) and continue
- If the second chance bit is ZERO, replace the page in that memory frame.

3 Problem Definition

In this experiment, you are expected to design a simpler version of the FIFO and Second Chance page replacement algorithms and to show efficiency of algorithms by comparing time complexity and the number of page faults. You will also implement binary search tree and unordered linked list data structures to search data that will be read from memory. Similarly you will show searching cost for these algorithms as well. Assume that you have a memory and it has variable number of frames. The CPU wants to read a stream of data items to memory from secondary storage. You can see an example in Figure 3 with memory with 4 frames.



Secondary Storage

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Data Stream and Sequences

a	c	a	b	k	c	d	p	y	b	p	e	t	c
1	2	3	4	5	6	7	8	9	10	11	12	13	14

Memory Frames

Figure 3: Example Sequence and Memory Frames

3.1 Remarks About the Experiment

- The data is read from Secondary Storage.
- Your program must take input file path as argument.
- You will consider alphabetic order for priority queue.
- For main memory , an unordered linked list and a heap tree data structures have to be created for page replacement scheme.
- Binary search tree will be used for searching the data items in the frames.
- For second chance algorithm data structures must be modified with reference bit configuration..
- You must show number of page faults.
- You must also show elapsed time for whole sequence).

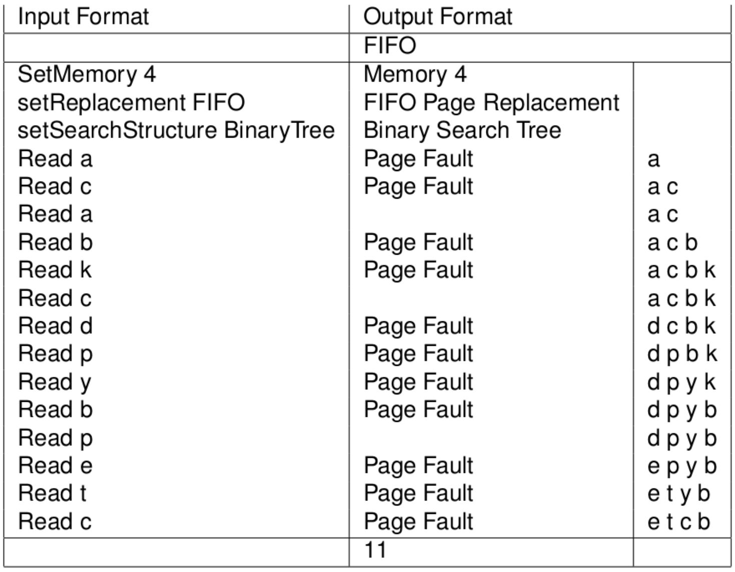
3.2 Scenario Samples

For example ,CPU wants to read data 'a' from memory:

- Check memory.(Binary search tree)
- if 'a' is not there(page fault) and memory have empty frame
  - Duplicate 'a' to empty memory frame and adjust data structures(Binary Search Tree, Unordered List,Queue or Heap Tree)
- if 'a' is not there(page fault) and memory do not have empty frame
  - You have to empty a frame from memory
  - Use FIFO or Second Chance Page replacement algorithms on appropriate data structures
  - Remove data in front element of data structures.(Consider Reference bit for Second Chance algorithm)
  - Duplicate 'a' to empty memory frame and adjust data structures.
- if 'a' is there,
  - Continue to sequences
  - Do not forget reference bit for Second Chance algorithm

3.3 Input and Output File

Consider example is given in Figure 4 and 5. We will supply codes for file operations to format output correctly.



Input Format	Output Format	
	FIFO	
SetMemory 4	Memory 4	
setReplacement FIFO	FIFO Page Replacement	
setSearchStructure BinaryTree	Binary Search Tree	
Read a	Page Fault	a
Read c	Page Fault	a c
Read a		a c
Read b	Page Fault	a c b
Read k	Page Fault	a c b k
Read c		a c b k
Read d	Page Fault	d c b k
Read p	Page Fault	d p b k
Read y	Page Fault	d p y k
Read b	Page Fault	d p y b
Read p		d p y b
Read e	Page Fault	e p y b
Read t	Page Fault	e t y b
Read c	Page Fault	e t c b
	11	

Figure 4: Input-Output File Format(FIFO)

Input Format	Output Format		
	Second Chance		
SetMemory 4	Memory 4		
setReplacement SecondChance	Second Chance Page Replacement		
setSearchStructure UnorderedList	Unordered Linked List		
Read a	Page Fault	a	
Read c	Page Fault	a c	
Read a		a c	
Read b	Page Fault	a c b	
Read k	Page Fault	a c b k	
Read c		a c b k	
Read d	Page Fault	a c d k	Second Chance a c
Read p	Page Fault	a c d p	
Read y	Page Fault	y c d p	
Read b	Page Fault	y b d p	
Read p		y b d p	
Read e	Page Fault	y b e p	
Read t	Page Fault	t b e p	Second Chance p
Read c	Page Fault	t c e p	
	11		

Figure 5: Input-Output File Format(Second Chance)

Notes

- Do not miss the deadline.
- Save all your work until the assignment is graded.
- The assignment must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.
- You can ask your questions via Piazza. (https://piazza.com/hacettepe.edu.tr/spring2018/bbm204) and you are supposed to be aware of everything discussed in Piazza.
- You will submit your work from https://submit.cs.hacettepe.edu.tr/index.php with the file hierarchy as below:

  <student id.zip>
  → src <DIR>
- The class name in which main method belongs should be **`Assignment2.java`**. All classes should be placed in `src` directory. Feel free to create subdirectories, corresponding the package(s), but each should be in `src` directory. **Your program must take input file path as argument.**
- This file hierarchy must be zipped before submitted (Not .rar, only .zip files are supported by the system)

Policy

All work on assignments must be done **individually** unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an **abstract** way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) **will not be tolerated**. In short, turning in someone else's work (from internet), in whole or in part, as your own will be considered **as a violation of academic integrity**. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.