Unit 11 Sample Problems - Virtual Memory (SOLN)

In this exercise, we will review the concepts of virtual memory.

- Length: 50 minutes with discussion.
- Questions: Q1, Q3-Q4, Q6-Q10 (optional: Q2, Q5, Q11-Q14)

1 Background

1. [Acuña] What is a big advantage of using virtual memory over main memory paging? [2 points]

   Ans: [Acuña]

   (Multiple answers are possible.)

   It completely decouples logical memory from a specific physical device (i.e. RAM) by providing an extra layer of redirection/abstraction.

2. [Acuña] For the idea of *paging* that was used to previously implement main memory, we had a single page table. Does virtual memory require adding an entirely new table? [2 points]

   Ans: [Acuña]

   Conceptually, no. Moving to a paging system simply means that we will need to store more information. The table will probably have extra columns to specify additional information about where it exists on secondary storage (address, swap file name, drive letter, etc).

3. [Acuña] Consider the following diagram:

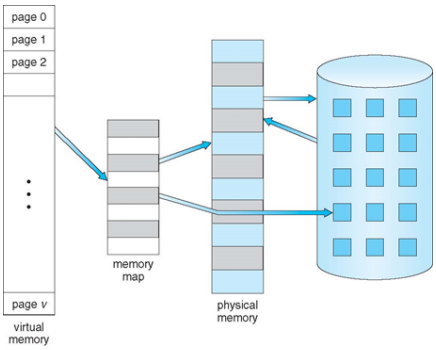

Figure 1: Overview of a typical virtual memory system. (Diagram from Operating System Concepts by Silberschatz, Galvin, and Gagne.)

Do the arrows represent addresses/pointers? Explain. [2 points]**Ans: [Acuña]**

No. Notice that, for example, there is an arrow from the hard drive to the physical memory - this would imply that at some point the hard drive would be storing addresses. This simply doesn't make sense in our model. Instead, they represent the flow of data between parts of a virtual memory system.

2 Demand Paging

4. [Acuña] Consider two programs: 1) A compression program that supports many compression formats. 2) Microsoft Paint. Which is more likely to have a higher relative number of pages in physical memory? **Explain.** Assume both are the same size, and have finished loading but used to do work. [2 points]

   Ans: [Acuña]

   The Paint program. Almost all of the functionality of the program is immediately available to the user and is likely to be used. Almost all the pages that store the program must be loaded to make sure all functionality is available. The compression program is likely to have much fewer because it supports many compression formats. Typically only one format is really loaded (the default such as ZIP or RAR), and using the others requires the user to specific select it.

5. [Bahremand] *Pure swapping* is the procedure of copying an entire process from main memory to secondary storage. What are the advantages and disadvantages for using *demand paging* over pure swapping? (Hint: page-replacement is used in both demand paging and pure swapping.) [2 points]

   Ans: [Bahremand]

   With demand paging, the operating system copies a disk page into physical memory only if an attempt is made to access it and that page is not already in memory; an example instance being a page fault.

   The advantage of using demand paging is that, unlike pure swapping, only the pages needed are loaded into memory. This allows for faster process execution as excess memory isn't allocated for unused memory addresses. Whereas with pure swapping, all memory for a process is swapped from secondary storage to main memory during the process startup.

   The disadvantage of using demand paging arises when the scenario involves continuously having to forcefully swap processes in and out. The concept of lazy loading technique applies to this situation. Processes are swapped from secondary storage to main memory repeatedly, and this can halt execution time as we keep waiting for processes to be loaded. With pure swapping, everything is loaded during process startup so we have less delay in execution.

6. [Acuña] Is it possible for a virtual memory system to have zero page faults? If so, **explain** when. [2 points]

   Ans: [Acuña]

   No. Initially RAM stores nothing so the the first time any piece of a data is needed, a page fault will be triggered. Even loading the virtual memory system itself will cause a page fault!

7. [Acuña] You are given a choice of using an array or list to store a collection of elements. Which is likely to result in fewer page faults? **Explain.** [2 points]

   Ans: [Acuña]

An array will result in fewer page faults. Considering the following: a page contains multiple pieces of data. In an array of integers, which is stored contiguously, this means that a page will contain data for multiple indices. So, if we have multiple accesses occurring, then one access will load a page, which may then be used by future accesses (which won't require another page fault because they can use the page already loaded).

3 Copy-On-Write

8. [Bahremand] Under Copy-On-Write (CoW), a new private copy is created only when a forked process actually writes to memory. **Describe** an example of CoW with respect to system calls other than fork and vfork. [2 points]

   Ans: [Acuña]

   Consider copying files to an external device like a USB stick. Sometimes the OS will show the file as instantly being copied, despite the fact that they are very large. This is a COW operation. Only when the USB stick is ejected, does the operating system carry out the actual write operation. If a user later decides not to copy a file, this approach saves time because the file won't need to be written. This also saves write cycles on the device. (In general, this may be seen as a lazy approach to file IO.)

4 Page Replacement

9. [Acuña] Consider the following reference string for page lookups: 4, 1, 6, 3, 1, 4, 2, 1. Compute the number of page faults that would occur with FIFO and give a trace of the cache, assuming 3 pages may be kept in memory. [2 point]

   Ans: [Acuña]

4	**1**	**6**	**3**	**1**	**4**	**2**	**1**
4	4	4	1	1	6	3	4
	1	1	6	6	3	4	2
		6	3	3	4	2	1
X	X	X	X		X	X	X

   A total of seven page faults will occur.

10. [Acuña] Consider the following reference string for page lookups: 4, 1, 6, 3, 1, 4, 2, 1. Compute the number of page faults that would occur with OPR and give a trace of the cache, assuming 3 pages may be kept in memory. [2 point]

    Ans: [Acuña]

4	**1**	**6**	**3**	**1**	**4**	**2**	**1**
4	4	4	4	4	4	4	4
	1	1	1	1	1	1	1
		6	3	3	3	2	2
X	X	X	X			X	

    (In last PF, can replace either 4 or 3). A total of five page faults will occur.

11. [Alvaran] What is an advantage and a disadvantage of using LRU page replacement over FIFO page replacement? [2 points]

Ans: [Alvaran]

An advantage that LRU page replacement has over FIFO page replacement is that LRU can prevent reduced performance in paging. By targeting the least-recently used page for replacement, the algorithm reduces chance of replacing a frequently used page. A disadvantage that LRU page replacement has with respect to FIFO page replacement is that LRU needs additional hardware to be implemented, whereas FIFO is compatible with the hardware of most current computer systems.

12. [Bahremand] Consider the following reference string for page lookups: 1, 2, 3, 1, 4, 2, 2, 1, 3, 4. Compute the number of page faults that would occur with LRU and give a trace of the cache, assuming 3 pages may be kept in memory. [2 point]

Ans: [Acuña]

1	2	3	1	4	2	**1**	2	3	4
1	1	1	1	1	1	1	1	1	4
	2	2	2	4	4	4	4	3	3
		3	3	3	2	2	2	2	2
X	X	X		X	X			X	X

A total of seven page faults will occur.

13. [Bahremand] Consider the following reference string for page lookups: 1, 2, 3, 1, 4, 2, 2, 1, 3, 4. Compute the number of page faults that would occur with MFU and give a trace of the cache, assuming 3 pages may be kept in memory. [2 point]

Ans: [Acuña]

1	2	3	1	4	2	**1**	2	3	4
1	1	1	1	4	4	4	4	4	4
	2	2	2	2	2	1	2	2	2
		3	3	3	3	3	3	3	3
X	X	X		X		X			

A total of five page faults will occur.

5 Allocation of Frames

14. [Acuña] On a home desktop system, would it make more sense (from a user experience perspective) to use global or local page replacement?

Ans: [Acuña]

Global would make more sense. It will yield higher performance since there are more options for pages which can be replaced. Notice also that the system is only supporting one user - that user is using some subset of the running processes. There is no chance that will end up slowing down another user's system requests.