Operating Systems–Mandatory assignment no. 3

Deadlocks, Main Memory, Virtual Memory

Exercise 1

Answer the following definitions and concepts:

1. **Define the conditions that a deadlock must meet.**
   There are 4 conditions necessary for a deadlock to occur these are:
   Mutual exclusion. At least one resource has to be held in non sharable mode so to prevent the other process from accessing it

   Resource holding: a process is holding a resource and requesting another one held by another process Resources can only be released voluntarily from the process holding it

   Circular wait: each process has to be waiting for a resource being held by another process waiting for the resource being held by the first proces

2. **What is the goal of ”Banker’s algorithm”? Explain what it is and how it works without mentioning implementation details.**

   The goal of the bankes algorithm is to avoid deadlocks. This is done by requiring all the processes to preemptively tell the algorithm the max amount of resources it can possibly need. The algorithm assumes that every process will at some point request its max amount of resources and subsequently terminate.

   Because of this fact for every resource request form a process the algorithm can check, given the resources currently used by all processes subtracted from the total amount of resources in the system. If i fulfill this request is it possible given a hypothetical set of resource requests to allow every process to achieve its max resource request and subsequently terminate.

3. **What is the difference between physical and logical address spaces**
   The physical address space is the physical addresses on the memory. The logical address space can be viewed like the processes own memory these addresses are generated by the cpu. It is a subset of the physical memory with an address space starting from 0 for each process. The addresses in the logical

space is shifted to correspond with the physical address space by the memory controller

If you mean paging it is very similar. Instead of the process being assigned a block of memory the memory is divided into blocks and the process is assigned one or more of these blocks. The logical address will in this case be a logical block (starts from 0 increases when over block size) and an offset. The physical address space will remain the same ish only in blocks

4. What is the device in charge of translating logical addresses into physical ones?How does it work?

The device in charge of translating from physical to logical memory is called the relocation register. This register shifts the logical addresses to match them to the process physical addresses

In a system with paging a paging table is used to translate the logical blocks into the blocks on the physical address space. To avoid having to do a lookup in main memory for the lookup a special fast lookup table called Translation lookaside buffer is used.

5. What is virtual memory and what is its function?

Virtual memory is the logical memory for all active processes. The size of this virtual memory can exede the size of the physical memory because not all parts of it need to be loaded at once. This is possible because programs tend to only use parts of their memory at any time, this means you can store big parts of a processes memory on disk until it is used.

This system of only loading what memory is used to disk allows for a more responsive system(apps can launch quicker because only the active part of their code needs to be loaded initially) and the ability to have more processes running at the same time.

6. What is segmentation?

Segmentation is a memory management feature that divides the main memory into segments. Where each segment can be allocated to a process. Segments are stored in a segment table with a segment base and a limit. Where the segment base is the starting physical address in memory, and the limit is the length of the segment. A process uses logical memory, which consists of two parts; a segment number and offset into the segment.

7. Explain the following concepts:”swapping”,”thrashing”and”shared page”.

Swapping: swapping is the process of reclaiming physical memory from processes. And writing it to disk.

Thrashing: thrashing is when the memory is full and and processes get their active memory swapped out before they can use it. This makes the processes constantly requesting pages which are swapped out before they can be used. A consequence of this is that the cpu utilization is going down because all the processes are idle, which in turn leads to more and more processes being generated.

Shared pages: a shared page is a page in memory that is shared between more than one process. This is possible because a lot of programs share libraries and code.

8. What is a Translation Lookaside Buffer (TLB) and what is its function?

The TLB (translation lookaside buffer) is used when translating virtual memory addresses to physical addresses. It is a cache that stores the most commonly used page table entries, to reduce the time consumed accessing a page table entry. It resides inside the MMU (Memory management unit) together with the CR3 register (on x86 -arch).

Exercise 2 - Paging

Consider the page table shown below for a system with 16-bit virtual and physical addresses and with 4096-byte pages. All numbers below are given in hexadecimal. (A dash for a page frame indicates that the page is not in memory.)

1. **How many bits are used for page numbers?**
   (2^16/(2^(16-4)) = (2^16/(2^12) = 2^4 = 4 bit are used
   4

2. **If you write the page number as a hexadecimal number, how many hex digits long it will be?**
   1

3. **How many hex digits long is the offset part?**
   3

4. **How many bits are in the offset part of the address?**
   12

5. **Write the formula – how can we translate from virtual addresses to physical addresses?**
   pageStart + offset = physical

6. **Convert the following virtual addresses to their equivalent physical addresses in hexadecimal.**
   Size of logical is 2^m and page size is 2^n page number = m-n

   a) 9EF5 = 0EF5 = 0x0EF5 is the physical address
   b) 1000 = 0x1FFF
   c) 7853 = PageFault
   d) 0FFF = PageFault

Page Number	Physical Frame Number
0	-
1	2
2	C
3	A
4	-
5	4
6	3
7	-
8	B
9	0

Exercise 3 - Implementation

Implement a program in Java that simulates Optimal Page Replacement.

Use the provided Java Maven project as a template. It already implements abstract ReplacementAlgorithm base class. There is also an example of FIFO replacement algorithm. Some unit tests are also provided.

Your task is to fill in code in OptimalReplacement class, and also create some unit tests for it in OptimalReplacementTest class.
The reference string includes comma-separated memory references without specific limits.

Example: 5, 4, 3, 2, 1
The process() method of the algorithm must return number of page replacements (not counting page faults which resulted in loading the page into an empty frame).

Write Unit-tests for your program. For each unit test give a short comment – why do you include this test? What does it test?

Please review the attached code in the .zip-archive.