Solution for Homework Assignment 4

Silberschatz et al. exercise 9.2 Explain the difference between internal and external fragmentation.

Internal fragmentation is the area in a region or a page that is not used by the process it is allocated to. The space is wasted until the process terminates. External fragmentation occurs when there is enough free space to satisfy a request for memory, but none of the free "holes" between processes in memory is large enough to satisfy the request.

Silberschatz et al. exercise 9.4 When a process is rolled out of memory, it loses its ability to use the CPU (at least for a while). Describe another situation where a process loses its ability to use the CPU, but where the process does not get rolled out.

This situation occurs every time an interrupt occurs.

Silberschatz et al. exercise 9.7 Why are page sizes always powers of 2?

Part of the paging algorithm is to break up the logical address into a page and offset. The easiest way to do this is to break the address up into k bits for the page number and l bits for the offset. Since each bit position represents a power of two, with l bits for the offset, we are forced to have pages with size 2^l.

Silberschatz et al. exercise 9.8 Consider a logical address space of eight pages of 1024 words each, mapped onto a physical memory of 32 frames. (a) How many bits are there in the logical address? (b) How many bits are there in the physical address?

The logical address requires 3 bits for the page number (there are 8 of them) and 10 bits for the offset (there are 1024 of them), for a total of 13 bits. The physical address requires 5 bits for the frame number (there are 32) and 10 bits for the offset (frame size = page size), for a total of 15 bits.

Silberschatz et al. exercise 9.10 Consider a paging system with the page table stored in memory. (a) If a memory reference takes 200 nanoseconds, how long does a paged memory reference take? (b) If we add associative registers, and 75 percent of all page-table references are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time, if the entry is there.)

(a) A paged memory reference in this system takes 400 ns (200 ns for the page table lookup and 200 ns for the reference itself). (b) With a TLB hit ratio of 0.75, we have an effective access time of $0.75 \times (0 + 200\text{ns}) + 0.25 \times (200\text{ns}+200\text{ns}) = 250\text{ns}$.

Silberschatz et al. exercise 10.1 Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

A page fault occurs when an access to a page that has not been brought into main memory takes place. The operating system verifies the memory access, aborting the program if it is invalid. If it is valid, a free frame is located and I/O is requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and the instruction is restarted.

We may optionally, after the I/O request, do a context switch to another ready process, and allow that process to run until the I/O is complete.

Silberschatz et al. exercise 10.2 Assume a page reference string for a process with m frames (initially all empty). The page reference string has length p with n distinct page numbers occurring in it. For any page-replacement algorithms, (a) What is a lower bound on the number of page faults? (b) What is an upper bound on the number of page faults?

(a) n. (b) p.

Silberschatz et al. exercise 10.3 A certain computer provides its users with a virtual-memory space of 2^{32} bytes. The computer has 2^{18} bytes of physical memory. The virtual memory is implemented by paging, and the page size is 4096 bytes. A user process generates the virtual address 11123456. Explain how the system establishes the corresponding physical location. Distinguish between software and hardware operations.

The virtual address in binary form is 0001 0001 0001 0010 0011 0100 0101 0110. Since the page size is 2^{12}, the page table size is 2^{20}. Therefore the low-order 12 bits 0100 0101 0110 are used as the offset into the page, while the remaining 20 bits 0001 0001 0001 0010 0011 are used as the offset into the page table. The page table entry provides the 6-bit frame number, which is combined with the 12-bit offset.

All of this work is done in hardware — I think the "Distinguish between hardware and software operations" is a bit of a red herring.

Silberschatz et al. exercise 10.9 Consider a demand-paging system with the following time-measured utilizations: CPU utilization 20%, Paging disk 97.7%, Other I/O devices 5%. Which (if any) of the following will (probably) improve CPU utilization? Explain your answer. (a) Install a faster CPU. (b). Install a bigger paging disk. (c) Increase the degree of multiprogramming. (d) Decrease the degree of multiprogramming. (e) Install more main memory. (f) Install a faster hard disk or multiple controllers with multiple hard disks. (g) Add prepaging to the page fetch algorithms. (h) Increase the page size.

The system is thrashing. If the level of multiprogramming is reduced or the amount of memory is increased, we could allocate more memory frames to the running processes, the system would page fault less frequently, and the CPU utilization would increase. A faster CPU or bigger disk or prepaging would not do anything, since the bottleneck would still be between the disk and main memory. Increasing the degree of multiprogramming and increasing the page size would only make things worse, since each process would have fewer frames available and the page fault rate would increase.

Silberschatz et al. exercise 10.11 Consider the following page reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, or seven frames? Remember all frames are initially empty, so your first unique pages will all cost one fault each: LRU replacement, FIFO replacement, Optimal replacement.

Number of frames	LRU	FIFO	Optimal
1	20	20	20
2	18	18	15
3	15	16	11
4	10	14	8
5	8	10	7
6	7	10	7
7	7	7	7