COMP 2432 Operating Systems

Tutorial 9

1. Virtual Memory.

Consider a virtual memory system implemented on a machine of physical memory size 32KB, based on 24-bit virtual addresses. The size of a memory page is 2KB.

(*a*) What is the virtual address space?
(*b*) What is the length of the physical address?
(*c*) What is the length of the offset field?
(*d*) What is the length of the page number field?
(*e*) How many frames are there in main memory?
(*f*) What is the length of the frame number?

2. Virtual Memory.

Consider a virtual memory system implemented on a machine of physical memory size 64KB, based on 32-bit virtual addresses. The size of a memory page is 1KB. Assume that *two additional bits* in the page table are used as the *valid-invalid bit* and the *dirty bit*. You may want to compare your answers with those in **Question 1**.

(*a*) What is the virtual address space?
(*b*) What is the length of the physical address?
(*c*) What is the length of the offset field?
(*d*) What is the length of the page number field?
(*e*) How many frames are there in main memory?
(*f*) What is the length of the frame number?
(*g*) What is the maximum possible size of the page table for a process?
(*h*) Suppose that an average process is of size of 1MB, what is the size of a page table for such an average process?

3. Page Table in Virtual Memory.

Consider the following reference string generated by a process *P* that occupies 8 pages (from page 0 to page 7), storing contents *A*, *B*, *C*, *D*, *E*, *F*, *G* and *H* respectively. There are only 4 memory frames available for this process *P* with frame numbers 42, 11, 24 and 35. Assuming that *FIFO page replacement algorithm* is used, *draw a diagram* to indicate the content of the frames and the page table after all the pages indicated by the reference string are accessed:

1 3 0 2 3 5 1 5 0 4

4. Memory Access Time.

Consider a page-based system supporting runtime address binding. There is a page table used to translate each virtual memory address into physical address. The main memory access time is 100 ns. TLB is used to improve the translation efficiency, through the use of associative cache. Assume that the cache has an access latency of 20 ns. What is the *effective memory access time* when locality implies a TLB hit rate of 95%? What is the value for a TLB hit rate of 99%?

With a simple virtual memory system based on a fast hard disk, it is possible to yield a page fault service time of 5 ms (inclusive of OS system call overhead). *Without* using TLB and assuming that the page fault rate is 1%, compute the *effective virtual memory access time*. If the locality of page reference improves so as to attain a page fault rate of 0.1%, compute the corresponding effective virtual memory access time.

Consider the use of TLB in the virtual memory system. Compute the *effective virtual memory access time* with TLB hit rate of 95% and page fault rate of 1%. Compare with the case of TLB hit rate of 99% and page fault rate of 0.1%.