

## Test Problems

- 4.1 Suppose you have 16M bytes of main memory. Using the list method. You have an overhead of eight bytes per memory block. Using the bitmap method, you use an allocation granularity of 128 bytes. How many blocks are there when the space overhead of both methods is the same? What is the average block size for this many blocks?
- 4.2 Consider the following page reference string 7,0,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,e How many page faults would occur in the case?
- a. LRU
  - b. FIFO
  - c. Optimal algorithms assuming three frames

Note that initially all frames are empty.

- 4.3 Assume that we have a paging system with 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% 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.
- 4.4 In two level nested loops, the outer index (i) runs from 1 to 5 and the inner index (j) runs from 1 to 10. The page faults seem to occur for every 7<sup>th</sup> innermost iterations. If it takes 0.02 micro second to load a new page – what is the extra time required because of occurrence of page faults.
- 4.5 Indi-Comp is a new computer system. It has 32 bits address space. The system employs paging with page of 4K size. Suppose the page table is stored in a semiconductor memory that requires 0.01 microseconds to load the page table. Processes are swapped every 10 millisecond. Each time a process runs – its page table need to be loaded. Compute the percentage of time that gets used in loading a page as an overload.

4.6 On scanning primary memory pm addresses 00..0. We find holes of the following sizes 10K, 15K, 5K, 32K and 2K. In what order would the allocation happen if we need to allocate segments having sizes 2K, 7K, 27K and 1K if we followed

- a. First fit policy
- b. Next fit policy
- c. Best fit policy