Tutorial 4

1. Select the correct alternative in each of the following questions:
   a. A worst-fit allocator always splits the largest free memory area while making an allocation. A free list contains three memory areas of sizes 6 KB, 15 KB and 12 KB. The next four memory requests are for 10 KB, 2 KB, 5 KB, and 14 KB of memory. The only placement strategy that would be able to accommodate all four processes is
      i. First-fit,
      ii. best-fit,
      iii. worst-fit,
      iv. next-fit.
   b. Three processes requiring 150 KB, 100 KB, and 300 KB of memory are in operation in an OS employing a paging system with a page size of 2 KB. The maximum internal memory fragmentation due to memory allocation to the three processes is
      i. Approximately 2 KB
      ii. Approximately 6 KB
      iii. 275 KB
      iv. None of (i)–(iii)
   c. A reentrant program is one that
      i. Calls itself recursively
      ii. Can have several copies in memory that can be used by different users
      iii. Can have a single copy in memory that is executed by many users Concurrently
2. Consider a logical address space of 64 pages of 1,024 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?
3. Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (ill order), how would the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Which algorithm makes the most efficient use of memory?
4. An OS has 110 MB available for user processes. The maximum memory requirement of a process for its own code and data is 20 MB, while the average memory requirement of a process is 10 MB. If the OS uses contiguous memory allocation and does not know sizes of individual processes, what is the average internal and external fragmentation?
5. A page-replacement algorithm should minimize the number of page faults. We can achieve this minimization by distributing heavily used pages evenly over all of memory, rather than having them compete for a small number of page frames. We can associate with each page frame a counter of the number of pages associated with that frame. Then, to replace a page, we can search for the page frame with the smallest counter.
   a. Define a page-replacement algorithm using this basic idea. Specifically address these problems:
      i. What is the initial value of the counters?

ii. When are counters increased?
iii. When are counters decreased?
iv. How is the page to be replaced selected?

b. How many page faults occur for your algorithm for the following reference string with four page frames?

1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2.

c. What is the minimum number of page faults for an optimal page replacement strategy for the reference string in part b with four page frames?

6. 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, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.

- LRU replacement
- FIFO replacement
- Optimal replacement

7. Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of the CPU and the paging disk. The results are one of the following alternatives. For each case, what is happening? Can the degree of multiprogramming be increased to increase the CPU utilization? Is the paging helping?

a. CPU utilization 13 percent; disk utilization 97 percent
b. CPU utilization 87 percent; disk utilization 3 percent
c. CPU utilization 13 percent; disk utilization 3 percent