Memory management

Deadline: 5/31/2011 (Due in class)

1. Consider the following process for generating binaries. A compiler is used to generate the object code for individual modules, and a linkage editor is used to combine multiple object modules into a single program binary. How does the linkage editor change the bindings of instructions and data to memory addresses? What information needs to be passed from the compiler to the linkage editor to facilitate the memory-binding tasks of the linkage editor?

2. Given five memory partitions of 100 KB, 500KB, 200 KB, 300 KB, and 600 KB (in order), how would the first-fit, best-fit and worst-fit algorithms place processes of size 212 KB, 417KB, 112KB and 426KB (in order)? Which algorithm makes the most efficient use of memory?

3. Most systems allow a program to allocate more memory to its address space during execution. Allocation of data in the heap segments of program is an example of such allocated memory. What is required to support dynamic memory allocation in the following schemes?

- Contiguous memory allocation
- Pure segmentation
- Pure paging

4. Consider a paging system with the page table stored in memory.

- If a memory reference takes 200 nano seconds, how long does a pages reference take?
- If we add TLBs, and assume that 75% of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page in the TLBs takes zero time, if the entry is there.)

5. Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses of the following logical addresses?

- 0, 430
- 1, 10
- 2, 500
- 3, 400
- 4, 112

6. Consider a demand-paging system, with the following time-measured utilizations:

- CPU Utilizations: 20%

- Paging disk: 97.7%
- Othe I/O devices: 5%

For each of the following, say whether it will (or is likely to) improve CPU utilization:

- Install a faster CPU
- Install a bigger paging disk
- Increase the degree of multiprogramming
- Decrease the degree of multi-programming
- Install more main memory
- Install a faster hard disk or multiple controllers with multiple hard disks
- Add prepaging to the page-fetch algorithm
- Increase the page size.

7. 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.

1. Define a page-replacement algorithm using this basic idea. Specifically address these problems:
   (a) What is the initial value of the counters?
   (b) When are counters increased?
   (c) When are counters decreased?
   (d) How is the page to be replaced selected?
2. How many page faults occur for your algorithm for the following reference string with 4 page frames?

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

3. What is the minimum number of page faults for an optimal page-replacement strategy for the reference string in part b with 4 page frames.