

## Operation System Midterm Exam

### Fall, 2017

1. Answer TRUE or FALSE in the following questions, and **justify your answers**. (3 points each)
  - a) API must be implemented by system call.
  - b) All I/O instructions must be privilege instructions.
  - c) When CPU utilization is low, OS should always launch more processes to increase the degree of multi-programming.
  - d) As long as a process is not using CPU, OS can immediately swap the process to backing store.
  - e) We can use **dynamic loading** approach to prevent processes from loading the same library into memory.
  - f) A smaller page size leads to smaller page tables.
  - g) The threads created through thread library can always run concurrently on multiple CPU cores.
  - h) The total memory allocation size from processes must be limited by the size of physical memory.
  - i) A **user-level** process cannot modify its own page table entries.
  - j) It's programmers' responsibility to prevent and handle page fault.
  - k) When pure demand paging scheme is used, the first memory reference to a page always causes page fault.
2. Microkernel OS implements its components in user space. What is the advantage and disadvantage of this approach? Justify your answer. (6 points)
3. Give 3 events that will insert a process into ready queue. (6 points)
4. Answer the following questions about Belady's anomaly. (7 points total)
  - a) What is Belady's anomaly? (2 points)
  - b) Why most of Oss prefer to use a page replacement policy without Belady's anomaly? (2 points)
  - c) Explain why LRU algorithm doesn't have Belady's anomaly. (3 points)
5. Answer the following questions about page table. (16 points total)
  - a) Why it becomes an issue if the page table cannot fit into the size of a page? (3 points)
  - b) Use an example to illustrate how hashed page table works. (3 points)

- c) Use an example to illustrate how inverted page table works. (3 points)
  - d) Explain why hashed page table can have better memory access performance than hierarchical page table? (3 points)
  - e) What is the advantage and disadvantage if using inverted page table? (4 points)
6. Answer following questions about threading. (8 points total)
- a) What are the differences between kernel threads and user threads? (4 points)
  - b) What is the main limitation of many-to-one model? (2 points)
  - c) What is the main limitation of one-to-one model? (2 points)
7. Answer following questions about memory accessing. (8 points total)
- a) Explain why supporting **runtime address binding (logical address)** and **virtual memory** can slow down memory access time. (4 points)
  - b) Explain why program locality can reduce the performance impact from these two memory management mechanisms, respectively. (4 points)
8. Answer following questions about TLB(Translation Lookaside Buffer). (6 points total)
- a) Why TLB must be flushed after context switch? (3 points)
  - b) Assume TLB access time is **10 ns** and memory access time is **100 ns**. What is the EMAT(effective memory access time) when a computer using **two-level paging** scheme has 80% of TLB hit rate? (3 points)
9. Consider a byte addressable computer using a one-level paging scheme with segmentation. The logical address has 12 bits. The physical address has 16 bits. Page size is 1 KB. Answer the following questions. (10 points)
- a) What is the size of physical memory? (2 points)
  - b) How many entries are in the page table? (2 points)
  - c) What is the maximum memory size of a process? (2 points)

10. If logical address 1A5 (in hexadecimal digits) translates to physical address 87A5 (in hexadecimal digits) using the segment table below, what is the linear address? (12 points)

Entry	Segment base address (in hexadecimal digits)
0	4600
1	1820
2	0002
3	AA01