

Ceng 328 Operating Systems
Final
May 31, 2011 14.00–16.00
Good Luck!

You are allowed to use only CALCULATOR.

No any other electronic equipment is allowed.

Answer all the questions.

1. (50 pts) Choose only **five** questions.
 - i Compare the synchronous and asynchronous I/O requests and discuss programming aspects.
 - ii Ordinarily the *exec()* system call follows the *fork()*. Explain what would happen if a programmer were to inadvertently place the call to *exec()* before the call to *fork()*.
 - iii Explain the main differences between a short-term and long-term scheduler.
 - iv Processes (or threads) can be in one of three states (blocked, running, ready). For each of the following three examples, write down which state the process (or thread) is in:
 - i Waiting in Domain Read() for a message from some other process to arrive.
 - ii Spin-waiting for a variable *x* to become non-zero.
 - iii Having just completed an I/O, waiting to get scheduled again on the CPU.
 - v What are the three general ways that a deadlock can be handled?
 - vi Describe a wait-for graph and how it detects deadlock.
 - vii What is external fragmentation? When does it occur?
 - viii What is thrashing in virtual memory concept? Explain in detail.
 - ix Explain the basic method for implementing paging.
 - x Where on a disk should the disk directories be physically located? What is the essential difference between a block special file and a character special file? Explain what hard and symbolic links are.
 - xi If you were creating an operating system to handle files, what would be the six basic file operations that you should implement?

2. (10 pts) Consider the following set of processes, with the length of the CPU-burst time given in milliseconds. Arrival time is the time at which the process is added to the ready queue.
 - i Draw appropriate charts illustrating the execution of these processes using FCFS, SJF non-preemptive (SJF-non), and SJF preemptive (SJF-pre).
 - ii Calculate the wait (W) and turnaround (T) times of each process for each strategy. Also calculate the average wait and turnaround times under each strategy.
 - iii Discuss which strategy is the best according to the table.

Process	Burst Time	Arrival Time	FCFS W Time	FCFS T Time	SJF-non W Time	SJF-non T Time	SJF-pre W Time	SJF-pre T Time
P1	10	0						
P2	5	0						
P3	3	0						
P4	4	0						
P5	2	4						
P6	1	6						
		Average						

3. (10 pts) Describe four ways to **prevent** deadlock by attacking the conditions required for deadlock. Is it possible to attack to these conditions and prevent deadlock?
4. (10 pts) Why caches are useful? What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?
5. (10 pts) What causes a page fault? What actions may be taken by the OS when servicing a page fault?
6. (10 pts) What is the context switch time, associated with swapping, if a disk drive with a transfer rate of 2 MB/s is used to swap out part of a program that is 200 KB in size? Assume that no seeks are necessary and that the average latency is 15 ms. The time should reflect only the amount of time necessary to swap out the process.
7. (10 pts) Explain the UNIX index node (inode) structure in detail.