Santa Clara University
Department of Computer Engineering
Advanced Operating Systems (COEN383, Section-1)
Midterm (30 points) Answer all Question
Time: 60 minutes

Name: Benita Rego

1. Multiple jobs can run in parallel and finish faster than if they had run sequentially. Suppose that two jobs, each needing 20 minutes of CPU time, start simultaneously. How long will the last one take to complete if they run sequentially? How long if they run in parallel? Assume 50% I/O wait – 6pts?

Answer: 50% I/O wait time signifies that a process is not running (i.e., the CPU is idle) for 50% of the time that it needs from the CPU to finish (its execution). As a result, CPU Utilization equals /whereas 50% I/O time indicates that it will take 50% of the overall execution time (10 minutes) to do its I/O.
50% = 50/100 = 0.5
Thus, it will take 20 minutes to do a task that uses 10 minutes of CPU time, where CPU time required by process/CPU usage is 10/CPU utilization is 10/0.5.
When two operations are carried out consecutively (one after the other), the total time needed is equal to 10/0.5 + 10/0.5 = 20 + 20 = 40 minutes.
Since two processes are running simultaneously, the formula becomes
The approximate CPU utilization is = {1 - (IO wait)^# of Processes}
=1 - (0.5)^2 = 1 - 0.25 = 0.75 in the case of parallel execution.
Now that 1 process is using 0.75/2 = 0.375 of the CPU, the necessary time will be equal to CPU time required by process/CPU usage, which is 10/0.375 = 26.67 minutes.
Since the two processes are operating simultaneously, the time needed by each will equal the time needed by both processes combined, or 26.67 minutes.

2. You are given the following data about a virtual memory system [6 pts]:
   a) The TLB can hold 1024 entries and can be accessed in 1 clock cycle (1 nsec).
   b) A Page table entry can be found in total of 100 clock cycles or 100 nsec,
   c) The total average page fault time is 6 msec
   If page references are handled by the TLB 99% of the time, and only 0.01% lead to page fault, what is the effective address-translation time?

Answer: By temporarily moving data from random access memory (RAM) to disk storage, virtual memory is a memory management feature of an operating system that enables a computer to make up for physical memory shortages.
Address Translation: Also called the Base and bound approach. Each CPU will require two hardware registers: the base register and the bounds register.This base-and-bounds combination will let us set the address space anywhere we wish in physical memory while yet limiting the process's ability to access addresses outside of its own address space. Each application is written and built in this configuration as though it were loaded at address 0.
By temporarily moving data from random access memory (RAM) to disk storage, virtual memory is a memory management feature of an operating system that enables a computer to make up for physical memory shortages.
Coming to the problem,
The likelihood of a hit for the TLB is 0.99, for the page table it is 0.0099, and for a page fault it is 0.0001. (i.e., only 1 in 10,000 references will cause a page fault).
Thus, 0.99 1 + 0.099 100 + 0.0001 6 106 602 clock cycles represent the effective address translation time in nsec. Even though page faults only occur once per 10,000 references, the effective address translation time is fairly large since the page replacement time dominates it.

3. What would happen if the bitmap or free list in memory containing the information about the free disk blocks was completely lost due to a crash? Is there any way to recover from this disaster, or it is over to the disk? Discuss your answer for UNIX file system [6 pts]

Answer: Disk utilities and recovery methods can fix this disaster. Scanning is an option for UNIX. Search for free entries in the FAT file by scanning it. UNIX and the FAT file system are not affected when a crash occurs. Disk utilities and recovery methods can fix it. Making a list of all the blocks in all the files and using the complement as the new free list is the recovery algorithm. This may be accomplished in UNIX by scanning every i-node. Because there is no free list in the FAT file system, the issue cannot arise. But even if there were, recovering it would only need a quick search of the FAT for empty entries.

4. Explain how an OS can facilitate installation of new device without any need for recompiling the OS? [6 pts]

Answer: Each table entry is a C struct that contains pointers to the methods for opening, closing, reading, and writing as well as a few other items from the device. The table is indexed by the device number. A new entry must be established in this table and the pointers must be filled in, frequently to the newly loaded device driver, in order to install a new device. Device tables, which are included in Linux operating systems, are indexed in accordance with the number of devices. Each device is represented in the table by a device data structure, and there is a reference to every function of the device for each item in the database. These pointers keep track of the entry for read, write, open, close, etc., allowing for the updating, adding, and removing of entries. A new entry is produced in the table each time a new device is added and set up on the operating system. After that, points are filled in for the new device's drivers.

5. A disk rotates at 7200 RPM. It has 500 sectors of 512 bytes around the outer cylinder. How long does it take to read a sector (only consider the data transfer component of the IO)? [6 pts]

Answer: At 7200 RPM, a revolution completes in around 8.33 milliseconds since there are 120 rotations every second.
(8.33 * 10^6)/500 = 16.67 seconds.
Thus, it yields a sector time of approximately 16.67 seconds when divided by 500.