# Operating Systems
## Major Exam
## COL 331, 633, and ELL 405

**100 marks – 9 questions**

**Answer all your questions very briefly. [Just the main points]**

**EASY**

1. Answer the following: [1-2 sentences each]
   a) What is the difference between a virus and a worm?
   b) What is the difference between the contents of a directory, and the contents of a file?
   c) Define the following: a Trojan, and a key logger.
   d) Is NFS a stateful file system? Justify your answer.
   e) What is a type 0 hypervisor?

   [10 marks]

2. Describe the differences, and the pros and cons of memory mapped I/O, and I/O mapped I/O. [8 sentences max.] [10 marks]

3. Describe the trap and emulate method. How does it work for interrupts, privileged instructions, and system calls? [8 sentences max.] [12 marks]

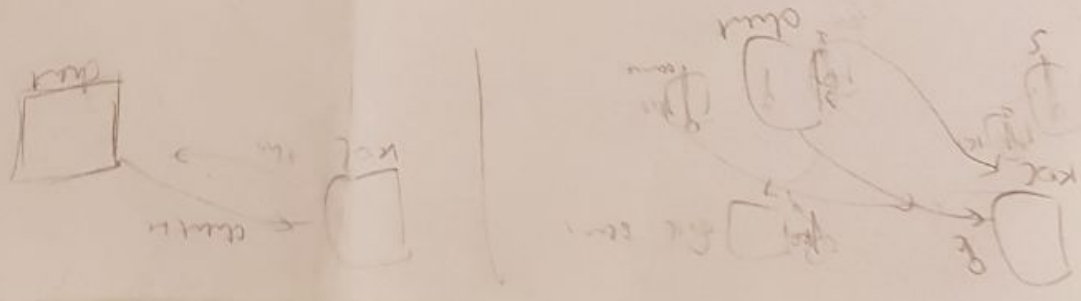
***MEDIUM***

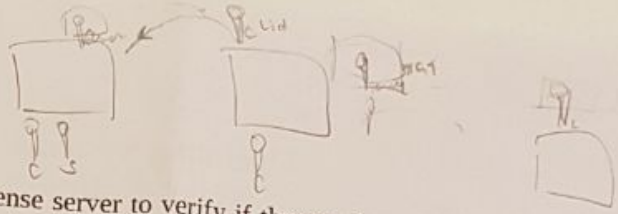
4. Describe the operation of a dynamic loading library (DLL). Recall that a DLL is a set of functions that are originally not a part of the program. The DLL is loaded at runtime. Subsequently, the code in the DLL needs to be made accessible to the program. Discuss the implications of using a DLL in terms of the file system, the memory management system, and the program itself. The answer should at least address the following issues: [15 sentences maximum]

   a) What happens if multiple programs need to use the same DLL concurrently?
   b) How do we manage two versions of the same DLL?
   c) What if the location of the DLL changes across two versions of the same operating system? Will programs stop working?
   d) How can DLLs have their global and static variables? If the same DLL is being used concurrently, wouldn't this cause a problem? [12 marks]

5. Answer the following questions: [2 sentences each max.] (3 + 3 + 4)
   a) Assume a system has practically an infinite amount of physical memory (much more than what all of your processes need). Would we still need virtual memory? Justify your answer.
   b) We often use bit vectors to store the list of free blocks in file systems. Can we optimize the bit vectors, and reduce the amount of storage?
   c) Describe the pros and cons of stream and block ciphers. [10 marks]

6. Describe the following concepts: [Maximum 6 sentences for each sub-part]
   a) Top halves and bottom halves of interrupt handlers in Linux.
   b) The NTFS file system. [10 marks]





**HARD**

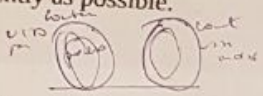
7. Most proprietary software use a license server to verify if the user has sufficient credentials to run the software. Think of a "license server" as an external server. The client sends its id, and IP address (cannot be spoofed), along with some more information. After several rounds of communication, the server sends a token that the client can use to run the application only once. The next time a fresh token is required. Design a cryptographic protocol that is immune to changing the system time on the client machine, replay attacks, and man-in-the-middle attacks. Assume that the binary of the program cannot be changed.
[Recall all the concepts we have learnt: public-private keys, one-way hashes, timestamps, random numbers, ciphers, Kerberos] [12 marks]

8. We want to implement containers in an operating system. A container is a mini virtual machine. Here are the features that a container should have:

a) Every container has its own set of user ids and passwords.
b) We can launch a process inside a container. It will not be visible to processes in other containers. At any point of time, processes from multiple containers might be running. Note that a container is **not** an operating system, nor a VMM. It is just a thin layer on top of the OS.
c) Initially, every process sees the native file system of the underlying OS. However, the moment, we change a file, a fresh copy is created for that container. For example, if we create a new version of */etc/passwd* in a container, then only that container will see the updated version of this file. This change will not be visible outside the container.
d) It should be possible to restrict the privileges of processes in a container. For example, they might not be able to access the network, or certain sectors on the disk.

Suggest a mechanism to implement all of these as efficiently as possible. [12 marks]

**RESEARCH [Fit to be a Ph.D Student]**



9. Let us design an operating system that supports record and replay. We first run the operating system in *record* mode, where it executes a host of applications that interact with I/O devices, the hard disk, and the network. A small module inside the operating system records all the events of *interest*. Let us call this the *record phase*.

After the record phase terminates, later on, we can run a *replay phase*. In this case, we shall run the operating system and all the constituent processes exactly **the same way** as they were running in the record phase. The OS and all the processes will show exactly the same behavior, and also produce exactly the same outputs in the same order. To an outsider both the executions will be *indistinguishable*. Such systems are typically used for debugging and testing, where it is necessary to exactly reproduce the execution of an entire system.

Your answer should at least address the following points:
1. What do we do about the time? It is clear that we have to use some notion of a logical time in the replay phase.
2. How do we deliver I/O messages from the network or hard disk, and interrupts with exactly the same content, and exactly at the same times?
3. What about non determinism in the memory system such as TLB misses, and page faults?
4. How do we handle inherently non-deterministic instructions such as *read_current_time* and *generate_random_number*? [12 marks]