Student Name:

CSCI U511 – Operating Systems

Homework-1Key, Weight: 50 points

Due on Tuesday, Sept. 3, 2019 at the beginning of the lecture (Hard Copy)
Note: You need to include your calculation and details to receive full credit!

Q1. [10 points] How would you design a system to run an entire operating systems as an application on top of another operating systems?

ANS: The sue of virtual machine would help to run an entire operating systems as an application on top of another operating system. The host operating system creates an abstract virtual machine that is identical to the hardware the host operating system is running on. The guest operating system then runs in the virtual machine as if it were running on the real hardware.

Q2. [10 points] How should operating system support communication between applications? Explain your reasoning.

ANS:

There an operating systems communicate between applications in three ways: producer-consumer-model, client-server-model, and file-system. In producer-consumer-model, the communication is one way where the producer writes and the consumer reads. The client-server-model allows two way communication between processes. The file system allows for communication that can be separated in time.

Q3. [10 points] Distinguish between the client–server and peer-to-peer models of distributed systems.

ANS: The client-server model firmly distinguishes the roles of the client and server. Under this model, the client requests services that are provided by the server. The peer-to-peer model doesn't have such strict roles. In fact, all nodes in the system are considered peers and thus may act as either clients or servers—or both. A node may request a service from another peer, or the node may in fact provide such a service to other peers in the system. For example, let's consider a system of nodes that share cooking recipes. Under the client-server model, all recipes are stored with the server. If a client wishes to access a recipe, it must request the recipe from the specified server. Using the peer-to-peer model, a peer node could ask other peer nodes for the specified recipe. The node (or perhaps nodes) with the requested recipe could provide it to the requesting node. Notice how each peer may act as both a client (it may request recipes) and as a server (it may provide recipes).

Student Name:

Q4. [10 points] What is the difference between a process and a program? What hardware enables the operating systems to efficiently implement the process abstraction?

ANS: A process is an instance of a program with restricted rights. A program is a set of instruction that is waiting to be executed by the user. It cannot do anything until it is called. The hardware, like, memory, input-output devices, help to create abstraction as the OS can group them together to create high level abstractions for easier use.

Q5. [10 points] Explain the steps that an operating system goes through when the CPU receives an interrupt.

ANS: After the hardware saves the old stack pointer, the old program counter, and other registers, it switches to the kernel stack at the beginning of the (assembly language) interrupt handler stub. The interrupt handler stub must then save (and restore) the remaining register state, set up (tear down) the interrupt handler stack frame, and call into (return from) the interrupt handler, and execute the handler. The handler itself is device specific, but in general it will at least read control registers off the device to determine what operation has completed.