**Class Activities CS 446**

**Chapter 2**

**Question 1:** What is the purpose of system calls?

Answer: System calls allow user-level processes to request services of the operating system.

**Question 2:** What are the five major activities of an operating system with regard to

process management?

**Answer:**

The five major activities are:

1. The creation and deletion of both user and system processes
2. The suspension and resumption of processes
3. The provision of mechanisms for process synchronization
4. The provision of mechanisms for process communication

e. The provision of mechanisms for deadlock handling

**Question 3:** What are the three major activities of an operating system with regard to memory management?

**Answer:**

The three major activities are:

1. Keep track of which parts of memory are currently being used and by whom.
2. Decide which processes are to be loaded into memory when memory space becomes available.

c. Allocate and deallocate memory space as needed.

**Question 4:** What are the three major activities of an operating system with regard to secondary-storage management?

**Answer:**

The three major activities are:

1. Free-space management.
2. Storage allocation.
3. Disk scheduling.

**Question 5:** What is the purpose of the command interpreter? Why is it usually

separate from the kernel?

**Answer:**

It reads commands from the user or from a file of commands and executes them, usually by turning them into one or more system calls. It is usually not part of the kernel since the command interpreter is subject to changes.

**Question 6:** What is the purpose of system programs?

**Answer:**

System programs can be thought of as bundles of useful system calls. They provide basic functionality to users so that users do not need to write their own programs to solve common problems.

**Question 7:** What is the main advantage of the layered approach to system design? What are the disadvantages of using the layered approach?

**Answer:**

As in all cases of modular design, designing an operating system in a modular way has several advantages. The system is easier to debug and modify because changes affect only limited sections of the system rather than touching all sections of the operating system. Information is kept only where it is needed and is accessible only within a defined and restricted area, so any bugs affecting that data must be limited to a specific module or layer.

**Question 8:** List five services provided by an operating system, and explain how each creates convenience for users. In which cases would it be impossible for user-level programs to provide these services? Explain your answer.

**Answer:**

The five services are:

1. **Program execution**. The operating system loads the contents (or sections) of a file into memory and begins its execution. A user-level program could not be trusted to properly allocate CPU time.
2. **I/O operations**. Disks, tapes, serial lines, and other devices must be communicated with at a very low level. The user need only specify the device and the operation to perform on it, while the system converts that request into device- or controller-specific commands. User-level programs cannot be trusted to access only devices they should have access to and to access them only when they are otherwise unused.
3. **File-system manipulation.** There are many details in file creation, deletion, allocation, and naming that users should not have to perform. Blocks of disk space are used by files and must be tracked. Deleting a file requires removing the name file information and freeing the allocated blocks. Protections must also be checked to assure proper file access. User programs could neither ensure adherence to protection methods nor be trusted to allocate only free blocks and deallocate blocks on file deletion.
4. **Communications.** Message passing between systems requires messages to be turned into packets of information, sent to the network controller, transmitted across a communications medium, and reassembled by the destination system. Packet ordering and data correction must take place. Again, user programs might not coordinate access to the network device, or they might receive packets destined for other processes.
5. **Error detection**. Error detection occurs at both the hardware and software levels. At the hardware level, all data transfers must be inspected to ensure that data have not been corrupted in transit. All data on media must be checked to be sure they have not changed since they were written to the media. At the software level, media must be checked for data consistency; for instance, whether the number of allocated and unallocated blocks of storage match the total number on the device. There, errors are frequently process- independent (for instance, the corruption of data on a disk), so there must be a global program (the operating system) that handles all types of errors. Also, by having errors processed by the operating system, processes need not contain code to catch and correct all the errors possible on a system.

**Question 9:** The services and functions provided by an operating system can be divided into two main categories. Briefly describe the two categories, and discuss how they differ.

**Answer:**

One class of services provided by an operating system is to enforce protection between different processes running concurrently in the system. Processes are allowed to access only those memory locations that are associated with their address spaces. Also, processes are not allowed to corrupt files associated with other users. A process is also not allowed to access devices directly without operating system intervention. The second class of services provided by an operating system is to provide new functionality that is not supported directly by the underlying hardware. Virtual memory and file systems are two such examples of new services provided by an operating system.

**Question 10:** Describe three general methods for passing parameters to the operating

system.

**Answer:**

1. Pass parameters in registers
2. Registers pass starting addresses of blocks of parameters
3. Parameters can be placed, or *pushed,* onto the *stack* by the program, and *popped* off the stack by the operating system

**Question 11:** Would it be possible for the user to develop a new command interpreter using the system-call interface provided by the operating system?

**Answer:**

An user should be able to develop a new command interpreter using the system-call interface provided by the operating system. The command interpreter allows an user to create and manage processes and also determine ways by which they communicate (such as through pipes and files). As all of this functionality could be accessed by an user-level program using the system calls, it should be possible for the user to develop a new command-line interpreter.

**Question 12:** What are the two models of interprocess communication? What are the

strengths and weaknesses of the two approaches?

**Answer:**

The two models of interprocess communication are message-passing model and the shared-memory model. Message passing is useful for exchanging smaller amounts of data, because no conflicts need be avoided. It is also easier to implement than is shared memory for intercomputer communication. Shared memory allows maximum speed and convenience of communication, since it can be done at memory transfer speeds when it takes place within a computer. However, this method compromises on protection and synchronization between the processes sharing memory.

**Question 13:** It is sometimes difficult to achieve a layered approach if two components of the operating system are dependent on each other. Identify a scenario in which it is unclear how to layer two system components that require tight coupling of their functionalities.

**Answer:**

The virtual memory subsystem and the storage subsystem are typically tightly coupled and requires careful design in a layered system due to the following interactions. Many systems allow files to be mapped into the virtual memory space of an executing process. On the other hand, the virtual memory subsystem typically uses the storage system to provide the backing store for pages that do not currently reside in memory. Also, updates to the file system are sometimes buffered in physical memory before it is flushed to disk, thereby requiring careful coordination of the usage of memory between the virtual memory subsystem and the file system.

**Question 14:** What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

**Answer:**

Benefits typically include the following: (a) adding a new service does not require modifying the kernel, (b) it is more secure as more operations are done in user mode than in kernel mode, and (c) a simpler kernel design and functionality typically results in a more reliable operating system. User programs and system services interact in a microkernel architecture by using interprocess communication mechanisms such as messaging. These messages are conveyed by the operating system. The primary disadvantages of the microkernel architecture are the overheads associated with interprocess communication and the frequent use of the operating system’s messaging functions in order to enable the user process and the system service to interact with each other.

**Question 15:** What are the advantages of using loadable kernel modules?

**Answer:**

It is difficult to predict what features an operating system will need when it is being designed. The advantage of using loadable kernel modules is that functionality can be added to and removed from the kernel while it is running. There is no need to either recompile or reboot the kernel.