King Saud University
Department of Computer Science
CSC227: Operating Systems
Tutorial No. 2

Q 1) What is the purpose of the command interpreter? Why is it usually separate from the kernel?
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.

Q 2) What is the purpose of system calls?
Ans: System calls allow user-level processes to request services of the operating system.

Q 3) What are the main advantages of the microkernel approach to system design?
Ans: 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.

Q 4) What are the typical components of a microkernel?
Ans: Typically, microkernels provide minimal process and memory management, in addition to a communication facility.

Q 5) What system calls have to be executed by a command interpreter or shell in order to start a new process?
Ans: In UNIX systems, a fork system call followed by an exec system call need to be performed to start a new process. The fork call clones the currently executing process, while the exec call overlays a new process based on a different executable over the calling process.

Q 6) What is the purpose of system programs?
Ans: 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.

Q 7) What is the main advantage of the layered approach to system design?
Ans: 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.

Q 8) What system calls have to be executed by a command interpreter or shell in order to start a new process?
Ans:
In UNIX systems, a fork system call followed by an exec system call need to be performed to start a new process. The fork call clones the currently executing process, while the exec call overlays a new process based on a different executable over the calling process.

Q 9) Why do some systems store the operating system in firmware, while others store it on disk?
Ans:
For certain devices, such as handheld PDAs and cellular telephones, a disk with a file system may be not be available for the device. In this situation, the operating system must be stored in firmware.

Q 10) How could a system be designed to allow a choice of operating systems from which to boot? What would the bootstrap program need to do?
Ans:
Consider a system that would like to run both Windows XP and three different distributions of Linux (e.g., RedHat, Debian, and Mandrake). Each operating system will be stored on disk. During system boot-up, a special program (which we will call the boot manager) will determine which operating system to boot into. This means that rather initially booting to an operating system, the boot manager will first run during system startup. It is this boot manager that is responsible for determining which system to boot into. Typically boot managers must be stored at certain locations of the hard disk to be recognized during system startup. Boot managers often provide the user with a selection of systems to boot into; boot managers are also typically designed to boot into a default operating system if no choice is selected by the user.

Q 11) How does the distinction between monitor mode and user mode function as a rudimentary (basic) form of protection (security) system?
Ans: By establishing a set of privileged instructions that can be executed only when in the monitor mode, the operating system is assured of controlling the entire system at all times.

Q 12) Which of the following instructions should be privileged?
a) Set value of timer.
b) Read the clock.
c) Clear memory.
d) Turn off interrupts.
e) Switch from user to monitor mode.
Answer: The following instructions should be privileged:
a) Set value of timer.

c) Clear memory.
d) Turn off interrupts.
e) Switch from user to monitor mode.

Q 13) Why is the separation of mechanism and policy desirable?
Ans: Mechanism and policy must be separate to ensure that systems are easy to modify. No two system installations are the same, so each installation may want to tune the operating system to suit its needs. With mechanism and policy separate, the policy may be changed at will while the mechanism stays unchanged. This arrangement provides a more flexible system.

Q 14) What are the two models of inter-process communication? What are the strengths and weaknesses of the two approaches?
Ans:
Message — passing model: In this, the communicating processes exchange messages with one another to transfer information. Messages can be exchanged between the processes either directly or indirectly through a common mail box. 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 inter computer communication. But the main disadvantage is it can handle only small amounts of data.

Shared—Memory model: In this, processes use shared memory creates and shared memory attaches system calls to create and gain access to regions of memory owned by other processes. Two or more processes can exchange information by reading and writing data in the shared areas. Shared memory allows maximum speed and convenience of communication, since it can be done at memory speeds when it takes place within a computer Problems exist, however, in the areas of protection and synchronization between the processes sharing memory.