King Saud University
Department of Computer Science
CSC227: Operating Systems
Tutorial – Chapter 1: Introduction

Q 1) What are the three main purposes of an operating system?

- To provide an environment for a computer user to execute programs on computer hardware in a convenient and efficient manner.
- Resource allocator. To allocate the resources of the computer as needed to solve the problem given. The allocation process should be as fair and efficient as possible.
- As a control program it serves two major functions: (1) supervision of the execution of user programs to prevent errors and improper use of the computer, and (2) management of the operation and control of I/O devices.

Q 2) What is the main advantage of multiprogramming?

Multiprogramming makes efficient use of the CPU by overlapping the demands for the CPU and its I/O devices from various users. It attempts to increase CPU utilization by always having something for the CPU to execute.

Q 3) What are the main differences between operating systems for mainframe computers and PCs?

The design goals of operating systems for those machines are quite different. PCs are inexpensive, so wasted resources like CPU cycles are inconsequential. Resources are wasted to improve usability and increase software user interface functionality. Mainframes are the opposite, so resource use is maximized, at the expense of ease of use.

Q 4) Definitions following terms:

a) Processor

The processor (also called Central Processing Unit, or CPU) is the part of the computer that actually works with the data and runs the programs.

b) Program

A sequence of instructions that a computer can interpret and execute.

c) Process

A process is a running instance of a program, including all variables and other state.

Q 5) How is I/O done in a computer?

To start an I/O operation, the device driver loads the appropriate registers within the device controller. The device controller, in turn examines the contents of these registers to determine what action to take (such as read). The controller starts the transfer of data from the device to its local buffer. Once the transfer of data is complete, the device controller informs the device driver via an interrupt that it has finished its operation. The device deriver then returns control to the operating system, possibly returning the data or pointer to the data if the operation was a read.

Q 6) What is DMA? How does it operate?

DMA stands for direct memory access. It is technique of handling data transfer between memory and peripheral (I/O) devices that bypasses the central processing unit. In this method the device controller transfers an entire block of data directly to or from its own buffer storage to memory. Only one interrupt is generated per block, to tell the device driver that the operation is completed, rather than the one interrupt per byte generated for low speed devices. While the device controller is performing these operations, the CPU is available to accomplish other work.

Q 7) Define the essential properties of the following types of operating systems:

a) Batch

Jobs with similar needs are batched together and run through the computer as a group by an operator or automatic job sequencer. Performance is increased by attempting to keep CPU and I/O devices busy at all times through buffering, off-line operation, spooling, and multiprogramming. Batch is good for executing large jobs that need little interaction; it can be submitted and picked up later.

b) Time sharing

Uses CPU scheduling and multiprogramming to provide economical interactive use of a system. The CPU switches rapidly from one user to another. Instead of having a job defined by spooled card images, each program reads its next control card from the terminal, and output is normally printed immediately to the screen.

c) Real time

Often used in a dedicated application. The system reads information from sensors and must respond within a fixed amount of time to ensure correct performance.

d) Distributed

Distributes computation among several physical processors. The processors do not share memory or a clock. Instead, each processor has its own local memory. They communicate with each other through various communication lines, such as a high-speed bus or telephone line.

e) Clustered

Like parallel systems, clustered systems gather together multiple CPUs to accomplish computational work. Clustered systems differ from parallel systems, however, in that they are composed of two or more individual systems coupled together.

f) Handheld

Operating systems are designed to provide an environment in which a user can easily interface with the computer to execute programs. Many varieties of handheld computers have come into fashion. These devices are mostly standalone, used singly by individual users. Some are connected to networks, either directly by wire or (more often) through wireless modems.

Q 8) What are the tradeoffs inherent in handheld computers?

- Limitation of size in memory. Usually in megabytes compared to gigabytes in computers.
- Processor speed is limited because of less power ability. the operating system has to be designed for low power consumption.
- Lack of physical space limits the I/O capability. The input method are small screen based keyboard or hand-writing recognition while small display screen for output.

Q 9) What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?

The main difficulty is keeping the operating system within the fixed time constraints of a real-time system. If the system does not complete a task in a certain time, it could cause a breakdown of the entire system it is running. Therefore, when writing an operating system for a real-time system, the writer must be sure that his scheduling schemes don't allow response time to exceed the time constraint.

Q 10) 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) Modify entries in device-status table
f) Switch from user to kernel mode
g) Access I/O device

a) Privileged. This instruction must be privileged, otherwise a process could reset the timer to its own advantage and dominate the CPU resources.
b) Not privileged. This instruction should not be privileged. There is no security effects of a process reading the clock value.
c) Privileged. This would wipe out the code and data of the OS and any other processes. User level processes should not be allowed to do this. Should be privileged.
d) Privileged. This must be a privileged instruction because turning off interrupts would disable many of the processor functions for all processes.
e) Privileged.
f) Not privileged. This is not a privileged instruction because switching to monitor mode would give the process operating system privileges.
g) Privileged. Reading/writing to a device will interfere with other processes (think some rogue process reading from the keyboard while passwords are being typed in...).

Q 11) In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation can result in various security problems.

a) What are two such problems?
b) Can we ensure the same degree of security in a time-shared machine as in a dedicated machine? Explain your answer.

a) Stealing or copying one's programs or data; using system resources (CPU, memory, disk space, peripherals) without proper accounting.

b) Probably not, since any protection scheme devised by humans can inevitably be broken by a human, and the more complex the scheme, the more difficult it is to feel confident of its correct implementation.

Q 12) The issue of resource utilization shows up in different forms in different types of operating systems. List what resources must be managed carefully in the following settings:

c) Mainframe or minicomputer systems
d) Workstations connected to servers
e) Handheld computers

a) Mainframes: memory and CPU resources, storage, network bandwidth.
b) Workstations: memory and CPU resources.
c) Handheld computers: power consumption, memory resources.

Q 13) Distinguish between the client–server and peer-to-peer models of distributed systems.
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.

Q 14) What is the purpose of interrupts? What are the differences between a trap and an interrupt? Can traps be generated intentionally by a user program? If so, for what purpose?
An interrupt is a hardware-generated change-of-flow within the system. An interrupt handler is summoned to deal with the cause of the interrupt; control is then returned to the interrupted context and instruction. A trap is a software-generated interrupt. An interrupt can be used to signal the completion of an I/O to obviate the need for device polling. A trap can be used to call operating system routines or to catch arithmetic errors.

Q 15) What network configuration would best suit the following environments?

a) An office: Local Area Network (LAN) connects network devices over a relatively short distance.
b) A university campus: LAN
c) A state: Metropolitan Area Network (MAN) - a network spanning a physical area larger than a LAN but smaller than a WAN, such as a city. A MAN is typically owned and operated by a single entity such as a government body or large corporation.
d) A nation: WAN - Wide Area Network (WAN) covers a large physical distance. The Internet is the largest WAN.

Q 16) What is the difference between:

a) mono-programming and multi-programming.

- In a mono-programming machine a single job occupies the system from start until end. In such a case CPU sits idle when there is a need of user action.
- A multi-programming machine increases CPU utilization by organizing jobs so that CPU always has something to execute.

b) mono-processor and multi-processor.

- In a mono-processor machine there is only one processor which is responsible for the execution.
- In a multi-processor machine there are more than one machine that may share the execution.

c) CPU-scheduler and job-scheduler.

- CPU scheduler assigns CPU to a process.
- Job scheduler selects job for execution.