Cheng | Midterm 1: Chapter 1 | Study Guide

Chapter 1: Introduction

OS Definition:

Since the OS looks different there is no universal definition, but we all know what it is meant to provide us:

A more common definition, and the one we usually follow, is that the operating system is the one program running at all times on the computer-usually called the kernel.

The operating system includes the always-running kernel, middleware frameworks that ease application development and provide features, and system programs that aid in managing the system while its running.

OS Goals:

The fundamental goal of computer systems is to execute programs and to make solving user problems easier. An operating system acts as an intermediary between the user of the computer and the computer hardware.

Idea of Abstraction:

The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner. It is a piece of special 'software' that provides services to support the applications to run, it manages the computer hardware. It is special because it acts as an intermediary translator.

The bridge is important because the users can be malicious and dumb. We need to have a mediated control that will provide a 'user friendly' interface to the hardware, because people can damage the hardware.

Resource Allocation

From the computer view, the OS is the resource allocator and acts as the manager of the resources. It will decide within conflicting requests for resources, OS decides how to allocate them to specific programs.

> *Different types of resources included CPU, memory, and I/O devices.*

Control Program

As a control program, it manages the execution of the user programs to prevent errors and improper use of the computer.

Computer System Structure: A computer system can be divided roughly into four components: the hardware, the operating system, the application programs, and a user.

Where the hardware- consists of the central processing unit, the memory, I/O devices, the application programs- consists of word processors, spreadsheets, compilers, and web browsers, the operating system- controls hardware and coordinates its use among application programs.

A modern computer is comprised of one or more CPUs and a number of device controllers connected through a common bus that provides access between components and shared memory.

Interrupts: This is how a controller informs the device driver that it has finished its operation.

Can you explain better the relationship between a device controller and device driver?

Interrupt Vector:

The basic function of an interrupt is that it transfers control to the appropriate interrupt service routine, where it will invoke a generic routine to examine interrupt information and then call the specific handler. But like imagine a gillion interrupts happening very frequently. Using a table of pointers is so much faster, you don't need the 'generic routine'.

These locations hold the addresses of the interrupt service routines for the various devices. This array, or interrupt vector, of addresses is then indexed by a unique number, given with the interrupt request, to provide the address of the interrupt service routine for the interrupting device.

Polling Vs. Vectored Interrupt System:

Polling:

Reads the status register over and over and over again until the busy bit is clear. This becomes inefficient when it is attempted repeatedly yet rarely finds a device ready for service.

"Are you ready? Are you ready? Are you ready?"

Vectored Interrupt System:

Arranges for hardware controller to notify the CPU when the device becomes ready for service.

Multiprocessor Architectures: (Symmetric vs. Asymmetric)

Primary advantage of a multiprocessor architecture is that by increasing the number of processors, more work done so you have increased throughput. But remember that the speed-up for having 7 processors is not 7, it is less than that because of the amount of overhead to switch between processors when doing a task.

Symmetric: Each CPU processor performs all task, including operating-system functions and user processes. Each CPU has it's own set of registers, but a physical memory is shared by a bus.

Benefit: Many processes can run at once, 7 processes for 7 CPUs.

Disadvantage: The CPU don't share certain data structures, so memory can't be shared dynamically, one CPU can be overloaded while one sits idle.

Asymmetric: All scheduling decisions, I/O processing and other system activities handled by a single master core.

Benefit: One core accesses the system data structures, reducing the need for data sharing.

Disadvantage: When master core becomes a bottleneck where overall system performance may be reduced.

Dual-mode: (User Mode vs. Kernel Mode)

Dual-mode operation allows OS to protect itself and other system components. A properly designed operating system must ensure that a malicious program can NOT cause other programs or the OS itself to execute incorrectly.

A bit, called the mode bit, is added to the hardware of the computer to indicate the current mode: kernel (0) or user (1). With the mode bit, we can distinguish between a task that is executed on behalf of the operating system and one that is executed on behalf of the user.

Mode bit provided by hardware. It provides the ability to distinguish when system is running user code or kernel code. Some instructions designated as privileged, only executable in kernel mode. System call changes mode to kernel, return from call resets it to user.

Note: At system boot time, the hardware starts in kernel mode. The OS loads and then starts the user applications in user mode. When the OS gains control of the computer, it is in kernel mode.

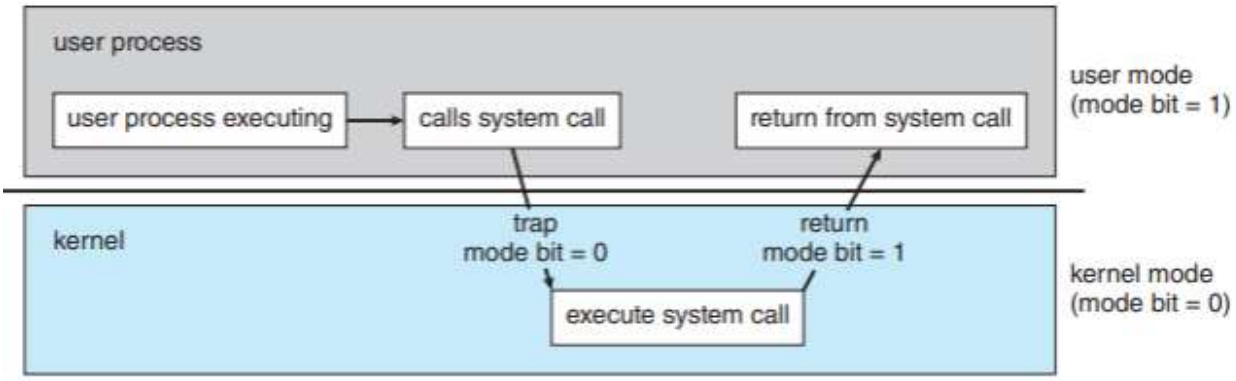

Figure 1.13 Transition from user to kernel mode.