Chapter 2

Computer Organisation

Lesson 7

Operating Systems

Objective —

- **Understand use of OS, multiprogramming, protection, virtual memory and privileged mode**

Outline

- **Operating System**
- Multiprogramming
- Protection
- Control of physical resources
- Privileged mode

The operating system

- Simply another program, one that knows about all the hardware in the computer, with one exception — it runs in privileged (or supervisor) mode,

Privileged Mode

- Allows it access to physical resources that user programs (tasks) cannot control and gives it the ability to start or stop the execution of user programs.

OS Responsibilities

- Managing the physical resources of the system,
- Loading and executing programs,
- Managing devices and network
- Controlling through *device drivers* the I/O devices .

Outline

- Operating System
- **Multiprogramming**
- Protection
- Control of physical resources
- Privileged mode

MULTIPROGRAMMING

- Most computer systems support multiprogramming (also called multitasking),a technique that allows the system to present the illusion that multiple programs are running on the computer simultaneously, even though the system may only have one processor.

MULTIPROGRAMMED SYSTEM

- User programs do not need to know which other programs are running on the system at the same time they are, or even how many other programs there are.

Multiuser computers

- Many multiprogrammed computer are also *multiuser* and allow more than one user to be logged in to the computer at once. Multiuser systems require that the operating system not only protect programs from accessing each other's data but prevent users from accessing data that is private to other users.

Multiprogrammed operating system

- It presents the illusion that multiple programs are running simultaneously by switching between programs very rapidly

Multiprogrammed operating system

- Each program is allowed to execute for a fixed amount of time, known as a *timeslice*.

Context Switch

- When a program's timeslice ends, the operating system stops it, removes it from the processor, and loads another program into the processor. This process is known as a *context switch.*

Context switch

- The operating system copies the contents of the currently running program's register file (sometimes called the program's context) into memory, and then copies the contents of the next program's register file out of memory and into the register file.

Context Switch

- Programs cannot tell that a context switch has been performed—to them, it looks like they have been continuously running on the processor

Context-switch Period

- 60 times per second, making tirneslices l/60th of a second,
- Lately systems have started to context-switch more frequently.

Fast Context Switch Time

- This has caused problems with programs that expect timeslices to be l/60th of a second and use that information to time events or determine performance.

EXAMPLE

- By context-switching 60 or more times per second, a computer can give each program an opportunity to execute sufficiently frequently that the system can prevent the illusion that a moderate number of programs are executing simulta-neously on the system.

EXAMPLE

- Obviously, as the number of programs on the system increases, this illusion breaks down—if the system is executing 120 programs, each program may only get a timeslice once every 2 seconds, which is enough of a delay for us to notice.

Increase of the running time in Multiprogramming

- Running time of applications increase
- The resources of the system are shared among all of the programs running on it.

Outline

- Operating System
- Multiprogramming
- **Protection**
- Control of physical resources
- Privileged mode

Main requirements of a multiprogrammed operating system

- One of the is that OS must provide *protection* between programs running on the computer.

Main requirements of a multiprogrammed operating system

- Result of any program running on a multiprogrammed computer must be the same as if the program was the only program running on the computer.

Operating system

- OS and the hardware provide *protection* for programs, preventing any program from accessing another program's data unless the two programs have specifically arranged to access each other's data.

Protection

- Programs must not be able to access other programs' data and must be confident that their data will not be modified by other programs. Similarly, programs must not be able to interfere with each other's use of the I/O subsystem

USE OF VIRTUAL MEMORY

- One technique that operating systems use to protect each program's data from other programs is *virtual memory*,

USE OF VIRTUAL MEMORY

- virtual memory allows each program to operate as if it were the only program running on the computer by *translating* memory addresses that the program references into the addresses used by the memory system. ,

Virtual memory system

- Ensures that two programs' addresses don't translate into the same address, programs can be written as if they were the only program running on the machine, since no program's memory references will access data from another program.

Outline

- Operating System
- Multiprogramming
- Protection
- **Control of physical resources**
- Privileged mode

CONTROL OF PHYSICAL RESOURCES

- Providing protection in a multiprogrammed or multiuser system requires that the operating system control the physical resources of the computer, including the processor, the memory, and the I/O devices.

Without Protection

- User programs could access any of the memory or other storage on the computer, gaining access to data that belongs to other programs or other users.

With Protection

- This also allows the operating system to prevent more than one program from accessing an I/O device, such as a printer, at one time

Outline

- Operating System
- Multiprogramming
- Protection
- Control of physical resources
- **Privileged mode**

Privileged mode

- Ensures that the operating system is the only program that can control the system physical resources,
- User programs execute in *user mode* (sometimes called unprivileged mode).

Context switches and memory allocations

- Certain tasks, such as accessing
- an I/O device, require that a program be in privileged mode. If a user-mode program tries to
- perform one of these tasks, the hardware prevents it from doing so and signals an error.

User-mode programs

- If want to do something that requires privilege mode, they send a request to the operating system, known as a *system call*, which asks the operating system to do the operation for them.

User System calls

- If the operation is something that the user program is allowed to do, the operating system performs the operation and returns the result to the user. Otherwise, it signals an error

User Interface

- Because OS controls the physical resources of the computer, the it is also responsible for the low-level user interface.

Example

- When a user presses a key or otherwise sends input to the computer, the operating system is responsible for determining which program should receive the input and sending the input value to that program.

Example

- When a program wants to display some information for the use such as printing a character on the monitor, it executes a system call to request that the operating system display the data.

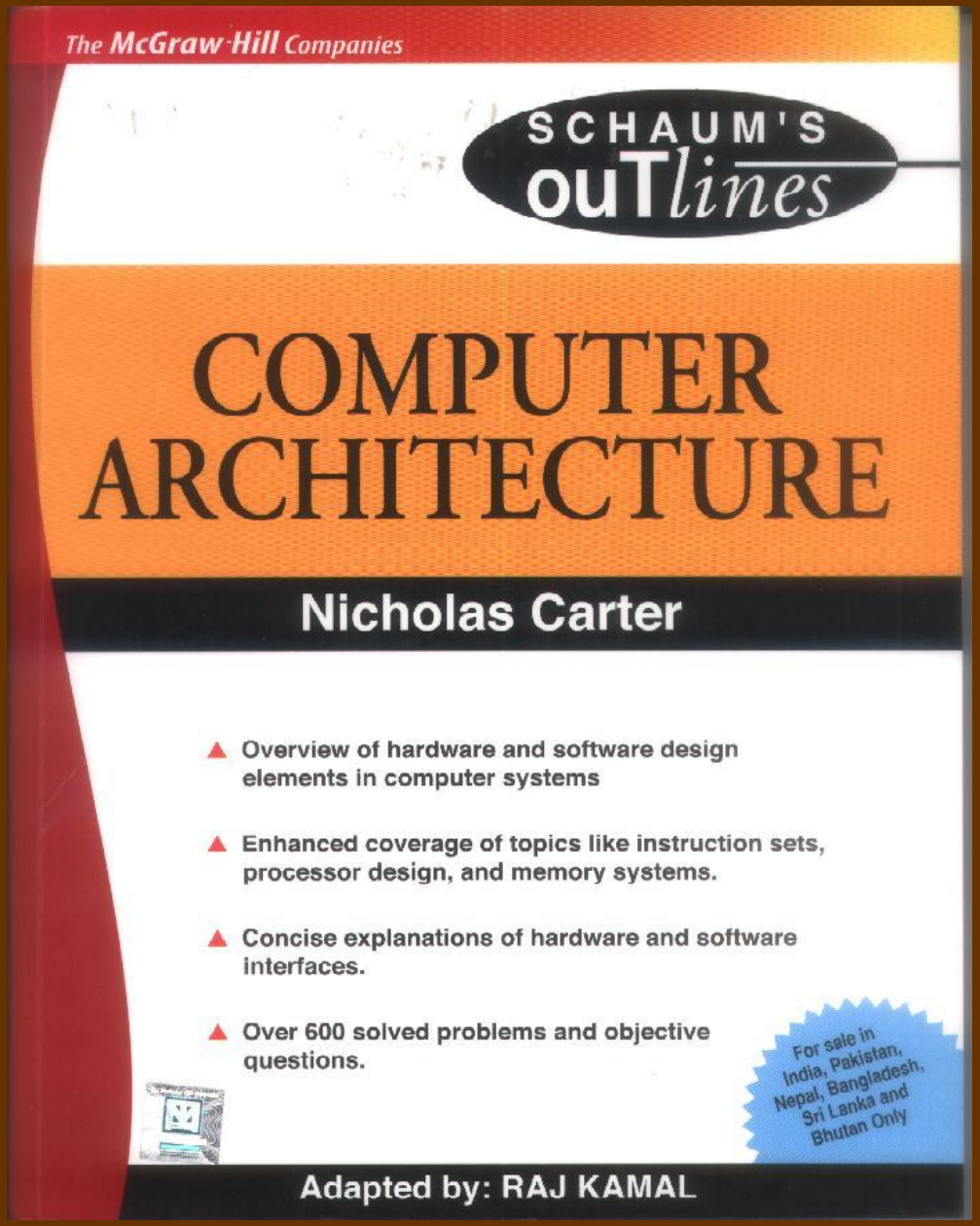
Summary

We learnt

- OS control of resources,
- multiprogramming,
- Context Switch
- protection,
- Time slice
- Privilege Mode and user mode

End of Lesson 7 on

Operating Systems


The McGraw-Hill Companies
SCHAUM'S
ouTlines
COMPUTER ARCHITECTURE
Nicholas Carter
Overview of hardware and software design elements in computer systems
Enhanced coverage of topics like instruction sets, processor design, and memory systems.
Concise explanations of hardware and software interfaces.
Over 600 solved problems and objective questions.
For sale in India, Pakistan, Nepal, Bangladesh, Sri Lanka and Bhutan Only
Adapted by: RAJ KAMAL

THANK YOU