SOLUTION

Institute of Business Administration
MIS & CS Department
Operating Systems, Fall Semester 2003
BCS IV
First Hourly Test
September 1, 2003

Time Allowed: One Hour **Total Marks: 100**

Instructions

a. Attempt all questions.
b. Maximum/Total Marks are 100.
c. Time allowed is 1 hour.
d. Do NOT write any thing on the Question Paper except your name. Provide your answers on the answer sheet provided for this purpose.

Question 1: Fill in the blanks *(40 Marks)*

1. SPOOL stands for Simultaneous Peripheral Operation OnLine.
2. JCL stands for Job Control Language.
3. Using multiprogramming to handle multiple interactive jobs is called timesharing.
4. The principle objective of batch multiprogramming is to maximize processor utilization.
5. A unit of activity characterized by a single sequential thread of execution, a current state, and an associated set of system resources is called a process.
6. Virtual Memory allows you to run programs bigger than the size of the system RAM.
7. In modern operating systems Process is a collection of one or more thread.
8. Linux/Solaris/AIX is an example of a modern UNIX system.
9. Interrupt driven I/O allows the processor to execute other instructions while an I/O operation is in progress.
10. According to the Principle of Locality, going down the memory hierarchy, the frequency of access of the memory by the processor decreases.

Question 2: *(20 Marks)*

Writing an operating system that can operate without interference from malicious or undebugged user programs requires some hardware assistance. Name three hardware aids for writing an operating system, and describe how they may be used together to protect the operating system. Your answer (including any figures) must not exceed two written pages.

Answer:

Four hardware aids for writing an operating system which can operate without interference from malicious or undebugged user programs are:

SOLUTION

1. Dual-Mode Operation
2. I/O Protection
3. Memory Protection
4. CPU Protection using Timers

Dual Mode Operation

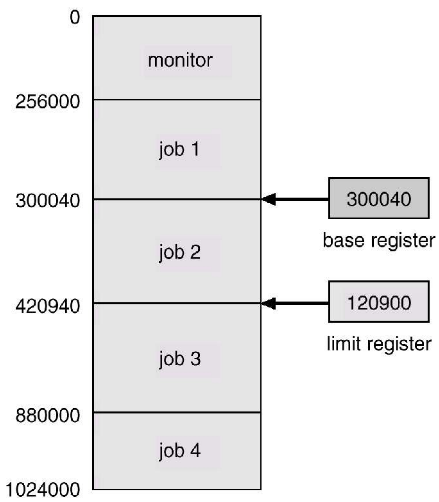
Dual-Mode of operation provides two separate modes: monitor/privileged mode and user mode. These modes are distinguished using a mode bit which is added to the CPU hardware. At system boot time, the hardware starts in monitor mode. The operating system is then loaded, and starts user processes in user mode. Whenever a trap or interrupt occurs, the hardware switches from user mode to monitor mode. The machine instructions which can cause harm such as I/O operations are executed in monitor mode only.

I/O Protection

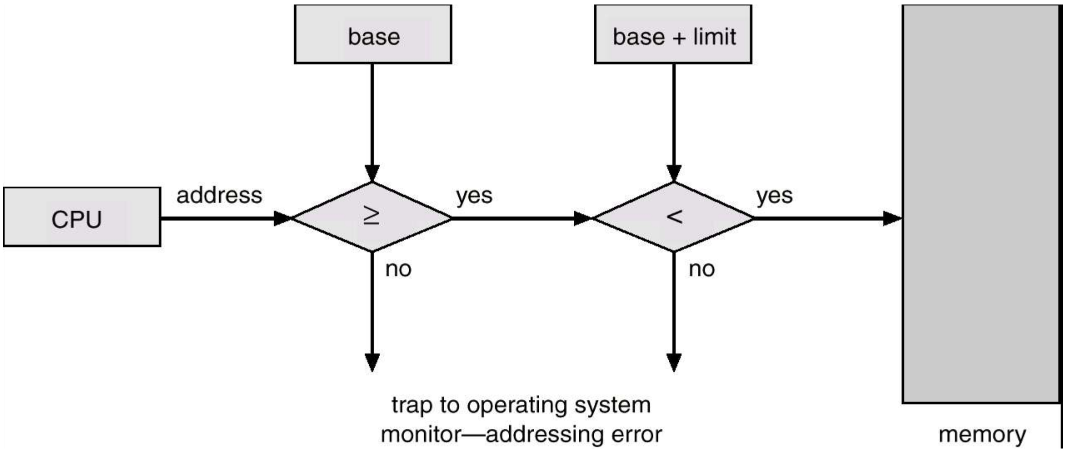
To prevent a user program from performing illegal I/O we can designate all I/O instructions to be privileged instructions. Thus users cannot issue I/O instructions directly; they must do it through the operating system. For I/O protection to be complete we must ensure that a user program can never gain control of the computer in monitor mode. If it could, I/O protection could be compromised.

Memory Protection

This is required to protect the operating system's memory area from programs and one program's memory area from other programs. Memory protection is accomplished by using two registers, usually called a *base* and a *limit*. For example, as shown in the figure below, if the base registers contains 300040 and the limit register is 120900, then the program can access all addresses from 300040 to 420940 inclusive.



The CPU hardware compares every address generated in user mode with these registers in accordance with the figure shown below. Any attempt by a program running in user mode to access monitor memory or other users' memory results in a trap to the monitor, which treats the attempt as a fatal error. The loading of the base and limit registers requires a privileged instruction and hence can only be done by the operating system.



CPU Protection

The operating system must prevent a user program from getting stuck in an infinite loop, and never returning control to the operating system. This can be done using a timer. The timer is implemented using a fixed rate clock and a counter. Before turning over control to a user program, the operating system must ensure that the timer is set to interrupt. When the timer interrupts, the control is passed to the operating system. Instruction that modify the operating of the timer have to be privileged.

Please note that the figures and some of the description provided above has been adapted from Operating System Concepts, Fourth Edition by Abraham Silberschatz and Peter B. Galvin.

Question 3: (20 Marks)

If a cache can be made as large as the device it is caching for (for instance, a cache as large as a disk) why not do so and eliminate the device?

Answer:

Even if a cache could be made as large as a device it is caching for (for instance, a cache as large as a disk) the device cannot be eliminated altogether because the cache may be volatile and disk is non-volatile. While the system is powered-up during operation the cache can retain its contents but if the power were to be removed, even for an instance, the cache would loose its contents for ever. The disk, however, stores its contents magnetically and can retain them if the power is removed.

SOLUTION

Question 4: *(20 Marks)*

A computer has a cache and a main memory. If a reference word is in the cache, 20 ns are required to access it. If it is in main memory but not in the cache, 60 ns are needed to load it into the cache and then the reference is started again. If the cache hit ratio is 0.9, what is the average time in nanoseconds required to access a referenced word on this system.

Answer:

Average Access Time = (Hit Rate × Hit Time) + (Miss Rate × Miss Time)

Average Access Time = $(0.9 \times 20 \times 10^{-9}) + ((1- 0.9) \times (60 \times 10^{-9} + 20 \times 10^{-9}))$

Average Access Time = $18 \times 10^{-9} + (0.1 \times 80 \times 10^{-9})$

Average Access Time = 26×10^{-9}

Average Access Time = 26 nanoseconds

--- THE END ---