Interfacing I/O devices to the Memory, Processor, and Operating System

- How is a user I/O request transformed into a device command and communicated to the device?
  - E.g., file read/write, mouse movement, keyboard stroke
- How is data actually transferred to or from a memory location?
- What is the role of the operating system?

20

Operating System Responsibilities wrt I/O system

- Characteristics of I/O devices
  - Shared by multiple programs
  - Interrupt driven
  - Low-level control is complex
- OS functions
  - Must provide protection
    - E.g. must not allow file owned by one user to be deleted by another user
  - Must provide abstractions for accessing device
    - E.g. file abstraction for a collection of blocks on disk
  - Must handle interrupts
  - Must try to provide “fairness” in accessing I/O devices
  - Must try and manage I/O devices so that throughput is maximized

21

Communication between I/O devices and the OS

- To perform its functions wrt I/O system, the operating system must be able to communicate with I/O devices and to prevent user programs from accessing the I/O devices directly
- Three types of communication
  - OS must be able to give commands to I/O devices
  - A device must be able to notify the OS when it has completed a command or if there is an error
  - Data must be transferred between the I/O device and memory

Giving commands to I/O devices

- CPU must be able to address the device and to supply one or more commands
- Two methods for addressing the device
  - Memory-mapped I/O
    - Portions of a program's address space are assigned to I/O devices
    - Reads and writes to these addresses are interpreted as commands to the device
    - These memory addresses are not directly accessible to user programs
  - Special I/O instructions
    - I/O instructions can specify both the device number and the command word (or the location of the command word in memory)
    - I/O instructions can only be executed by the operating system

Communication with the Processor

- Two methods
  - Polling
    - Device status bits are periodically checked to see if it is time for the next I/O operation
  - Interrupt-driven I/O
    - Device delivers interrupt to the CPU when it requires attention
    - Interrupts are like exceptions except that they are not associated with any instruction
    - CPU can check before starting a new instruction if an interrupt has been delivered
    - Interrupt-handling: Can be vectored or can use a Cause register (Recall Exception-handling from Chapter 5)

24

Polling vs Interrupt-driven I/O

Assume that the number of clock cycles for a polling operation is 100. For a processor executes at 50 MHz, what is the overhead of polling

1. For a mouse that is polled 30 times per second?
2. For a floppy disk that transfers data to the processor in 16-bit units and has a data transfer rate of 50 KB/second?
3. For a hard disk transferring data in 1 word chunks at 2 MB/sec?

For the mouse

clock cycles used per second for polling = 30 x 100 = 300

Fraction of processor cycles used for polling = 3000/(50 X $10^{6)}$

= 0.006%

25

Polling vs Interrupt-driven I/O cont'd

For the floppy drive

Number of polling operations per second

(if we don't want to lose data) = (50 KB/sec) / (2 bytes/access)

= 25 K polling accesses per second

Clock cycles for polling = 25 K X 100 = 25 X 1024 X 100

= 25.6 X 10^5 clock cycles per second

Fraction of CPU cycles = (25.6 X 10^5) / (50 X 10^6) = 5%

For the hard disk

Rate of polling = (2 MB/sec)/ (4 bytes per access)

= 500 K polling accesses per second

Clock cycles = 500 K X 100 = 51.2 X 10^6

Fraction of CPU cycles = 51.2 X 10^6/ 50 X 10^6 = 100% !!!!

26

Polling vs Interrupt-driven I/O cont'd

- Suppose overhead of interrupt handling is 100 clock cycles. How much overhead when floppy disk is active?

Rate of interrupts = (50 KB/sec) / (2 bytes/interrupt)

= 25 K interrupts per second

Clock cycles for handling interrupts

= 25 K X 100 = 25 X 1024 X 100

= 25.6 X 10^5 clock cycles per second

Fraction of CPU cycles = (25.6 X 10^5) / (50 X 10^6) = 5%

The difference from polling is that 5% of the CPU cycles per second are used for handling interrupts only if the floppy is busy

At other times, overhead is 0%. For polling, the overhead is always 5%

27

Transferring data between device and memory

- Two methods
  - Interrupt-driven I/O
    - Processor is involved in data transfer
    - Problem: 100% overhead in the case of the hard disk example
  - Direct Memory Access (DMA)
    - Data is transferred directly from the device to memory (or vice versa)
    - Processor is involved only in
      1. Initiating the DMA transfer
      2. Handling interrupt at the end of DMA transfer

28

DMA

- Implemented with special controller that transfers data between memory and I/O device independent of the processor
- Three steps in DMA transfers
  1. Processor sets up the DMA transfer by supplying identity of device, operation to perform, memory address that is source or destination of data, number of bytes to be transferred
  2. DMA controller starts the operation (arbitrates for the bus, supplies address, reads or writes data), until the entire block is transferred
  3. DMA controller interrupts the processor, which then takes the necessary actions

29

Hard disk DMA example

Find overhead for using DMA for data transfer from a hard disk.
Assume initial DMA setup = 1000 cycles
Interrupt handling on DMA completion = 500 cycles
Average size of data transfer = 4 KB

Each DMA transfer takes (4 KB)/(2 MB/sec) = 2×10^{-3} seconds
CPU cycles used for DMA transfer = 1000 + 500 = 1500
Total CPU cycles during DMA transfer = $(50 \times 10^6) \times (2 \times 10^{-3})$
$= 100 \times 10^3$
Fraction of CPU cycles used for DMA = $1500 / 100 \times 10^3$
= 1.5 %

30