INPUT/OUTPUT HANDLING

The user need not know how peripheral devices work. Os handles the devices on behalf of the user

Pillars of a computer System

- The four pillars on which a computer stands include
  - CPU
  - Memory
  - I/O subsystem
  - Filling System
- The I/O subsystem is the most complicated. Reason – the wide variety of I/O devices

Categories of I/O Devices

External devices that engage in I/O with computer systems can be grouped into three categories:

Human readable

- suitable for communicating with the computer user
- printers, terminals, video display, keyboard, mouse

Machine readable

- suitable for communicating with electronic equipment
- disk drives, USB keys, sensors, controllers

Communication

- suitable for communicating with remote devices
- modems, digital line drivers

Differences in I/O Devices

- Devices differ in a number of areas:

Data Rate

- there may be differences of magnitude between the data transfer rates

Application

- the use to which a device is put has an influence on the software

Unit of Transfer

- data may be transferred as a stream of bytes or characters or in larger blocks

Data Representation

- different data encoding schemes are used by different devices

Error Conditions

- the nature of errors, the way in which they are reported, their consequences, and the available range of responses differs from one device to another

Design Objectives of OS

- How is the OS designed to cope with the wide variety of devices
  - Character code independence
  - Device independence
  - Efficiency
  - Uniform treatment of devices

Character Code Independence

- Data representation varies in different peripheral devices
- Standard OS character code independent of any particular device used to represent data internally.
- A translational mechanism used to convert data to device specific format

Device Independence

- User programs should be independent from the devices they operate on.
- programs can access any I/O device without specifying device in advance (floppy, hard drive, or CD-ROM)

- Leads to an organization of the operating system into a series of layers
- Each layer performs a related subset of the functions required of the operating system
- Layers should be defined so that changes in one layer do not require changes in other layers

OS Layers – How is device independence achieved in OS?

- Application Programs
- i/o control system (s/w)
- Device Drivers(s/w)
- Device Controllers (H/w)
- Device(H/w)

- Application Programs – Within the application, I/O activity is expressed in user oriented terms. Print, save, open etc. System calls are made which invoke OS functions.
- I/O control system – part of os which deals with I/O related system calls. Performs the initial processing and validation on the request and routes it to the appropriate handler.

- Device drivers – Is a software module which manages the communication with and control of a specific I/O device.
- Device Controllers - I/O units typically consist of an electronic component, which often takes the form of a printed circuit card. A device controller is the electronic circuitry, attached to I/O bus of the computer and provides a hardware interface between the computer and the device itself
- Device – Actual peripheral device

Efficiency

- Major effort in I/O design
- Important because I/O operations often form a bottleneck
- Most I/O devices are extremely slow compared with main memory and the processor
- Block Device – Transmit a set of characters simultaneous
- Character device – Transmit one character at a time

Buffering/Spooling

- Technique used to remove bottlenecks in the computer system
- Fast and slow peripheral devices communicate via a buffer.
- Perform input transfers in advance of requests being made and perform output transfers some time after the request is made

Block-oriented device

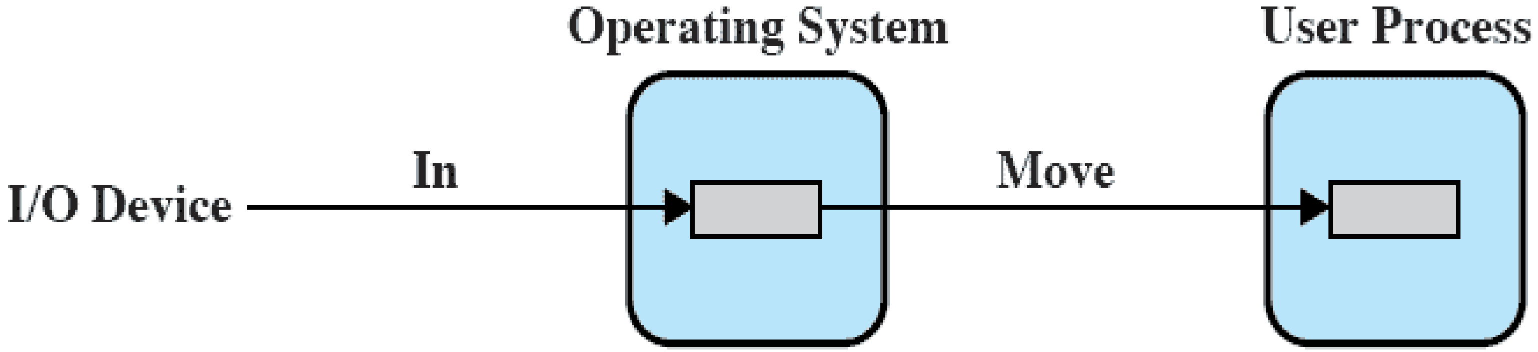
- stores information in blocks that are usually of fixed size
- transfers are made one block at a time
- possible to reference data by its block number
- disks and USB keys are examples

Stream-oriented device

- transfers data in and out as a stream of bytes
- no block structure
- terminals, printers, communications ports, and most other devices

Single Buffer

- Operating system assigns a buffer in main memory for an I/O request



(b) Single buffering

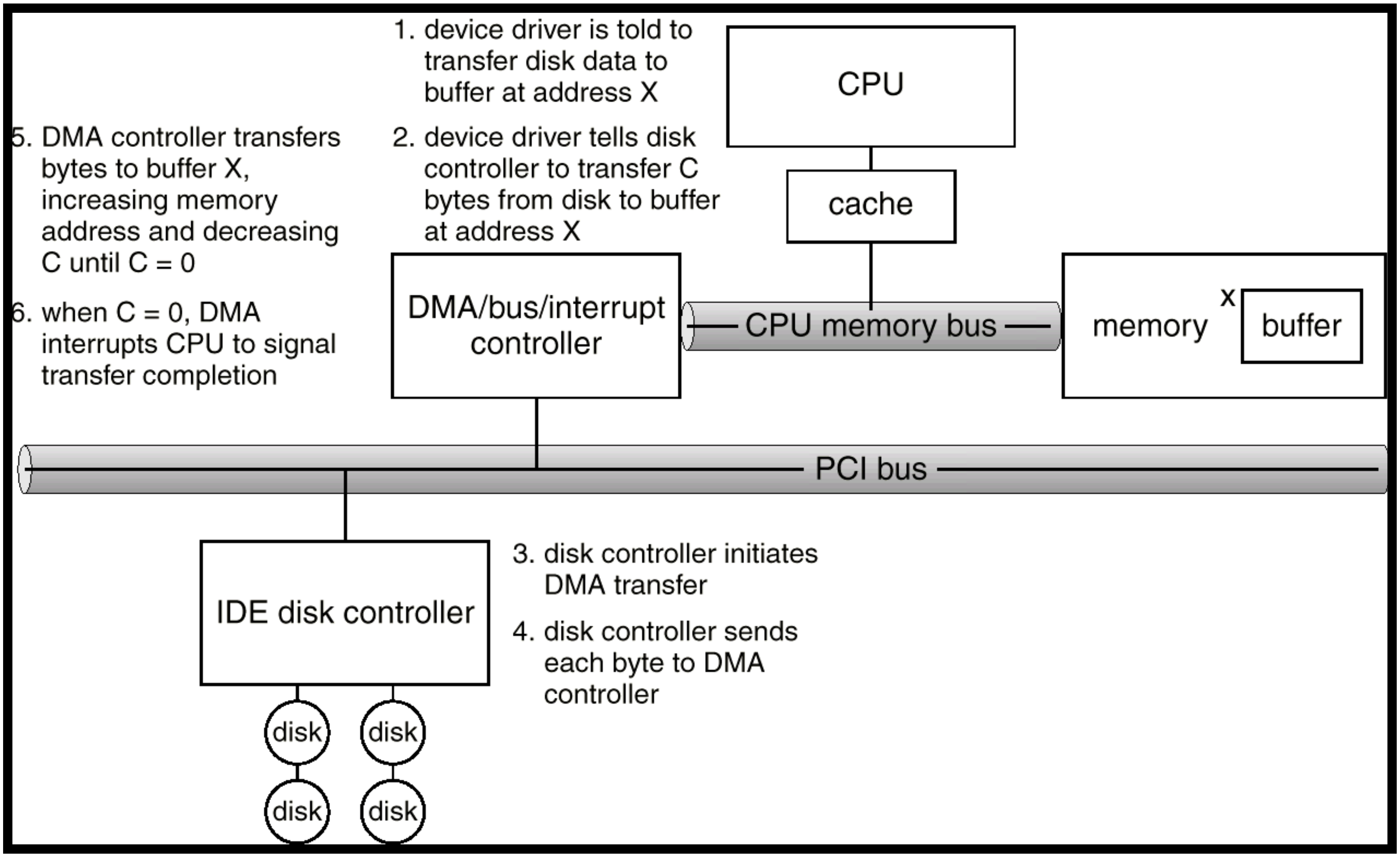
Uniform Treatment of devices

- All devices are handles the same.
- Design the I/O system & device routines such that dealing with different devices is done by changing parameters in the routine i.e the routine are very simillar if not identical in structure and operation. Different parameter values make the same routine handle devices differently according to their function.
- In unix all devices are regarded as files/streams
- Device descriptors store device characteristics

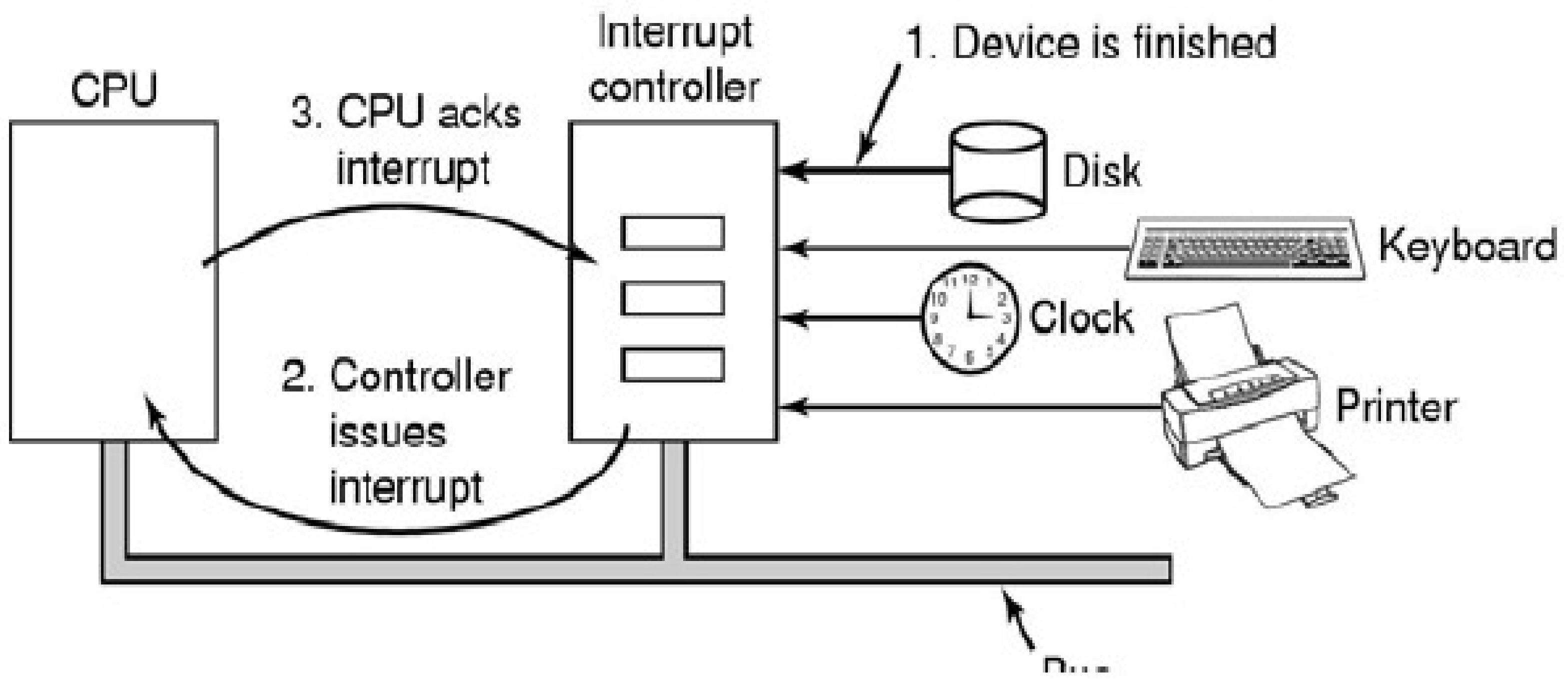
Organization of the I/O Function

- Three techniques for performing I/O are:
- **Programmed I/O**
  - the processor issues an I/O command on behalf of a process to an I/O module; that process then busy waits for the operation to be completed before proceeding
- **Interrupt-driven I/O**
  - the processor issues an I/O command on behalf of a process
    - Processor proceeds with other operations
    - An interrupt alerts the processor when the process is through
- **Direct Memory Access (DMA)**
- a DMA module controls the exchange of data between main memory and an I/O module.
- Requires DMA controller
- Bypasses CPU to transfer data directly between I/O device and memory

Six Step Process to Perform DMA Transfer



Interrupt I/O



- END