Operating Systems

End Semester Revision
December 2018

File Systems

Long-term storage

Need for long-term storage:

- Computer applications need to store and retrieve information
- There is need for long-term storage, because
  - Limited amount of space in main memory or virtual memory
  - Information must be retained beyond the life of a process
  - Multiple processes may need to access (parts of) same information

Requirements for long-term storage:

- Essential requirements for long-term information storage:
  - It must be possible to store a very large amount of information.
  - The information must survive the termination of the process using it.
  - Multiple processes must be able to access the information at once.

File Systems

Files:

- Files are logical units of information created by processes.
- Processes can read existing files and create new ones if need be.
- Files are managed by the operating system.
- The operating system dealing with files is known as the file system.

Files systems:

- Implement an abstraction (files) for secondary storage
- Organize files logically (directories)
- Permit sharing of data between processes, people, and machines
- Protect data from unwanted access (security)

File Types

1. Regular files are the ones that contain user information
2. Directories are system files for maintaining the file system structure
3. Character special files are related to input/output
4. Block special files are used to model disks

File Access

Sequential access

- a process could read all the bytes or records in a file in order, starting at the beginning, but could not skip around and read them out of order
- Used with magnetic tapes

Random-access

- Files whose bytes or records can be read in any order
- Essential for many applications, such as database systems
- Used with discs

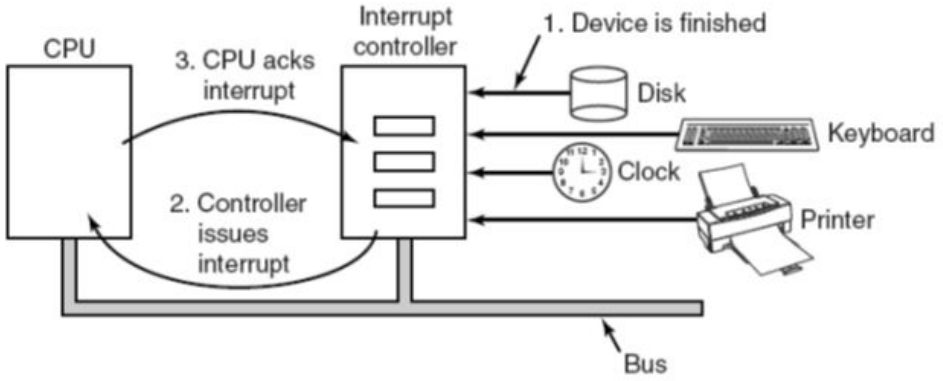
Input/Output

I/O devices

- I/O devices can be roughly divided into two categories:
- Block devices:
  - Stores information in fixed-size blocks, each one with its own address
  - All transfers are in units of one or more entire (consecutive) blocks
  - Each block can be written or read independently
  - Hard disks, Blu-ray discs, and USB sticks
- Character devices:
  - Delivers or accepts a stream of characters, without any block structure
  - Not addressable and does not have any seek operation
  - Printers, network interfaces, keyboards

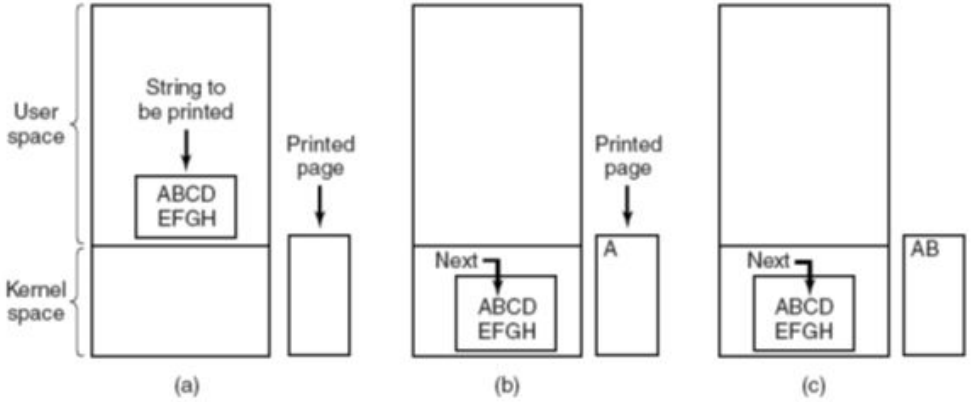
Interrupts

- When an I/O device has finished the work given to it, it causes an interrupt.
- It does this by asserting a signal on a bus line that it has been assigned.
- This signal is detected by the interrupt controller chip, which then decides what to do.
- The interrupt signal causes the CPU to stop what it is doing and start doing something else.



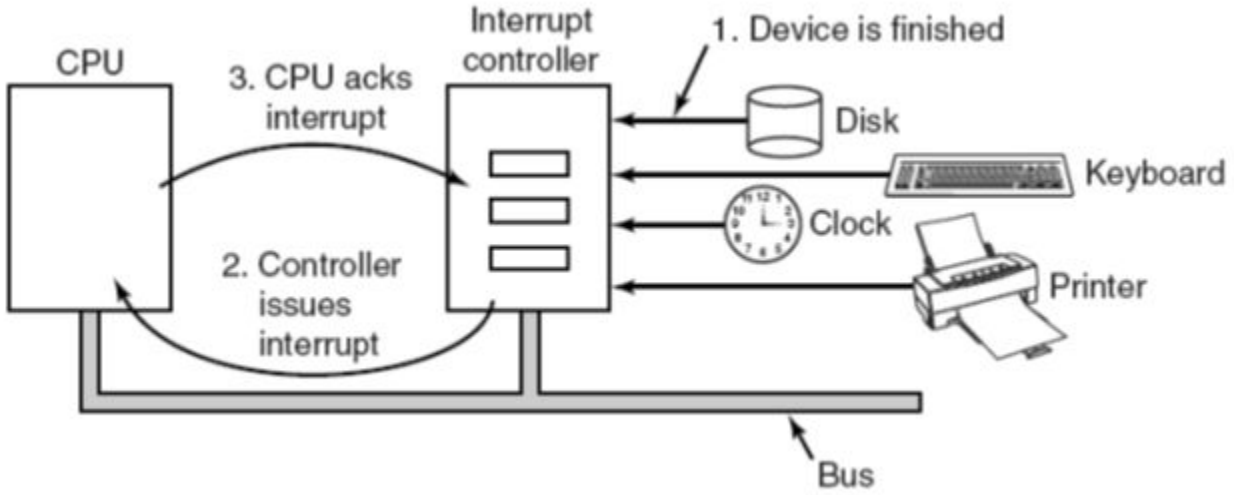
Types of I/O Methods - Programmed I/O

- CPU does all the work, simplest form of I/O
- Has the disadvantage of tying up CPU full time until all I/O is done (polling)
- Fine if waiting is short or CPU has nothing else to do
- In most complex systems, CPU has other things to do
- E.g. First the data are copied to the kernel. Then the operating system enters a tight loop, outputting the characters one at a time.



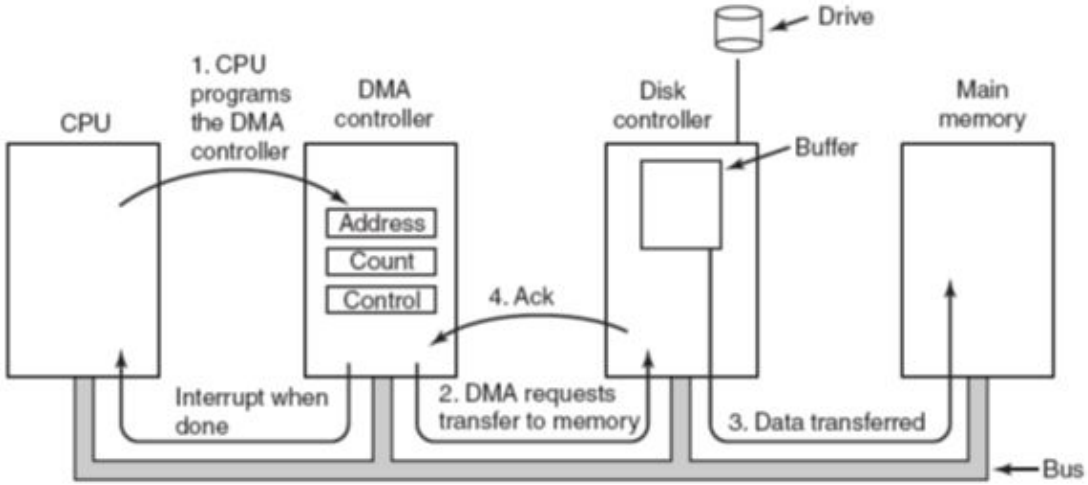
Types of I/O Methods - Interrupt-Driven I/O

- Allow the CPU to do something else while waiting for I/O.
- Whenever CPU is waiting for some I/O, it can switch to another process, until an interrupt is received from I/O on completion.
- However, too frequent interrupts within each I/O request can waste CPU time.

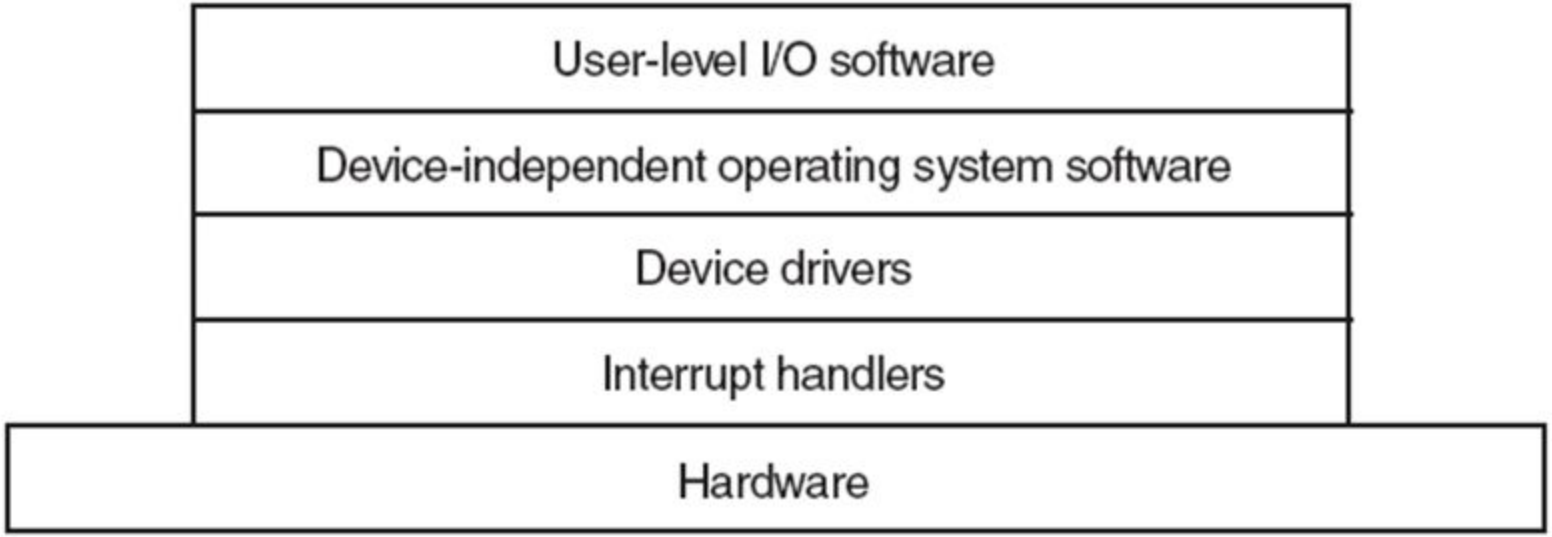


Types of I/O Methods - I/O Using DMA

- DMA controller manages the I/O request as a whole
- CPU is not interrupted within a I/O request
- Reduces the number of interrupts
- However, DMA controllers are usually much slower than main CPU.
  - So if CPU has nothing else to do, it may have to wait longer than if it did I/O on it's own.



I/O Software Layers



Security

Security Goals

- Confidentiality
  - if the owner of data decides to make available only to certain people and no others, the system should guarantee that release of the data to unauthorised people never occurs
- Integrity
  - unauthorised users should not be able to modify any data (changing the data, removing data and adding false data) without the owner's permission
- Availability
  - nobody can disturb the system to make it unusable, such as in the form of denial-of-service attacks that are increasingly common

Goal	Threat
Confidentiality	Exposure of data
Integrity	Tampering with data
Availability	Denial of service

Attacks

Passive attacks:

- try to steal information passively
- sniff the network traffic and tries to break the encryption to get to the data

Active attacks:

- try to make a computer program misbehave
- take control of a user’s Web browser to make it execute malicious code

Access Control

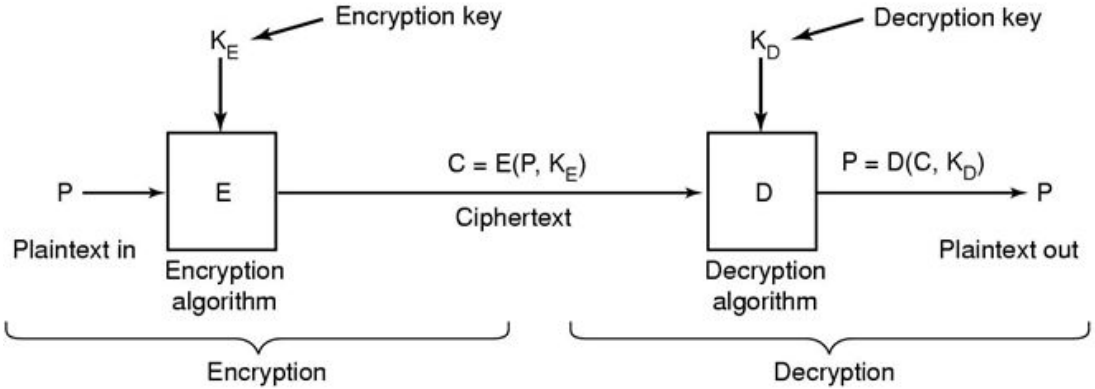
- A computer system contains many resources, or ‘‘objects,’’ that need to be protected.
- These objects can be hardware (e.g., CPUs, memory pages, disk drives, or printers) or software (e.g., processes, files, databases, or semaphores).
- A model of what is to be protected and who is allowed to do what is necessary for the operating system.
- There are various models for doing this,
    1. Protection Domains
    2. Access Control Lists
    3. Capabilities

Authentication

- Every secured computer system must require all users to be authenticated at login time.
- General principles of authenticating users:
  1. Something the user knows – Known things password, PIN
  2. Something the user has – Physical objects like smartcard, phone
  3. Something the user is – Biomatrices like fingerprint, iris scan

Basic Cryptography

- Cryptography plays an important role in security and operating systems use cryptography in many places.
  - Some file systems can encrypt all the data on disk
  - Protocols like IPSec may encrypt and/or sign all network packets
  - Most operating systems scramble authentication passwords
- Take a message or file, called the plaintext, and encrypt it into ciphertext in such a way that only authorized people know how to convert it back to plaintext.



Digital Signatures

- Digital signatures make it possible to sign emails and other digital documents in such a way that they cannot be repudiated by the sender later.
- One common way is to first run the document through a one-way cryptographic hashing algorithm that is very hard to invert.
- The hashing function typically produces a fixed-length result independent of the original document size.
- The most popular hashing functions used is SHA-1 (Secure Hash Algorithm), which produces a 20-byte result (NIST, 1995).