CD4002/CN4002 Computer Systems and Networks

Topic 6 Operating Systems

University of East London

Pioneering Futures Since 1898

Agenda

- The role of the operating system
- Types of operating system
- Operating system services and facilities

University of East London

Pioneering Futures Since 1898

Bare Bones Computer System

- It cannot load instructions into main memory
- No user interface except for I/O routines provided by executing program
- Is idle when waiting for user input
- No facility to store, retrieve, or manipulate files
- No way to control peripheral devices
- Can run only one program at a time; computer halts at end of each program

What is an Operating System?

“is an interface between hardware and user”

“is the program which is responsible for coordinating the activities of the various processes that a PC is executing”

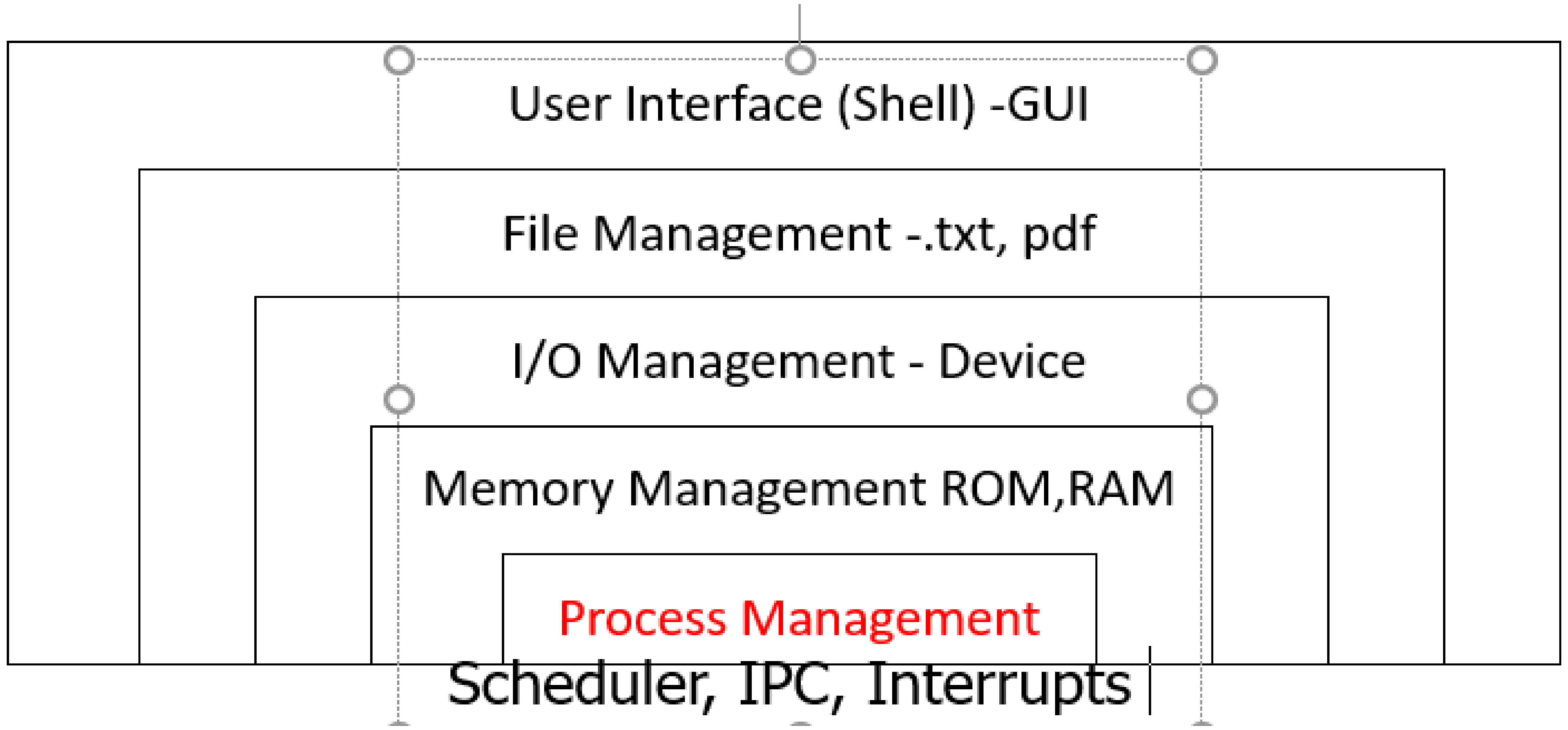
“is a software that controls the allocation and usage of hardware resources such as memory, CPU time, disk space, and input and output devices”

“software that controls the execution of computer programs and may provide various services”

The Functions of a Modern, General Purpose OS

- Resource Allocation and Management
  - Processor(s)
  - Memory
  - Devices
  - Data
- Provides an interface(s) to the underlying hardware for
  - The user and
  - Application programs

O.S. Structure



User Interface (Shell) -GUI

File Management -.txt, pdf

I/O Management - Device

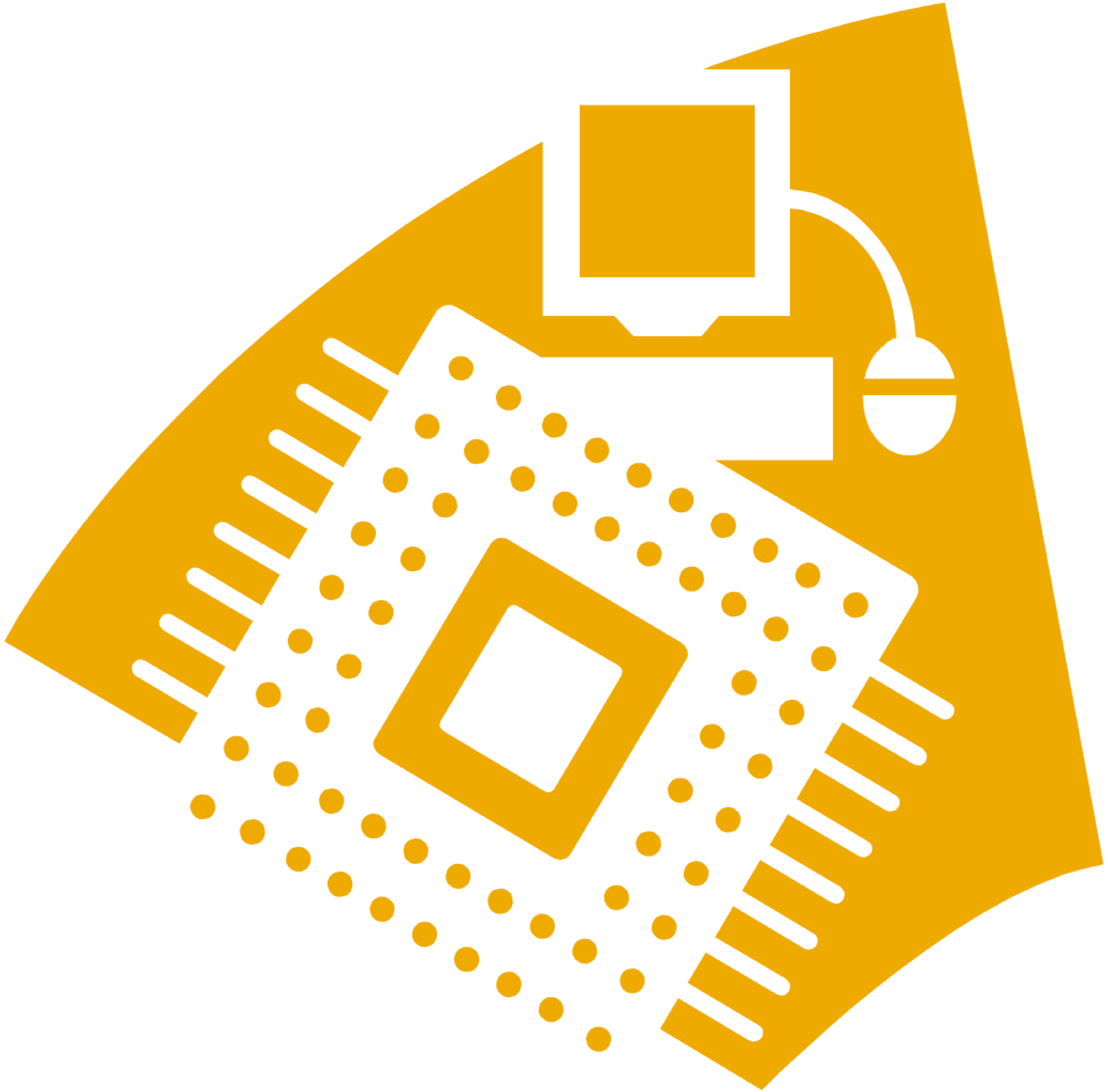
Memory Management ROM,RAM

Process Management

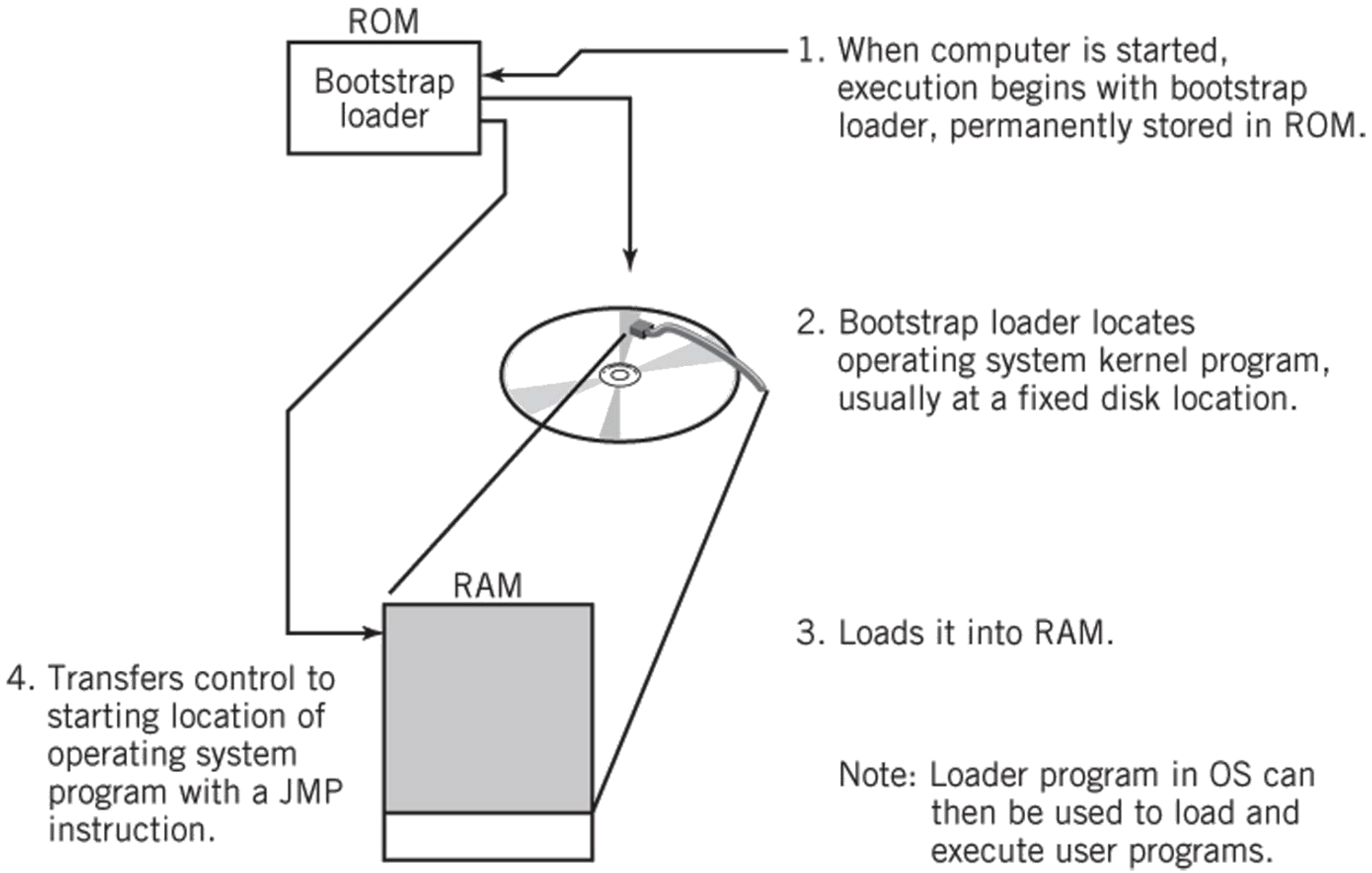
Scheduler, IPC, Interrupts

Component Parts of OS

- Memory Resident
  - Always loaded in memory
  - Commonly called the kernel
  - Contains essential services required by other parts of the operating system and applications.
  - Typically responsible for managing memory management, processes and tasks, and secondary storage
- Memory Non-resident
  - Applications
  - Infrequently used programs, software tools, and commands
- Bootstrap program



Bootstrapping



1. When computer is started, execution begins with bootstrap loader, permanently stored in ROM.

2. Bootstrap loader locates operating system kernel program, usually at a fixed disk location.

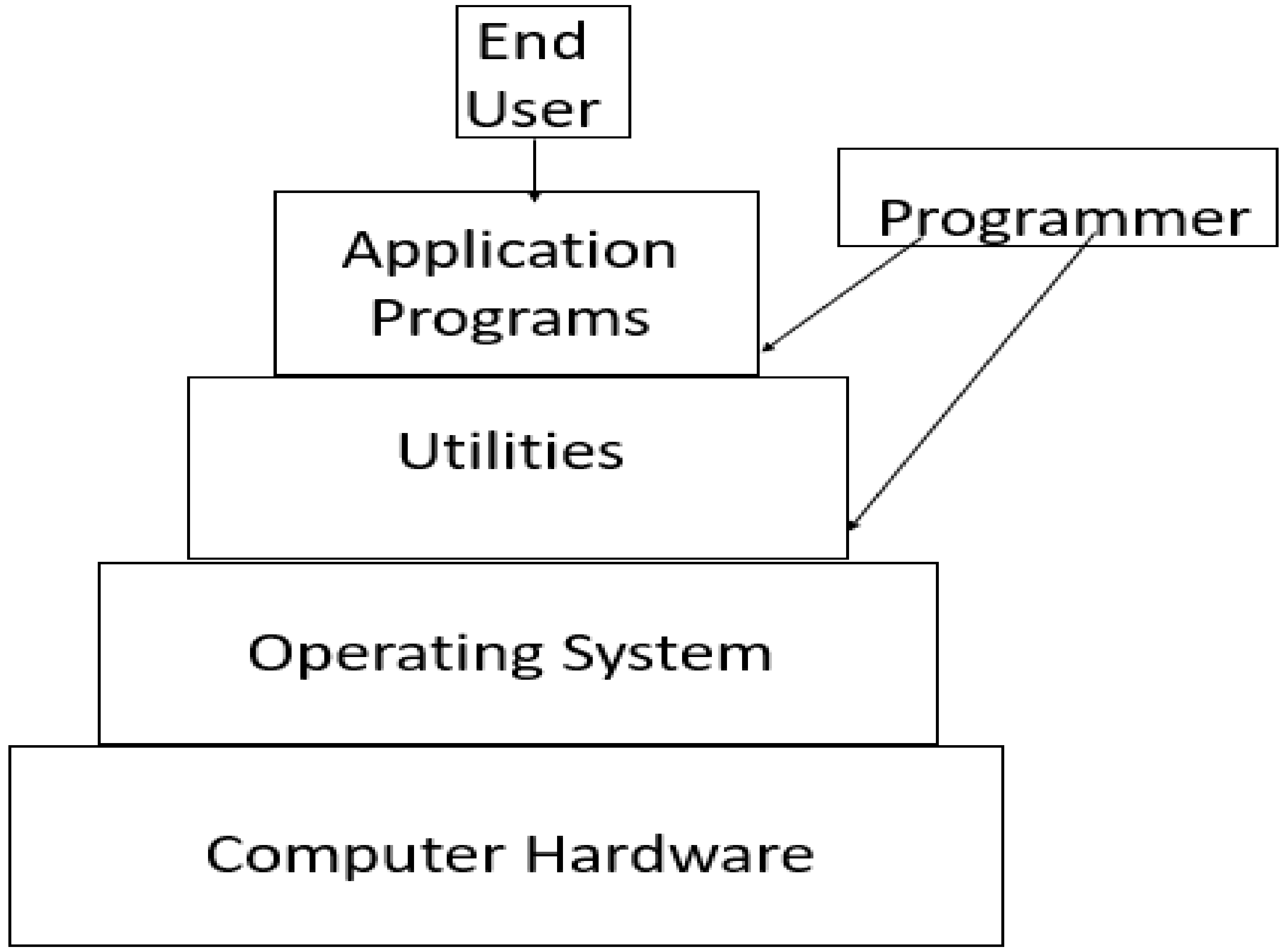
3. Loads it into RAM.

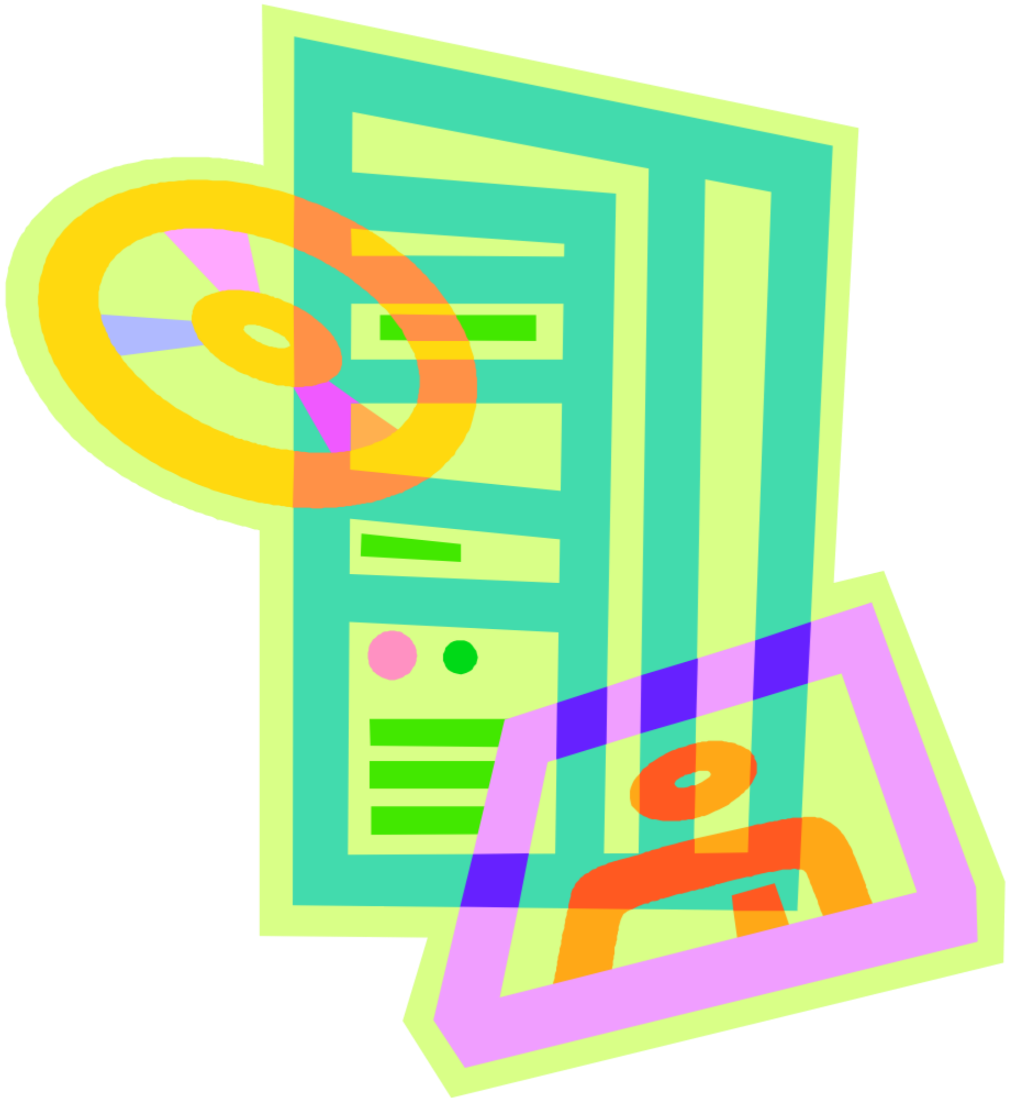
4. Transfers control to starting location of operating system program with a JMP instruction.

Note: Loader program in OS can then be used to load and execute user programs.

Hardware and the OS

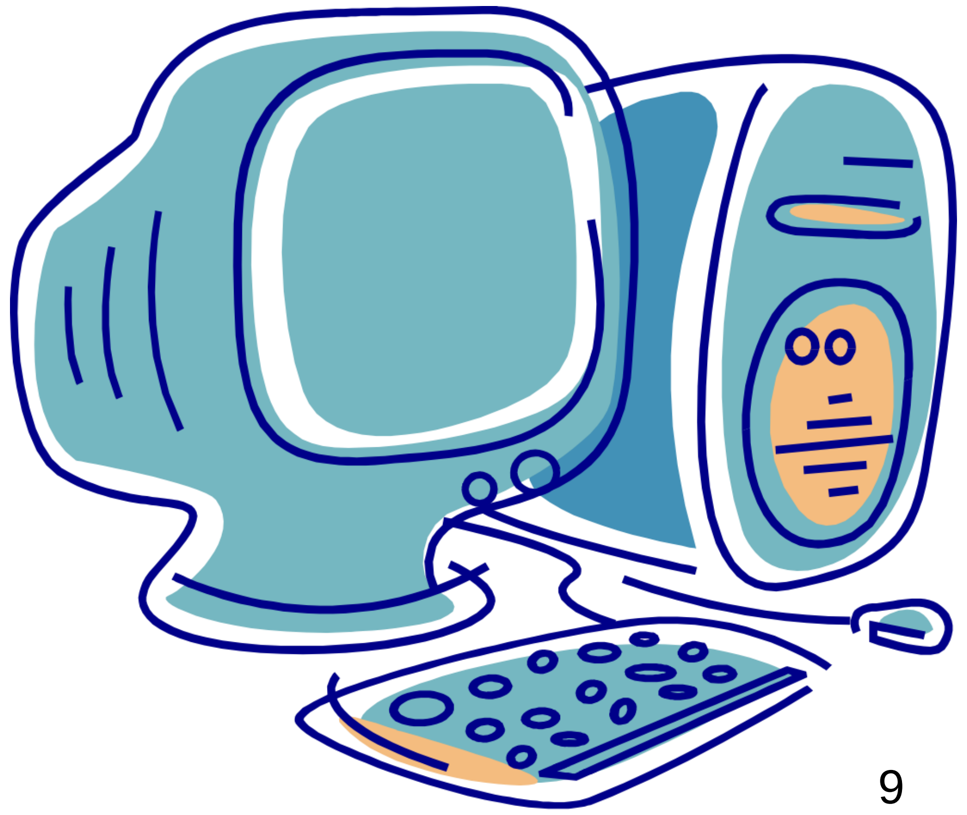
- A hardware platform may support a variety of operating systems
- An operating system may work on a variety of platforms
- A standard operating system that works on different hardware
  - Provides program and file portability
  - Enables user efficiency through recognisable interface
  - Is implemented through a systems programming language like C or C++ as opposed to assembly language





Types of Operating Systems

- Single user, single tasking (obsolete)
- Single-user systems and workstations
  - Predominant systems in use
    - MacOS and Windows
- Multi-user systems and workstations
  - Unix and Linux
- Mainframe systems
  - Manage large scale computing resources
  - Extensive I/O capability
- Network servers
  - Supporting clients connected to the server
  - Improved security, high reliability, backup facilities



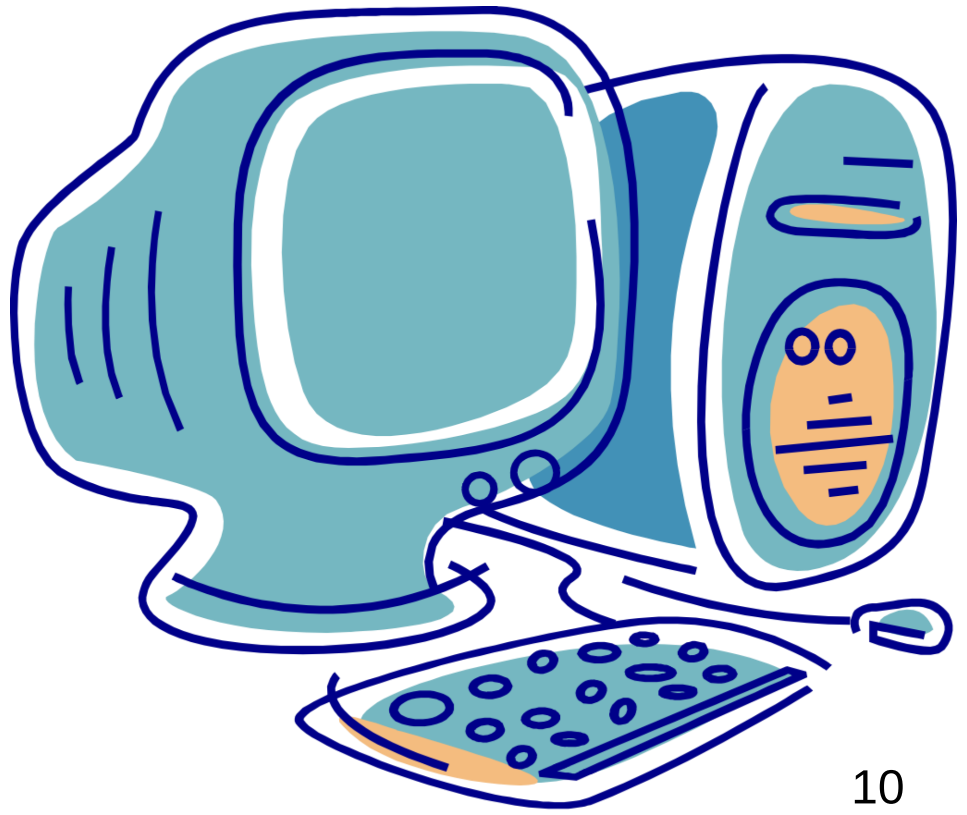
Types of Operating Systems (cont.)

- Real-time systems
  - One or more processes must be able to access the operating system immediately
  - Multitasking system where a real-time program’s interrupts have very high priority

- Distributed systems
  - Processing power distributed among computers in a cluster or network

- Embedded control systems
  - Specialised systems designed to control a single piece of equipment such as an automobile or a microwave oven

- Mobile operating systems
  - Small hand-held devices eg iOS and Android
  - Constraints on memory, storage, CPU execution speed and electrical power



Concurrent Processing

- Multitasking (multiprogramming)
  - Use of concurrent processing to simulate simultaneous execution of multiple programs even when using only a single CPU
  - Supports multiuser systems

- Multiprocessing
  - Actual simultaneous processing of multiple programs using either multiple CPUs or multiple CPU cores

- Early systems were neither multitasking nor multiprocessing

Additional Services Required for Concurrent Processing

- Allocation of resources such as memory, CPU time, and I/O devices to programs

- Protection of users and programs from each other and provision for inter-program communication

- Provision of feedback to system administrators to permit performance optimisation of the computer system

Achieving Multitasking

- While one program is waiting for I/O to take place, another program is using the CPU to execute instructions.

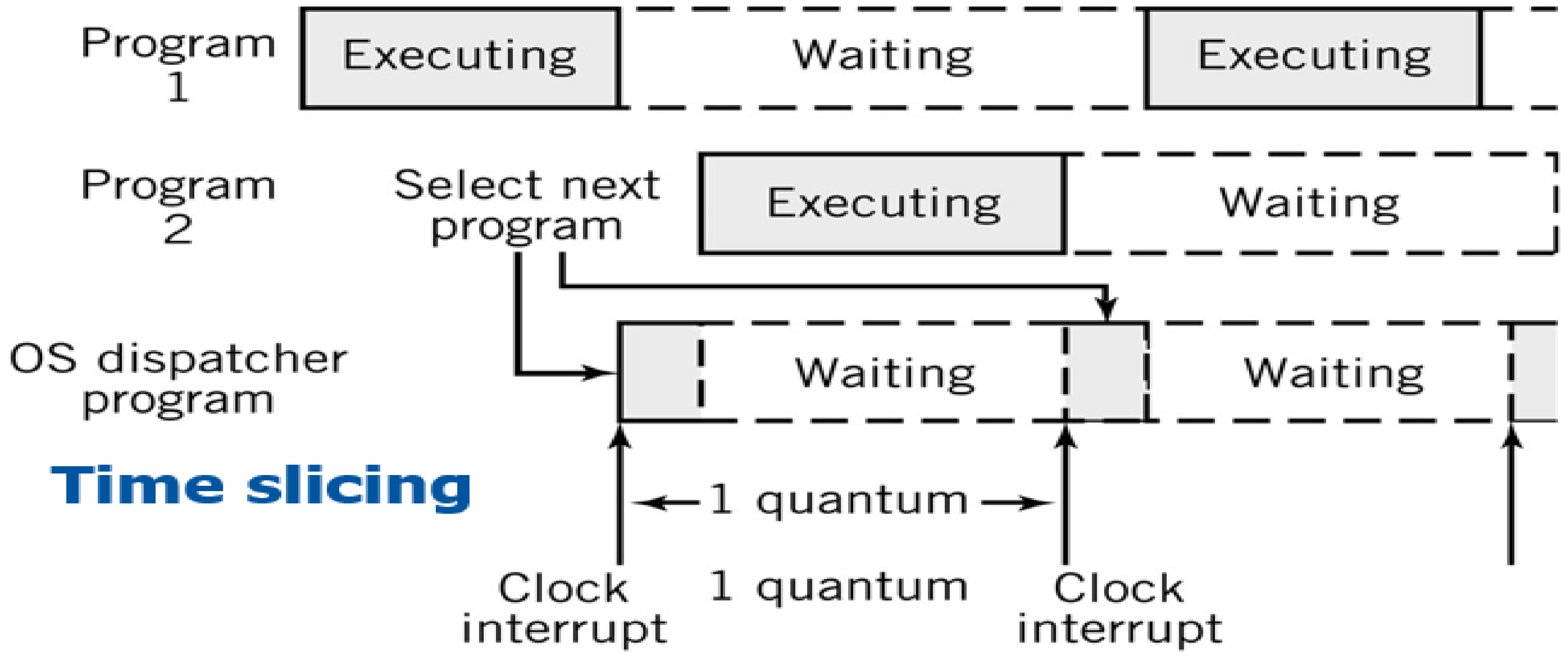
Dispatching

- is the process of selecting which program to run at any given instant

Time-slicing

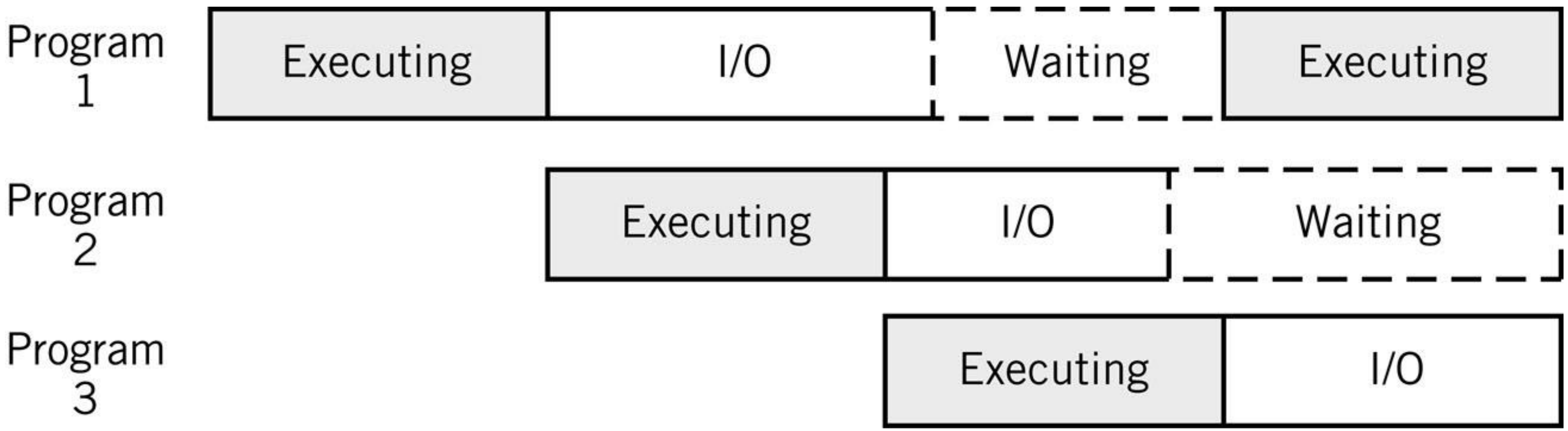
- The CPU may be switched rapidly back and forth between different processes

Time-sharing the CPU



Sharing the CPU during I/O Breaks

- I/O represents a large percentage of a typical program’s execution



Program 1
Executing
I/O
Waiting
Executing

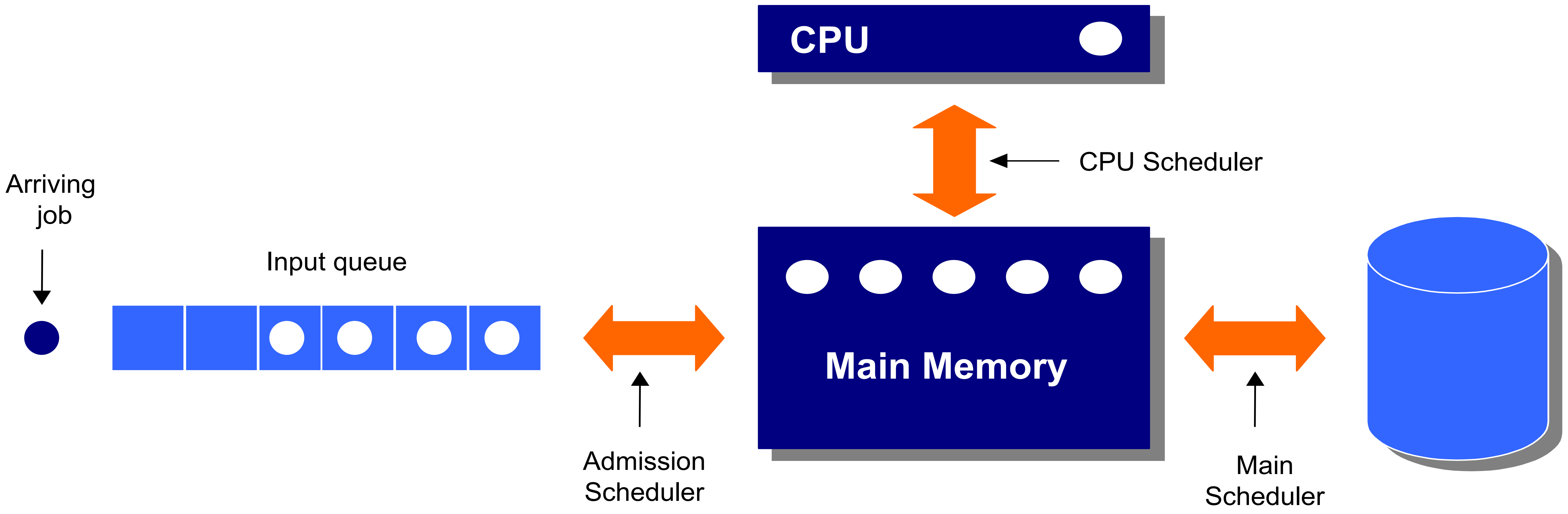
Program 2
Executing
I/O
Waiting

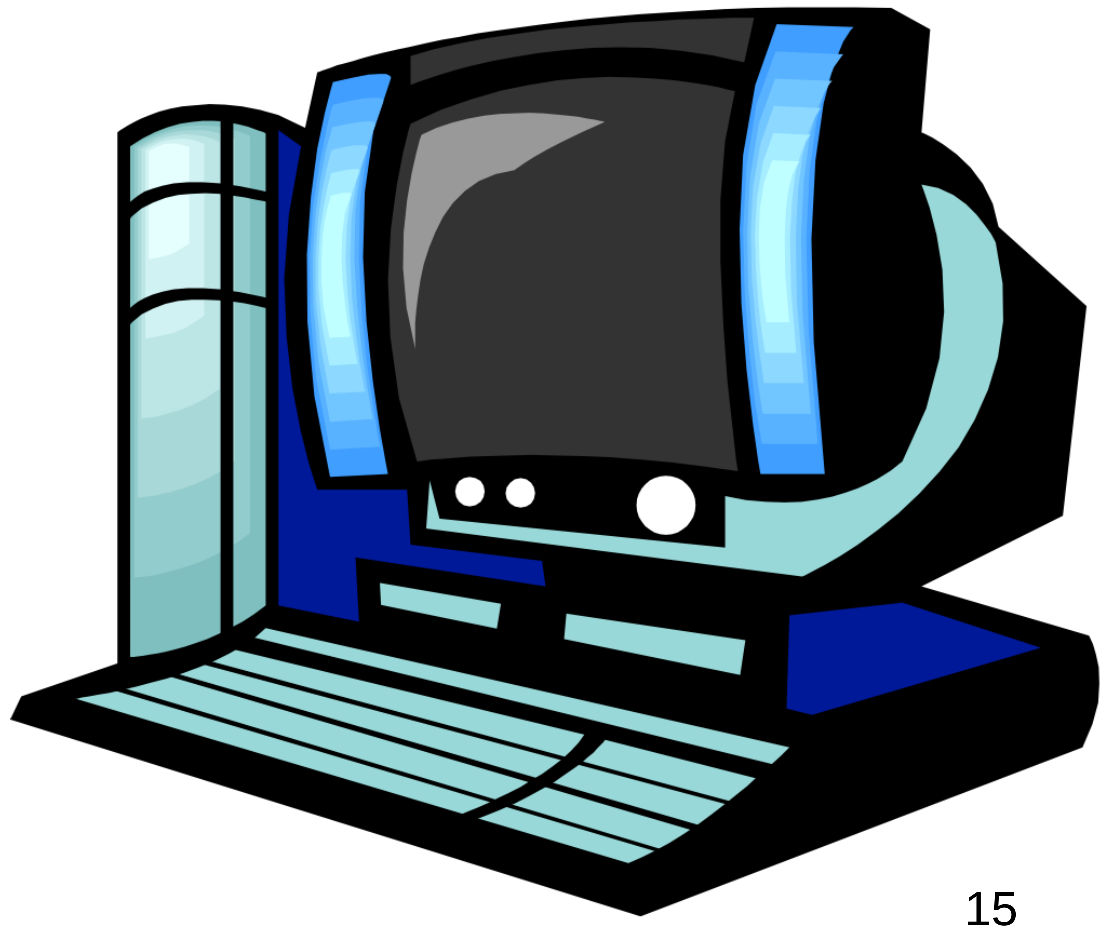
Program 3
Executing
I/O

Process Management

- A process is an executing program
- A thread
  - An individually executable part of a process
  - Shares memory and other resources with other threads of the same process
- Interprocess communication (IPC)
  - Example: a pipe in Linux or Windows that forms a temporary connection between two programs or commands

Scheduling



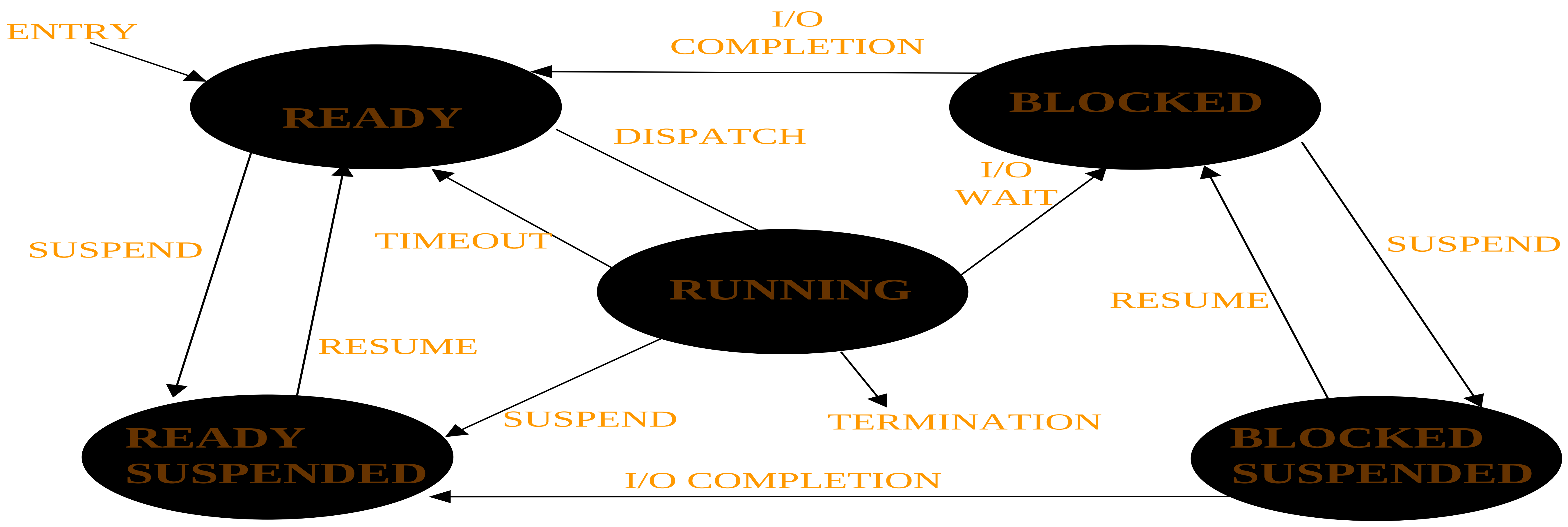


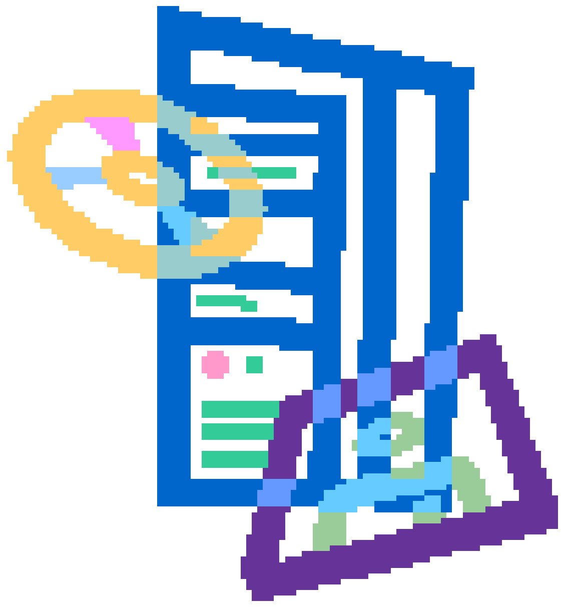
Scheduling

- High-level scheduling
  - Placed in queue based on level of priority and eventually executed
- Dispatching (Short-term scheduling)
  - Actual selection of processes that will be executed at any given time
  - Preemptive – uses clock interrupts
  - Non-preemptive – program voluntarily gives up control
- Context switching
  - Transfer control to the process that is being dispatched

Process State Diagram

- Complete



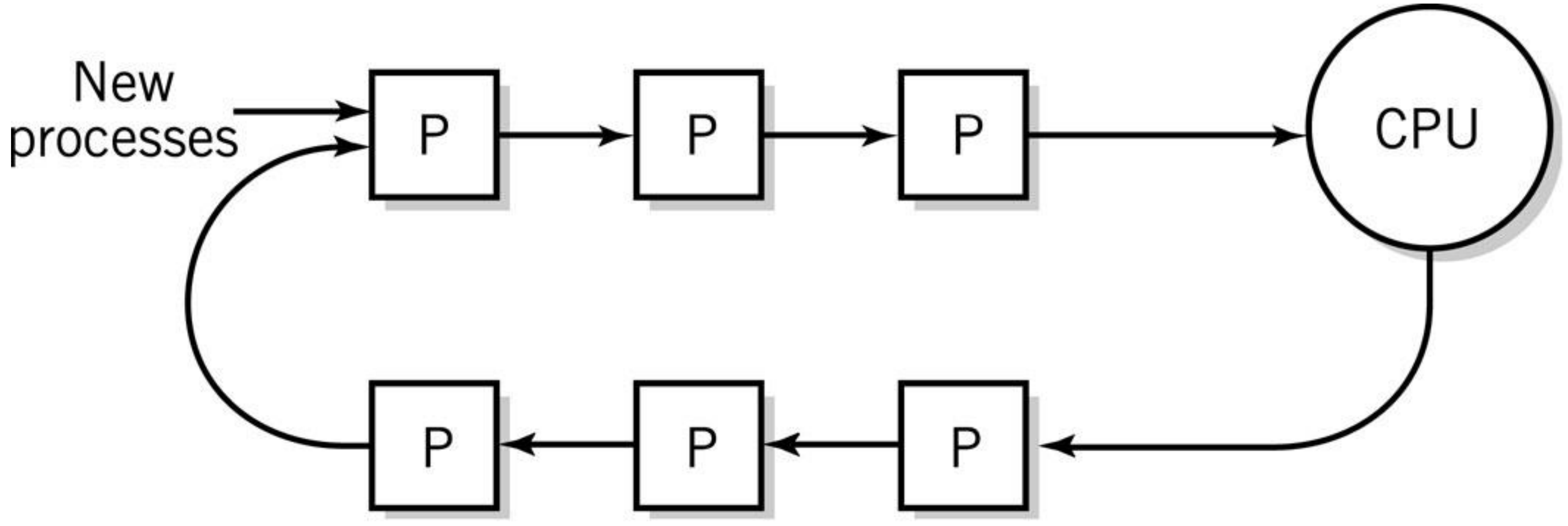


Non Preemptive Dispatching

- First in, first out (FIFO)
  - Unfair to short processes and I/O based processes
- Shortest Job First (SJF)
  - Longer jobs can be starved
- Priority Scheduling
  - Job with the highest priority is selected
  - If multiple jobs have the highest priority then dispatcher selects among them using FIFO

Preemptive Dispatching

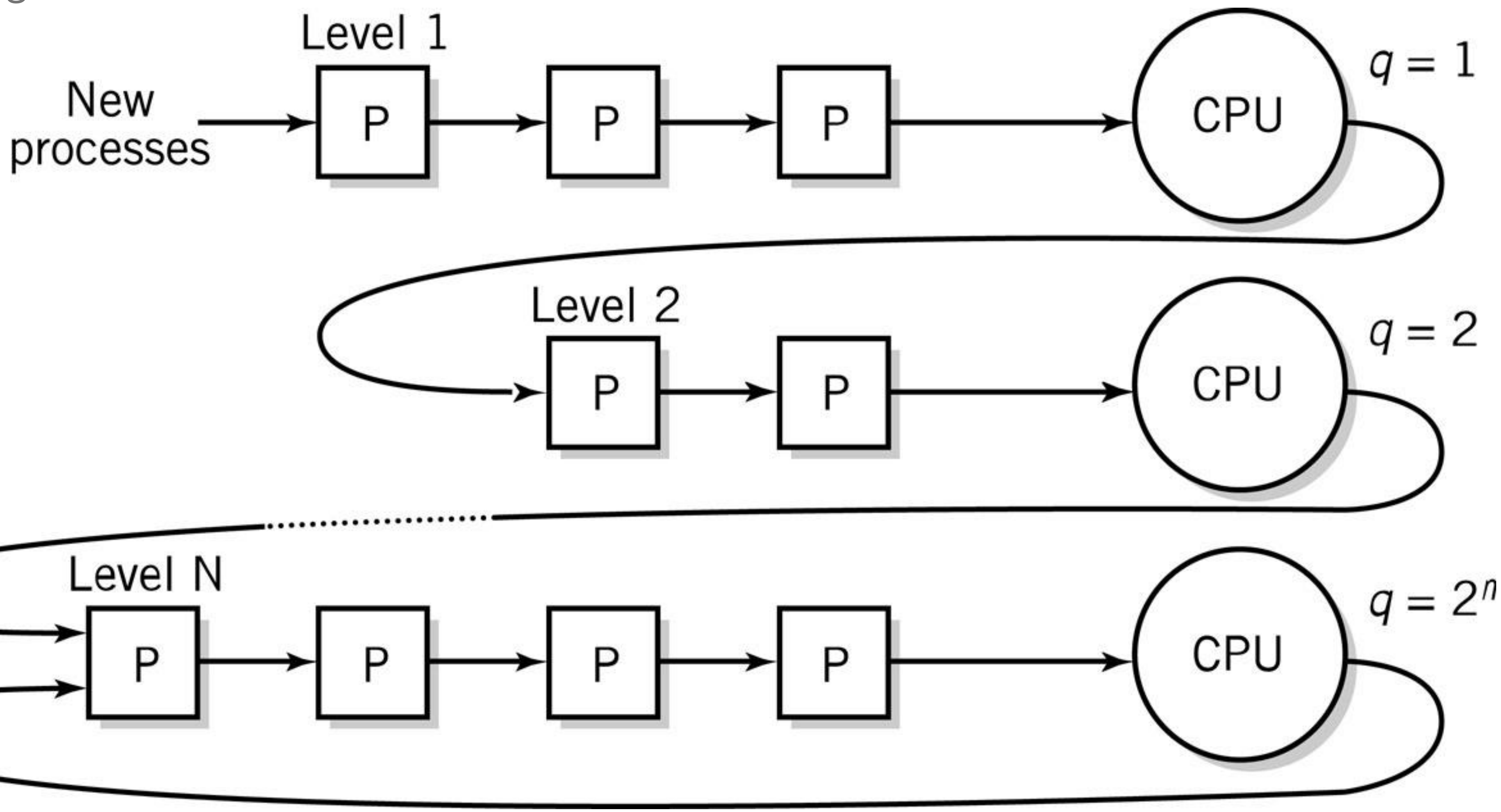
- Round robin
  - Inherently fair and maximises throughput



- Dynamic Priority
  - Based on ratio of CPU time to total time process has been in the system
  - Smallest ratio has highest priority

Preemptive Dispatching (cont.)

- Multilevel feedback queues
  - Favours short jobs, I/O bound jobs
  - Each level assigns more CPU time

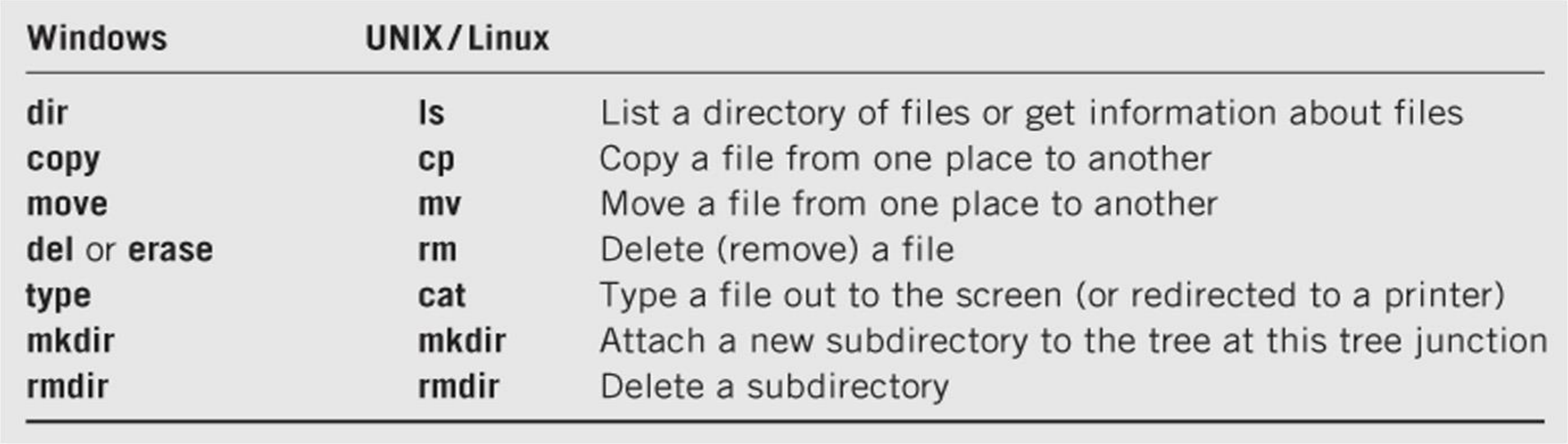


File Management and Common File Commands

File - logical unit of storage

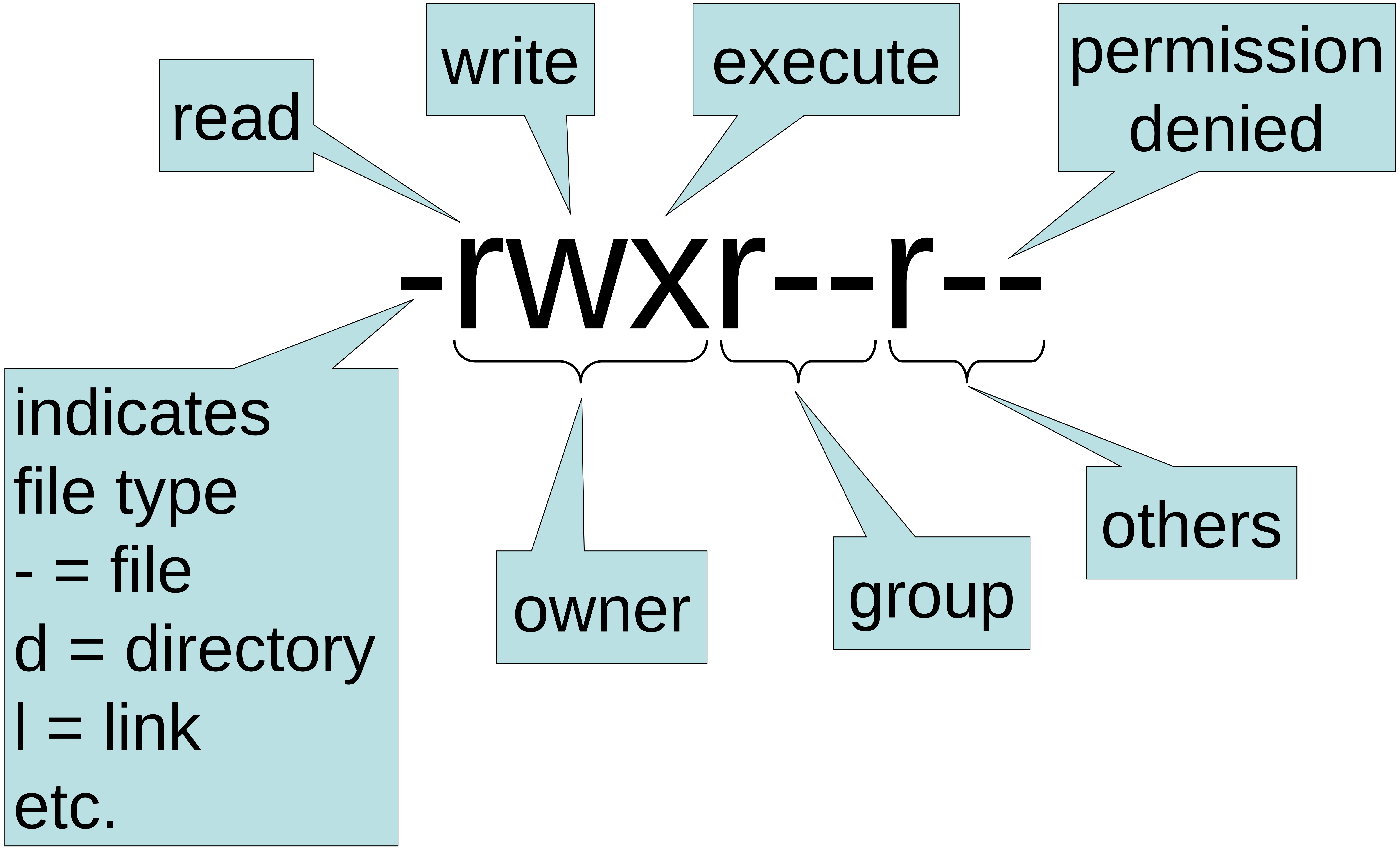
- Basic file management system provides
  - Directory structures for each I/O device
  - Tools to copy and move files
  - Information about each file in the system and the tools to access that information
  - Security mechanisms to protect files and control access

- Additional file management features
  - Backup, emergency retrieval and recovery
  - File compression
  - Transparent network file access
  - Journaling



Windows	UNIX/Linux	
dir	ls	List a directory of files or get information about files
copy	cp	Copy a file from one place to another
move	mv	Move a file from one place to another
del or erase	rm	Delete (remove) a file
type	cat	Type a file out to the screen (or redirected to a printer)
mkdir	mkdir	Attach a new subdirectory to the tree at this tree junction
rmdir	rmdir	Delete a subdirectory

Unix/Linux File Attributes - chmod



read

write

execute

permission denied

-rwxr--r--

indicates file type
- = file
d = directory
l = link
etc.

owner

group

others

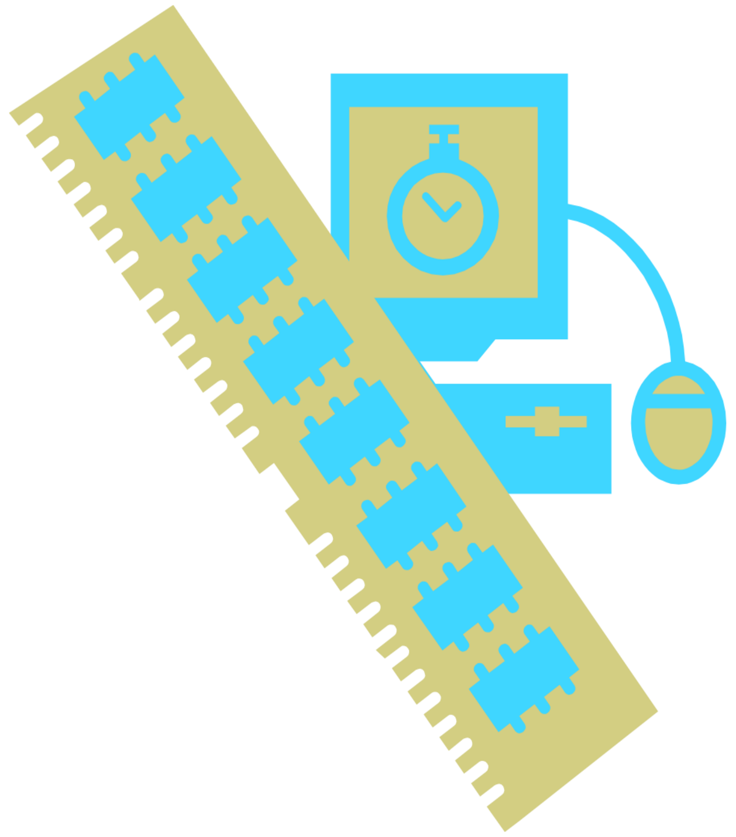
I/O Services Management

Startup configuration

- Device drivers that implement interrupts and provide other techniques for handling I/O
- Plug and play

Memory Management

- Keeps track of memory
  - Identifies programs loaded into memory
  - Amount of space each program uses
  - Available remaining space
  - Prevents programs from reading and writing memory outside of their allocated space
- Allocates memory to programs that are next to be loaded
- De-allocates a program's memory space upon program completion



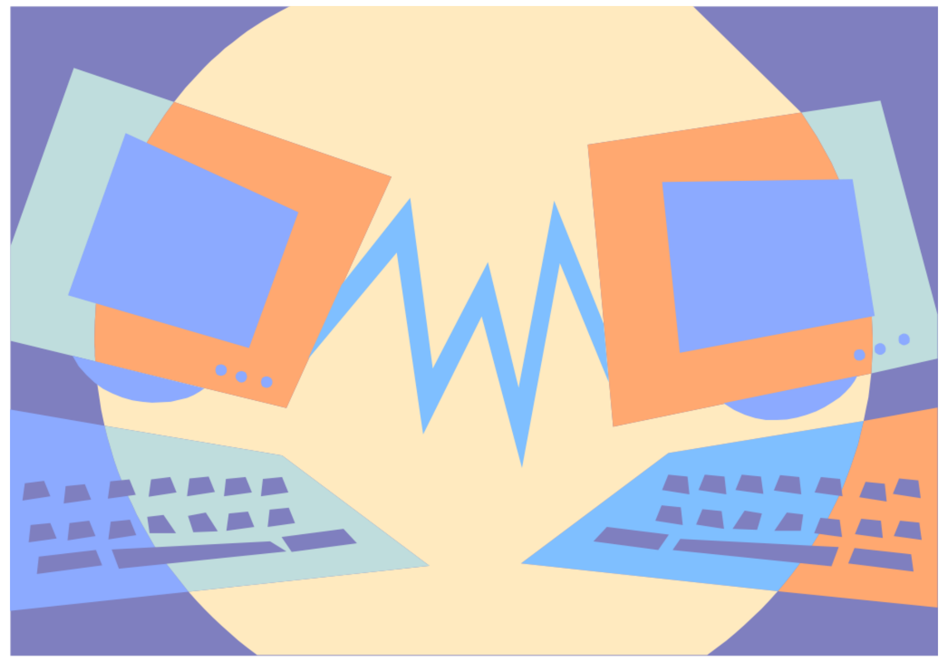
Secondary Storage and Security

- Secondary storage management
  - Optimises completion of I/O tasks for efficient disk usage
  - Combination of hardware and software

- Security and protection services
  - Protects OS from users
  - Protects users from other users
  - Prevents unauthorised entry to system
  - Prevents unauthorised system use by authorised users

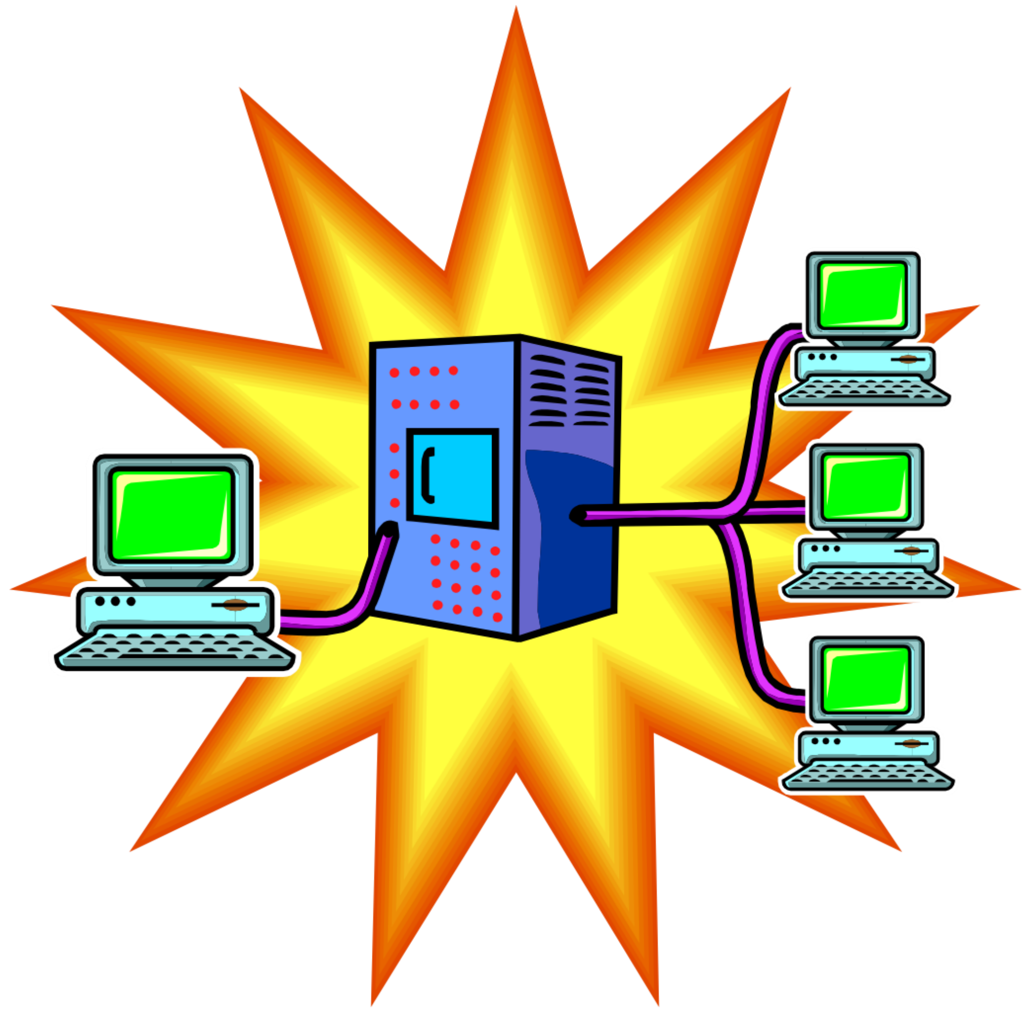
Network and Communication Services

- TCP-IP protocol suite
  - Locates and connects to other computers
  - Accesses files, I/O devices, and programs from remote systems
  - Supports distributed processing
- Network Applications
  - Email, remote login, Web services, streaming multimedia, voice over IP telephony, VPN
- Communication Services
  - Interface between communication software and OS I/O control system that provides network access



System Administration Support

- System configuration and setting group configuration policies
- Adding and deleting users
- Controlling and modifying user privileges
- System security
- File systems management



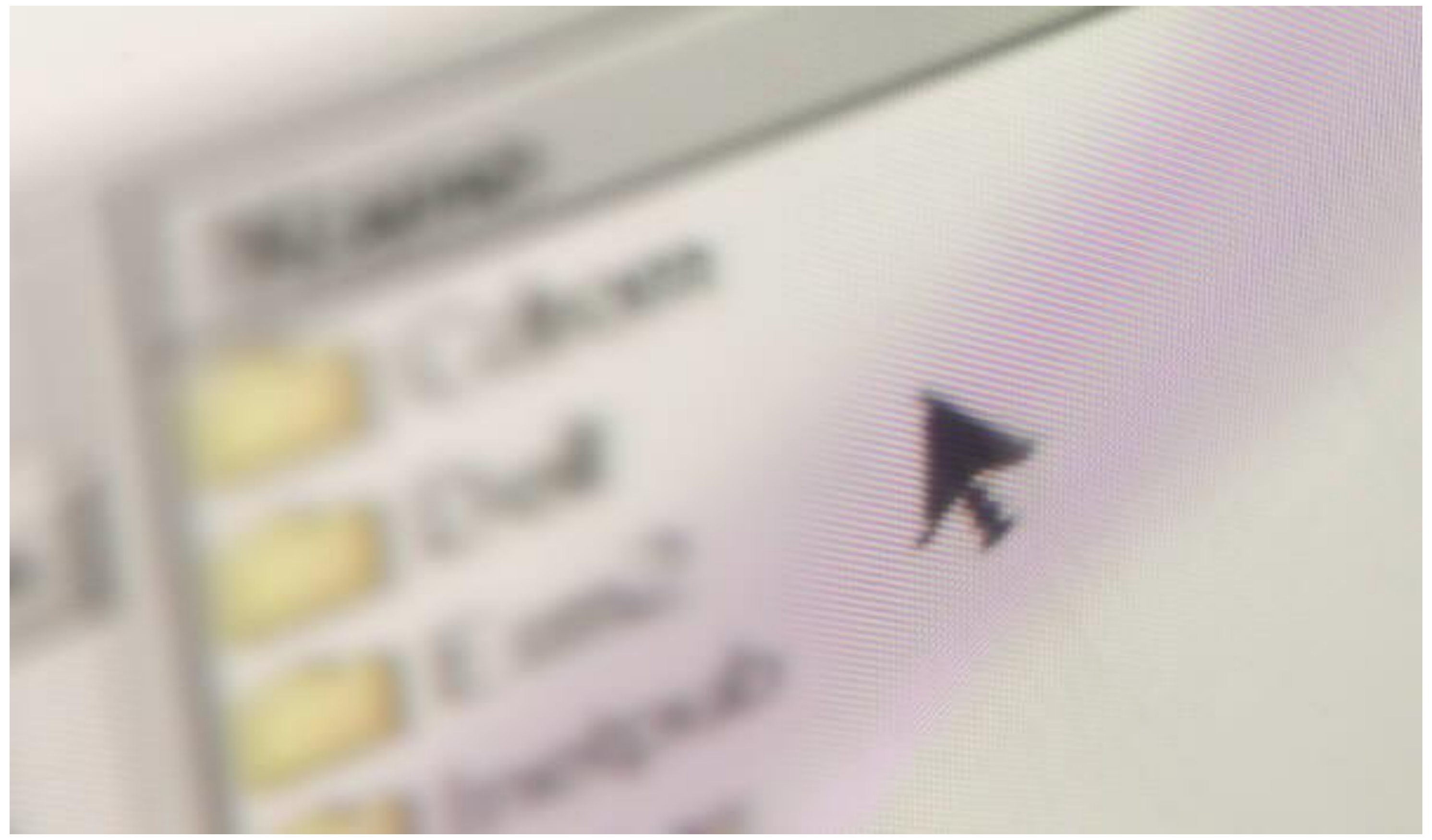
- Network administration
- Backups
- Software installations and upgrades
- OS installations (system generation), patches, and upgrades
- System tuning and optimisation
- Monitoring performance
- Recovering lost data

User Interface and Command Execution Services

- Types of user interfaces
  - CLI - Command Line Interface
  - GUI - Graphical User Interface
- Shell
  - User interface and command processor that interacts with the kernel
  - UNIX/Linux: C, Bash shells etc
  - Windows: command prompt, Powershell
- Command Languages
  - IBM Mainframes – JCL
  - MS Windows – BAT files, Powershell
  - Linux/UNIX – shell scripts

Graphical User Interfaces

- Mouse-driven and icon-based
- Windows
  - Are allocated to the use of a particular program or process
  - Contain desktop or screens, icons, windows, title bar, task bar, clock, menu bar, and gadgets or widgets



When are GUIs not enough?

- When one is not installed
  - on most Linux/Unix systems, the GUI is optional
  - the GUI is optional on Windows Server
- Limitation in functionality for advanced users
- GUIs require a lot more system resources
  - Icons
  - Fonts
  - Task bars

Graphical User Interface

- Advantages
  - Easy to learn and use
  - Little training
  - Amenable to multi-tasking
- Disadvantages
  - Harder to implement
  - Requires lots of memory
  - More hardware/software requirements
  - Software is complex and difficult to write

Examples of Windows and UNIX Command Line Syntax

Format: *CommandName (options) (arguments)*

- Each command is a sequence of ‘words’ separated by spaces

- An example from Linux: the *ls* command

```
ls
ls -al
ls d*
```

- An example from Windows: the *dir* command

```
dir
dir /s
dir /s Desktop
```

Note: spaces are very important!!

Pipes – Input and Output systems

Can be used to connect two programs

- the output of the first becomes the input to the second
- consider these examples:

```
dir c:\windows | more
dir c:\windows | find “log” /i | more
```

File system -Redirection

The standard input and / or the standard output to be bypassed

- consider these examples:

```
echo John is a good student
echo John is a good > file1.txt
echo John is a big student > file1.txt
echo John is a good >> file1.txt
```

File system -Redirection

- can be used to read input from a file
- consider these examples:

  sort

  sort < file1

- input and output redirection can be combined e.g.

  sort < file1 > file2

- redirection allows output to be captured for later use and programs which require input to be run unattended

Setting the Environment Variables

- used to store information about the environment in which a user's programs execute
- typically set during login
- can be listed using *set* command

- examples include

  path (Windows and Unix)

  username (Windows), user (Unix)

  home (Windows and Unix)

  tmp (Windows and Unix)

C Shell Commands – UNIX/Linux

- cp copy file(s)
- mv move (rename) file(s)
- rm remove (delete) file(s)
- ls list directory contains
- cat join (concatenate) files; display file
- chmod change file access rights
- pwdprint (display) working directory

- Search and sort tools
  - find search system for files
  - grep search files for text patterns
  - sort sort or merge files by rows
  - history (shows previous commands)

C Shell Commands –UNIX/Lunix

- echo display (text, arguments, variables)
- alias abbreviate series of commands

- System status and job control
  - date display date
  - finger display information about users
  - kill terminate a process
  - ps show process status
  - who show list of logged-on users

- Text and file manipulation
  - vi screen editor
  - more display text(file) one page at a time
  - wc count words, lines, characters
  - man on-line manual

Command Line Interface

- Advantages
  - Flexible - often more options, commands can be combined
  - Quick - if you do know the commands
  - Scriptable
  - Requires fewer resources
  - Remote display is easy

- Disadvantages
  - More difficult to learn and use
  - Obscure command syntax
  - A lack of feedback

Learning Objectives

On completion of this topic, you will be able to:

- Identify the components of a typical operating system
- Describe the basic functions of a typical operating system
- Distinguish between the different types of operating systems

Directed Reading

- The Architecture of Computer Hardware, Systems Software and Networking: An Information Technology Approach. 5th Edition – Irv Englander, 2013, Wiley, Chapter 15, 16, 17 and 18
- Computer Organization and Architecture: Designing for Performance. 10th Edition, W Stallings, 2016, Prentice Hall, Chapter 8