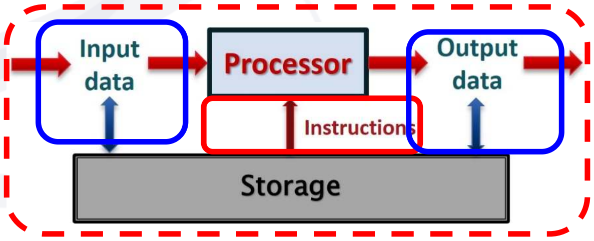
ENS1161 Computer Fundamentals

Module 9

Operating Systems

Moving forward..

- Last module:
  - I/O Software and communication
  - I/O modes
- Focus of this module:
  - Operating Systems



Module Objectives

On completion of this module, students should be able to:

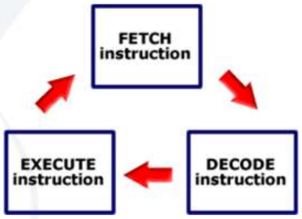
- Explain the role of the operating system in a computer system.
- List the main components of an operating system and describe their function and key operating principles.
- List and briefly describe the different types of operating systems.

Introduction

- **Module Scope**
  - The role of operating systems
  - Functions / components of an Operating System
  - Types of Operating System

How a Computer System Works *(recap – Module 1)*

- A processor needs a set of instructions
  - tells it what operations to perform on what data
- Instructions (programs) are stored in memory
  - Using only processors predefined *instruction set (Module 2)*
- The microprocessor:
  - *fetches* an instruction from memory
  - *decodes* it, and
  - *executes* the specified operation
- Sequence of *fetch*, *decode* and *execute* continues indefinitely
  - Until powered off



Types of software *(recap Module 1)*

- 2 broad categories:
  - **Operating system**
    - Main function is to control the hardware and enable other software to interface with the hardware
    - Also acts as 'control program' for other applications
      - e.g. Windows, macOS, Linux, Android
  - **Application software**
    - Designed to perform a certain type of function
      - E.g. wordprocessor, browser, spreadsheet, etc.

Using computing devices *(recap Module 5)*

- There are a wide variety of computing devices
  - From embedded systems in appliances, to mobile phones and laptops to desktop computers and servers.

Operating Systems – *interface and control*

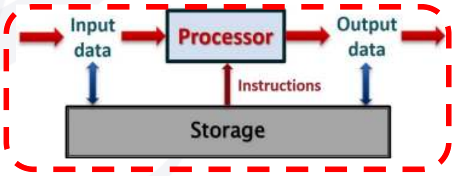
- We use these devices though a wide variety of **applications** (*apps*)

What is an Operating System?

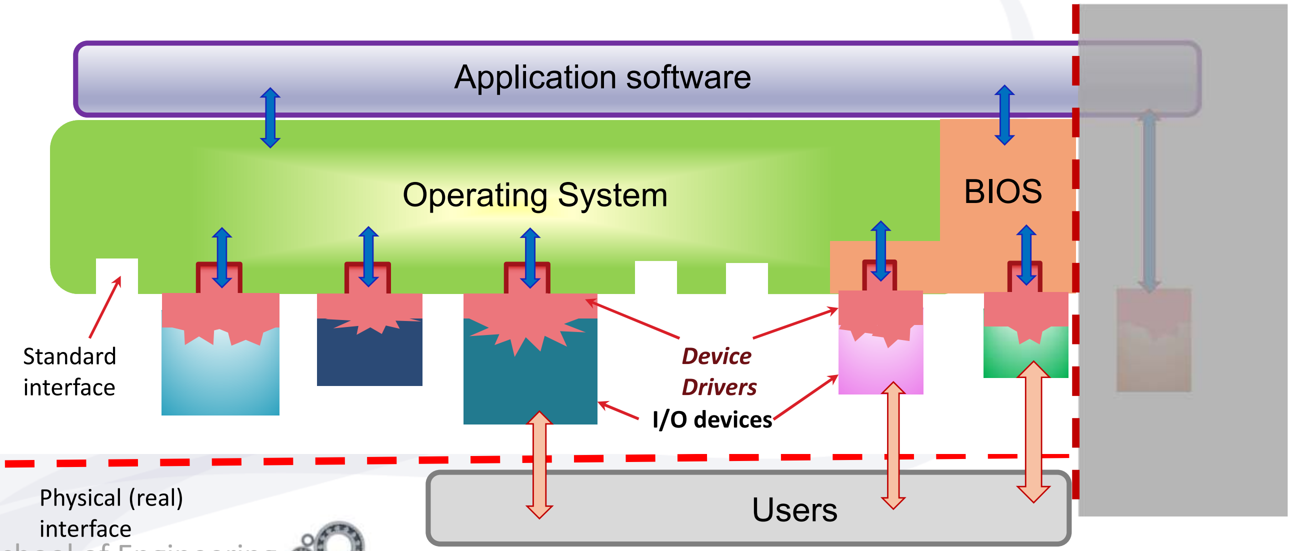
- Many definitions:
  - "Programs, implemented in software or firmware, that make the hardware usable"
    (Deitel , 1984)
  - Set of programs designed to coordinate the activities of a computer so that the most efficient use is made of its resources
  - Interface between the computer and the user
  - Software that manages the hardware and supplies services to application programs
- Essentially the software that controls the hardware and other software
  - The 'boss' software!

Why is an Operating System needed?

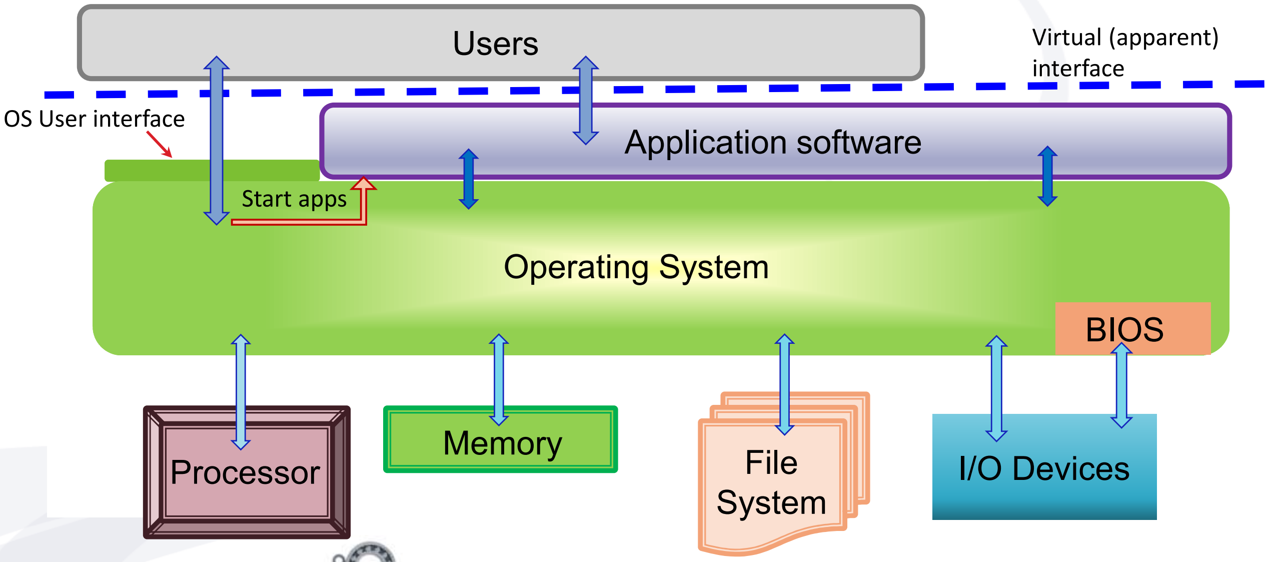
- The primary functions of an operating systems are:
  - Controlling and managing hardware
    - *Covered in Module 8*
  - Providing an interface
    - Application – Hardware interface
    - User Interface
  - Facilitating the running of software tasks
  - Allocation / management of resources
    - Processor, memory, secondary storage, I/O devices



Role of OS in I/O interfacing *(recap Module 8)*

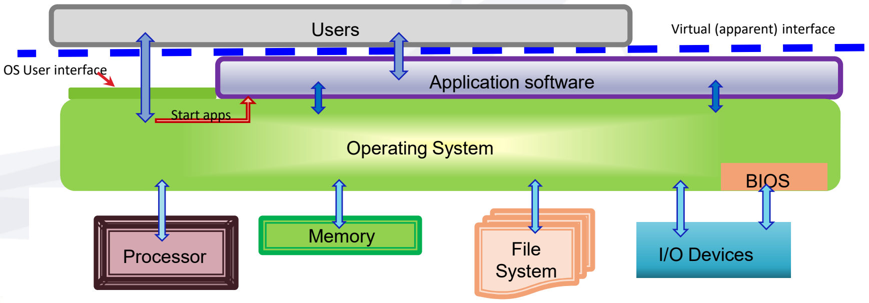


Role of OS in software interfacing

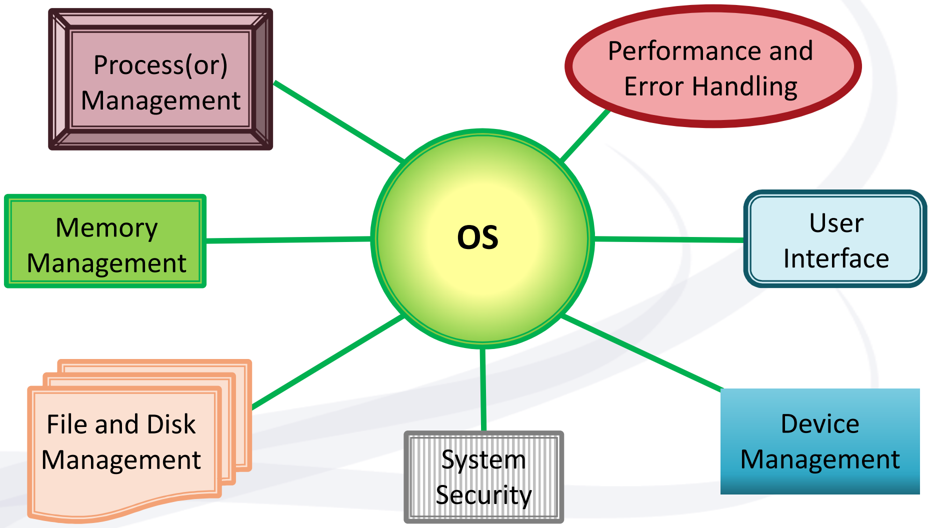


Key purpose of Operating Systems

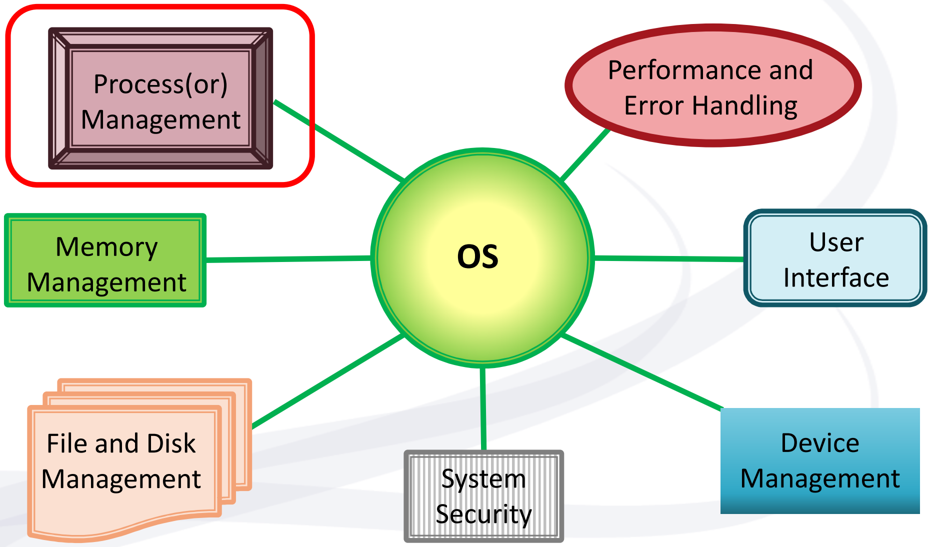
- To ‘abstract away’ low level hardware details
- Apps and users do not need to worry about details of actual hardware and how to manage that
- They use the OS, which has various parts to handle these tasks
  - E.g. memory management, file management, I/O management



Functions / components of an OS



Functions / components of an OS



Some terminology

- Program
  - A set of instructions that can be run
    - E.g. a program file on a disk
- Process
  - an instance of a program that is running in memory
    - Has its memory space, code, data, and other resources such as stack allocated by OS
- Thread
  - part of a process that can be executed independently

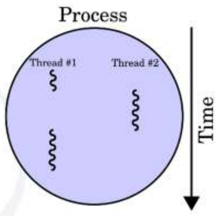


Image: Cburnett, 2007
https://commons.wikimedia.org/wiki/
File:Multithreaded_process.svg

Process Management

- Part of OS that controls which process gets to run on the processor and for how long
- Processes are kept in queues based on their status
- *Scheduler* will decide which process runs
  - Many different schemes to decide this
- Process may run until allocated time run out
  - Will go back to ready queue to allow other processes to run
- May be *blocked* because have to wait for some resource
  - E.g. waiting for I/O device to respond
- *Interrupts* often used to let OS know when an *event* process was waiting for has occurred
  - E.g. device ready

Example of process states

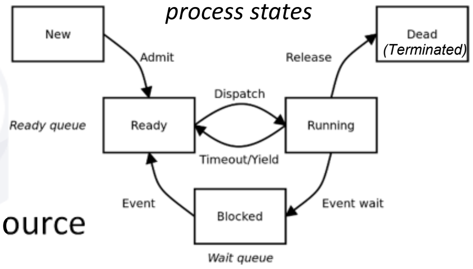


Image: MrDrBob, 2010
https://commons.wikimedia.org/wiki/File:Process_state.svg

Question 1

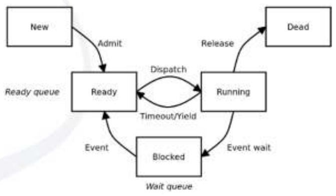
Why are blocked processes kept in a separate queue?

a. Because the blocked processes need to communicate with the I/O devices.

b. Because blocked processes are kicked out of the main memory

c. So that the scheduler does not need to consider them when working out which process runs next.

d. So that the blocked processes can rest.



Web Address: pollev.com/ens1161

Process Management

- Process management also includes tasks like:
  - Allowing new processes to start
  - Ensuring new processes get allocated required resources
  - Keeping track of processes (process ID)
  - Keeping track of process status
  - Managing inter-process communication (IPC)
  - Ensuring resources get released when a process is completed
  - Managing resource conflicts
    - E.g. *deadlock* situation
      - 2 processes can't continue because each needs a resource held by the other in order to proceed

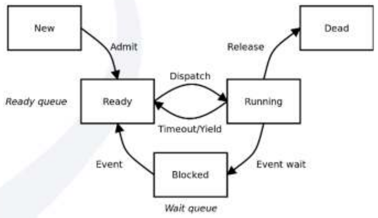
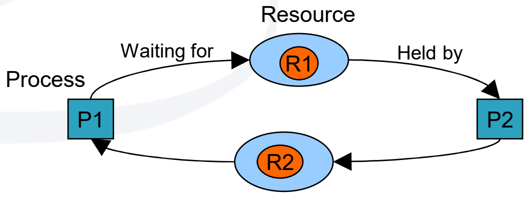
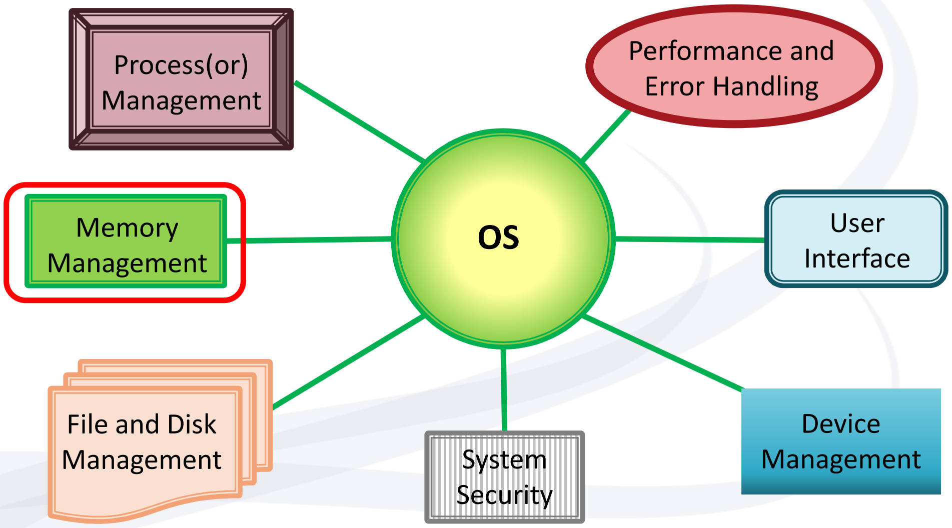


Image: MrDrBob, 2010
https://commons.wikimedia.org/wiki/File:Process_state.svg



Functions / components of an OS



OS

Process(or) Management

Performance and Error Handling

Memory Management

User Interface

File and Disk Management

System Security

Device Management

Memory Management

- To ensure that processes are allocated memory as required
  - Finite amount of physical memory
  - Lack of memory can stop process from running
  - There are different allocation schemes
    - *Beyond the scope of this unit*
  - Will only look at *virtual memory*

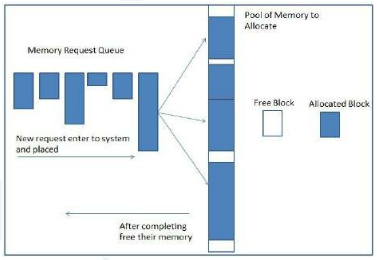

Pool of Memory to Allocate
Memory Request Queue
Free Block
Allocated Block
New request enter to system and placed
After completing free their memory

Image: Prabhudev Irabashetti
https://www.researchgate.net/figure/Memory-Allocation-Problem_fig1_265166374

- Other functions:
  - Translating addresses in processes to physical addresses in RAM
  - Ensuring that processes cannot access memory belonging to other processes
  - Managing shared memory

Virtual Memory

- Uses *virtual (logical) addresses* that don't correspond to physical memory locations
  - Memory Management Unit does translation of virtual to physical address
- Memory broken into blocks called *pages*
- If needed more RAM space, some pages copied (*swapped out)* to a secondary storage area known as *swap space*
- Pages *swapped in* (back to RAM) when required
- Advantage: Overcomes physical RAM space limitations
- Disadvantage: Swapping in/out takes time, and page translation also adds some overhead

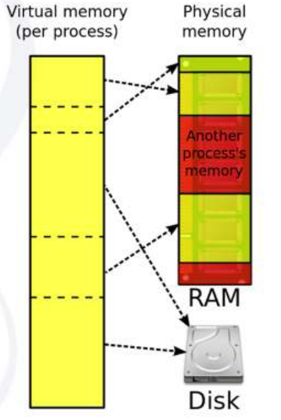
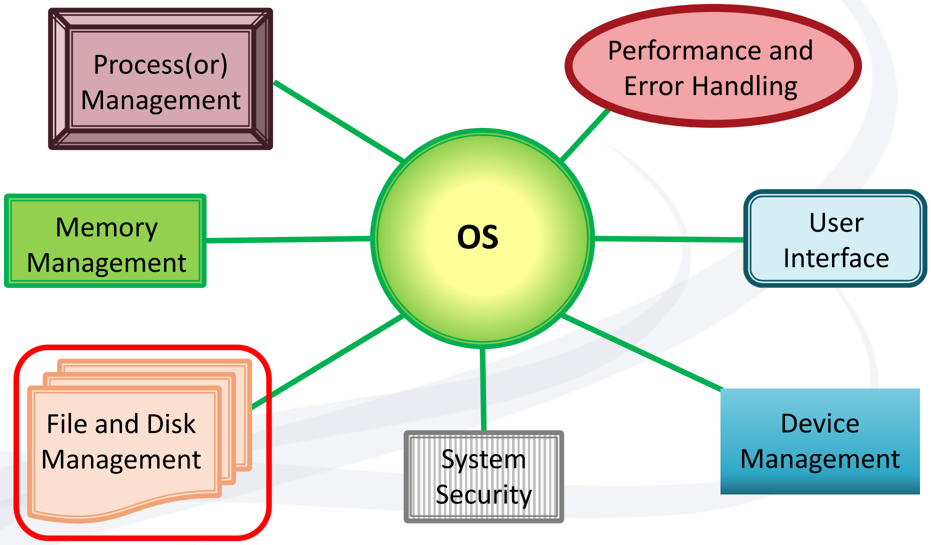


Image: Ehamberg, 2009
https://commons.wikimedia.org/wiki/File:Virtual_memory.svg

Functions / components of an OS



OS

Process(or) Management

Performance and Error Handling

Memory Management

User Interface

File and Disk Management

System Security

Device Management

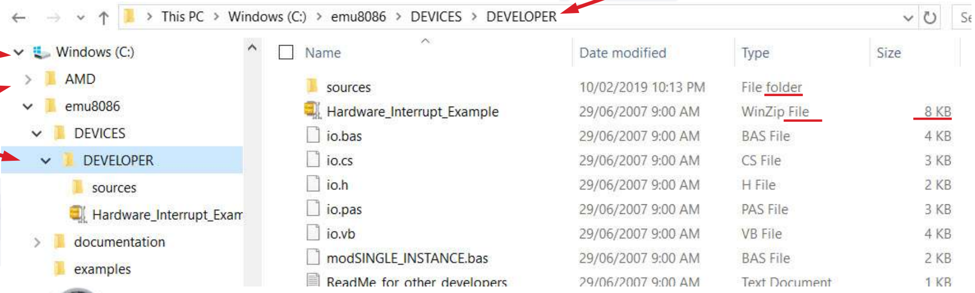
File and Disk Management

- **File:** named collection of data, normally resides in secondary storage
  - Some OS have an extension after last dot to indicate type of file
    - UNIX, DOS, Windows
- **Directories / Folders**
  - Hierarchical structure (logical) to help organise files

Hierarchical logical folder structure

Logical drive

Hierarchical logical folder structure



This PC > Windows (C:) > emu8086 > DEVICES > DEVELOPER

- Windows (C:)
  - AMD
  - emu8086
    - DEVICES
      - DEVELOPER
        - sources
        - Hardware_Interrupt_Exam
    - documentation
    - examples

Name	Date modified	Type	Size
sources	10/02/2019 10:13 PM	File folder	
Hardware_Interrupt_Example	29/06/2007 9:00 AM	WinZip File	8 KB
io.bas	29/06/2007 9:00 AM	BAS File	4 KB
io.cs	29/06/2007 9:00 AM	CS File	3 KB
io.h	29/06/2007 9:00 AM	H File	2 KB
io.pas	29/06/2007 9:00 AM	PAS File	3 KB
io.vb	29/06/2007 9:00 AM	VB File	4 KB
modSINGLE_INSTANCE.bas	29/06/2007 9:00 AM	BAS File	2 KB
ReadMe for other developers	29/06/2007 9:00 AM	Text Document	1 KB

File and Disk Management

- File Attributes
  - File size, Writeable / Read Only, etc.
- **File System**
  - Manages physical storage of data on secondary storage drives
  - Maps logical file structure to physical storage
  - A single file may be distributed over many blocks on a physical drive
    - *Refer Module 6*
  - Different OS may have different (incompatible) file systems

Example: File Allocation Table

Allocation table entry contains next block number

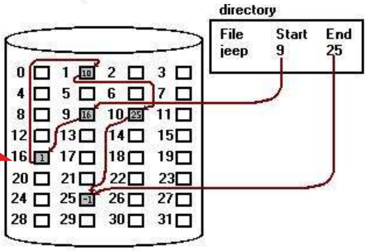
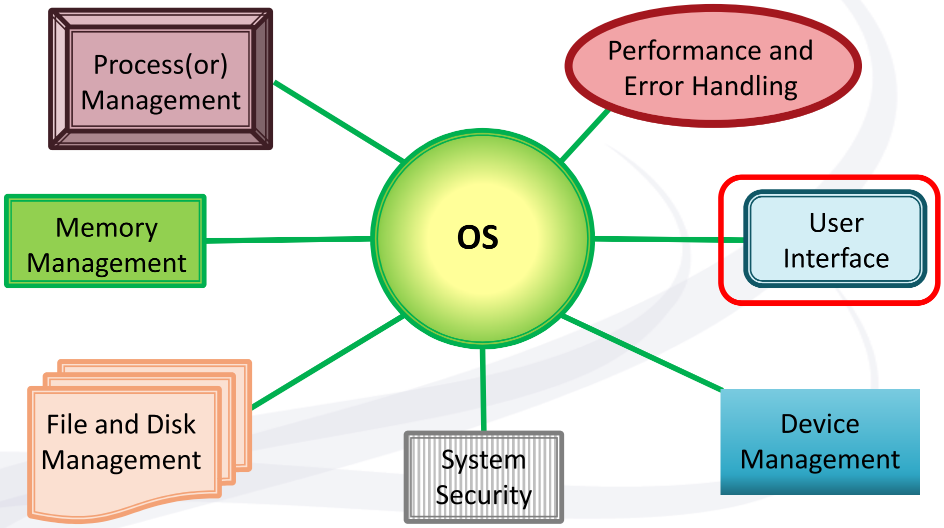


Image: Amarvashishth, 2016 https://commons.wikimedia.org/wiki/File:Linked_allocation_example.jpg

File and Disk Management

- **File System also manages:**
  - App / user requests for file open, save, etc.
    - Apps use *system calls* – run special OS functions
    - Users do this via utilities *(e.g. File Manager app)* or user interface (UI)
  - Access control (permissions) to files
    - Can a user / app a read a file, write to it, delete it, etc.
    - Linked to Security function of OS
- Disk management includes:
  - *Partitioning*: Dividing a physical drive into multiple logical drives
  - *Spanned volumes:* Combining multiple partitions / hard drives into a single logical drive

Functions / components of an OS



Process(or) Management

Performance and Error Handling

Memory Management

OS

User Interface

File and Disk Management

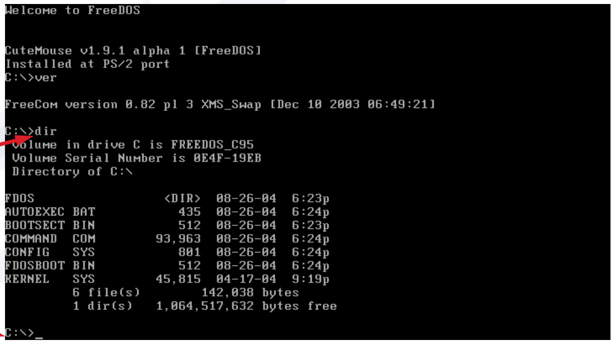
System Security

Device Management

Covered in Module 8

User Interface (UI)

- Part of the operating system that allows a user to enter and receive information
- **1. Text User Interface (TUI)**
  - Also know as *Command Line* interface or *shell*
  - Requires user to know commands and parameters
  - Text-based output
  - Mainly in older OS



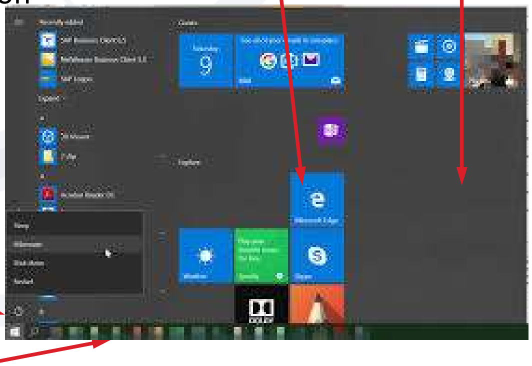
```
Welcome to FreeDOS


CuteMouse v1.9.1 alpha 1 [FreeDOS]
Installed at PS/2 port
C:\>ver

FreeCom version 0.82 pl 3 XMS_Swap [Dec 10 2003 06:49:21]

C:\>dir
 Volume in drive C is FREEDOS_C95
 Volume Serial Number is 0E4F-19EB
 Directory of C:\

FDOS                 <DIR>  08-26-04  6:23p
AUTOEXEC BAT           435  08-26-04  6:24p
BOOTSECT BIN           512  08-26-04  6:23p
COMMAND  COM        93,963  08-26-04  6:24p
CONFIG   SYS           801  08-26-04  6:24p
FDOSBOOT BIN           512  08-26-04  6:24p
KERNEL   SYS        45,815  04-17-04  9:19p
         6 file(s)         142,038 bytes
         1 dir(s)    1,064,517,632 bytes free

C:\>_
```

dir - command for directory listing

Command prompt and cursor for entry

User Interface (UI)

- **2. Graphical User Interface (GUI)**
  - Uses icons and menus
  - Pointing device used as main interface tool *(e.g. mouse)*
  - Requires more memory for graphical information
  - Most modern OS have GUI
  - Common components:
    - Start menu
    - Taskbar
    - Desktop
    - Shortcuts and icons



Shortcuts and icons

Desktop

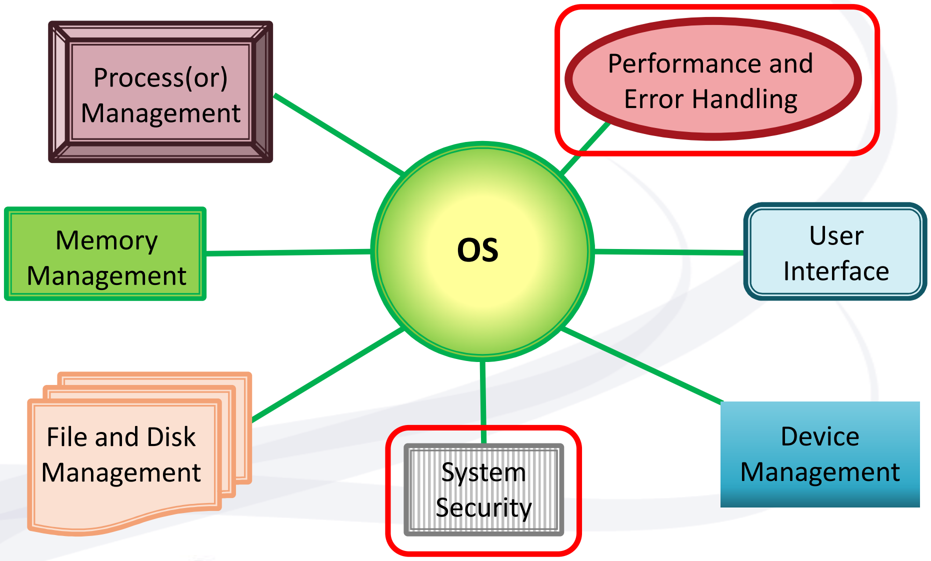
Start menu

Taskbar

User Interface (UI)

- **3. Natural User Interface (NUI)**
  - Uses more intuitive, natural human behaviour to interface
  - Newer OS, technology still developing for some types
  - Touchscreen
    - Most common – used by tablets, smartphones, some laptops
  - Voice commands and response
    - E.g. Siri, Alexa
  - Gesture recognition
  - Gaze-tracking
  - Brain-machine interfaces

Functions / components of an OS

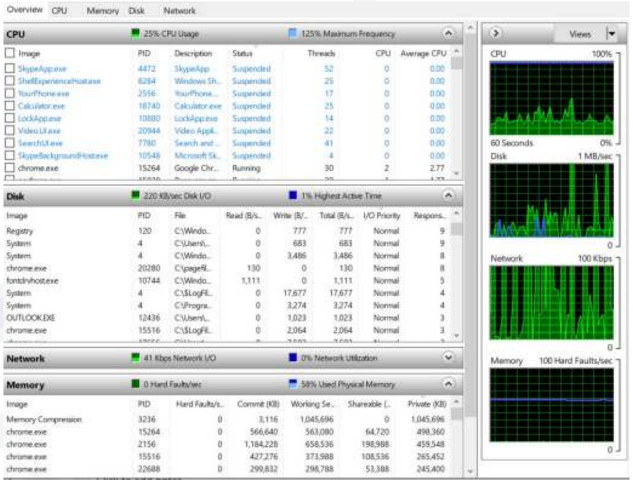


System Security

- Security must consider external environment of the system, and protect it from:
  - unauthorized access
  - malicious modification or destruction
  - accidental introduction of inconsistency
- User protection
  - Authentication, user roles, permission settings / access control
- Malicious process (*malware*) protection
  - Deliberately look for OS security weaknesses and exploit them
  - Lot harder to protect against
  - Often use special (3rd party) utilities to help with protection

Performance and Error Handling

- **System performance monitor**
  - Collects and reports key system performance indicators
  - Allows administrators to check how system is performing
  - Allows identification of bottlenecks or device issues



CPU — 25% CPU Usage — 125% Maximum Frequency

Image	PID	Description	Status	Threads	CPU	Average CPU
SkypeApp.exe	4472	SkypeApp	Suspended	52	0	0.00
ShellExperienceHost.exe	8284	Windows Sh...	Suspended	25	0	0.00
YourPhone.exe	2556	YourPhone...	Suspended	17	0	0.00
Calculator.exe	18740	Calculator.exe	Suspended	25	0	0.00
LockApp.exe	10880	LockApp.exe	Suspended	14	0	0.00
Video.UI.exe	20944	Video Appli...	Suspended	22	0	0.00
SearchUI.exe	7780	Search and ...	Suspended	41	0	0.00
SkypeBackgroundHost.exe	10548	Microsoft Sk...	Suspended	4	0	0.00
chrome.exe	15264	Google Chr...	Running	30	2	2.77

Disk — 220 KB/sec Disk I/O — 1% Highest Active Time

Image	PID	File	Read (B/s...	Write (B/...	Total (B/s...	I/O Priority	Respons...
Registry	120	C:\Windo...	0	777	777	Normal	9
System	4	C:\Users\...	0	683	683	Normal	9
System	4	C:\Windo...	0	3,486	3,486	Normal	8
chrome.exe	20280	C:\pagefil...	130	0	130	Normal	8
fontdrvhost.exe	10744	C:\Windo...	1,111	0	1,111	Normal	5
System	4	C:\$LogFil...	0	17,677	17,677	Normal	4
System	4	C:\Progra...	0	3,274	3,274	Normal	4
OUTLOOK.EXE	12436	C:\Users\...	0	1,023	1,023	Normal	3
chrome.exe	15516	C:\$LogFil...	0	2,064	2,064	Normal	3

Network — 41 Kbps Network I/O — 0% Network Utilization

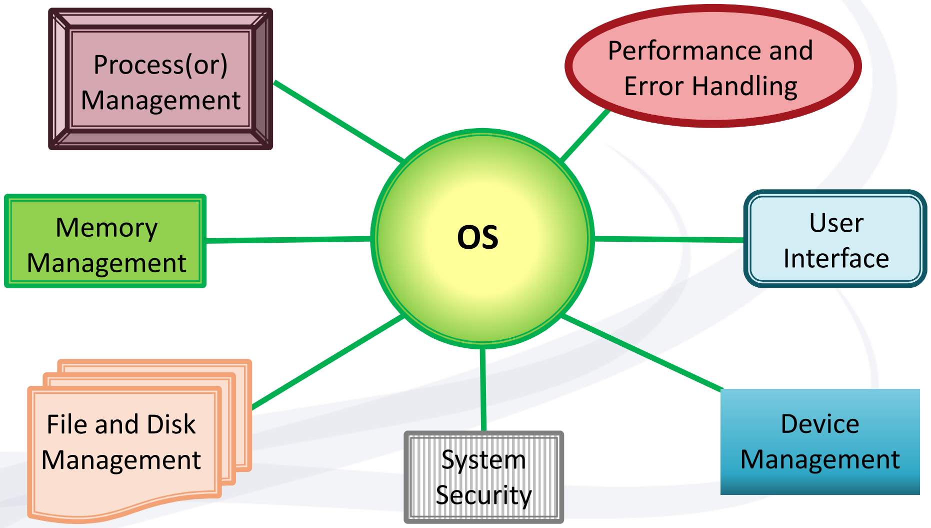
Memory — 0 Hard Faults/sec — 58% Used Physical Memory

Image	PID	Hard Faults/s...	Commit (KB)	Working Se...	Shareable (...	Private (KB)
Memory Compression	3236	0	3,116	1,045,696	0	1,045,696
chrome.exe	15264	0	566,640	563,080	64,720	498,360
chrome.exe	2156	0	1,184,228	658,536	198,988	459,548
chrome.exe	15516	0	427,276	373,988	108,536	265,452
chrome.exe	22688	0	299,832	298,788	53,388	245,400

Performance and Error Handling

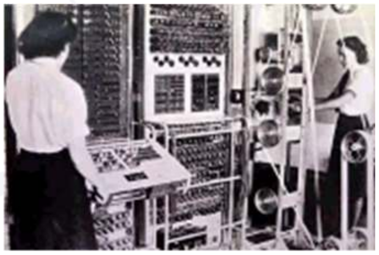
- **Error Handling**
  - Handles exceptions that may occur
    - Could be process related
    - Most commonly hardware related
  - Often handled via interrupts
  - Errors captured and logged for troubleshooting
  - System should handle (resolve) the exceptions
  - Otherwise should *degrade gracefully*
    - Maintain partial functioning even if parts inoperative
    - No catastrophic failure
      - E.g. controlled shutdown instead of just hanging

Functions / components of an OS



Types of Operating Systems

- There are different ways of classifying operating systems
  - based on the type of computers they manage and how they are used
- Following are some types of operating systems
  - These are NOT mutually exclusive
    - An operating system may fall under 2 or more types
- **Simple Batch OS**
  - multiprocessing of batch programs
  - few facilities for interaction or multi-access
  - generally use some form of JCL (job control language)
    - E.g. Early data processing systems, not used much now



Types of Operating Systems

- **Multi-access and timesharing OS**
  - multiple users (and processes)

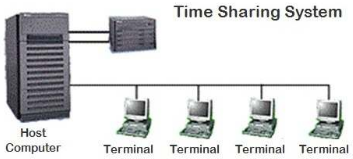


Image: Bp2010.heriadi, 2011 https://commons.wikimedia.org/wiki/File:TSS_Model.jpg

- **Single-tasking OS**
  - single-user, one program at a time
    - E.g. early PCs running DOS

- **Multitasking OS**
  - more than one process at a time
  - processor switches rapidly between processes
    - E.g. modern personal device OS – Windows, MacOS, Android

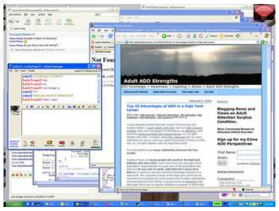
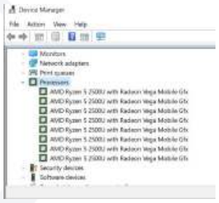


Image: Dennis Yang, 2006 https://www.flickr.com/photos/dennis/97827773/

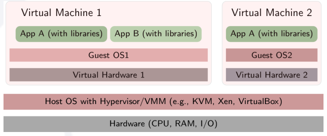
Types of Operating Systems

- **Multiprocessing OS**
  - more than one CPU, more than one process per CPU



Monitors
Network adapters
Print queues
Processors
AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx
Security devices
Software devices

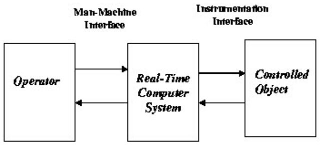
- **Virtual Machine(VM)**
  - makes single machine look like multiple machines



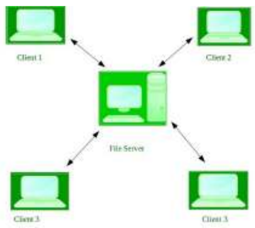
- **Distributed OS**
  - manages a group of distinct computers, makes them appear to be a single computer
    - *More in Modules 10/11*

Types of Operating Systems

- **Real-time OS**
  - capable of handling processes / requests in real time
  - Use for time-critical control systems
    - E.g. control of equipment like aircraft



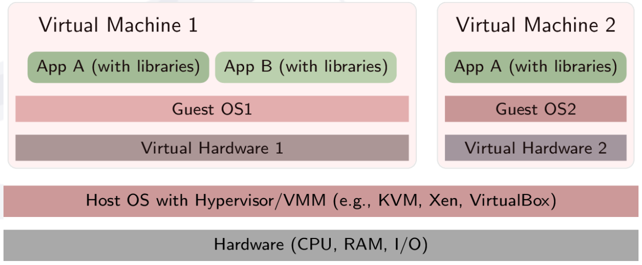
- **Network OS**
  - Run on servers
  - Gives server ability to manage data, users, network traffic, etc
    - *More in Module 10*



- **Mobile OS**
  - Designed to run on portable devices like tablets, phones, etc.
    - E.g. Android, iOS

Virtual Machines (VM)

- VM software creates multiple virtual machines on the one physical machine
  - Examples of VM software are VirtualBox and VMware
- Each VM can have its own OS as well as virtual hardware
- A *hypervisor* ensures each VM has access to the resources that it requires
  - also known as *virtual machine monitor- VMM*
- VMs are often used for:
  - Trying new OS
  - Testing apps in different environments
  - Running old apps that need specific settings or environments



Virtual Machine 1

App A (with libraries)

App B (with libraries)

Guest OS1

Virtual Hardware 1

Virtual Machine 2

App A (with libraries)

Guest OS2

Virtual Hardware 2

Host OS with Hypervisor/VMM (e.g., KVM, Xen, VirtualBox)

Hardware (CPU, RAM, I/O)

Image: Dr. Jens Lechtenbörger, DBIS Group, 2018
https://oer.gitlab.io/oer-on-oer-infrastructure/Docker.html#/sec-title-slide

Module Objectives

On completion of this module, students should be able to:

- Explain the role of the operating system in a computer system.
- List the main components of an operating system and describe their function and key operating principles.
- List and briefly describe the different types of operating systems.