

Operating System (OS) Intro

Last updated 6/3/22

OS Intro

- What is an Operating System
 - It's a Program
 - Starts before any user interaction is enabled
 - Controls access to hardware resources
 - Memory, A/D, SPI, ...
 - Schedules tasks for multiple “active” user programs
 - MS Word, Chrome, Outlook, ...
 - Schedules tasks for multiple “active” system programs
 - Hard drive access, user I/O, security

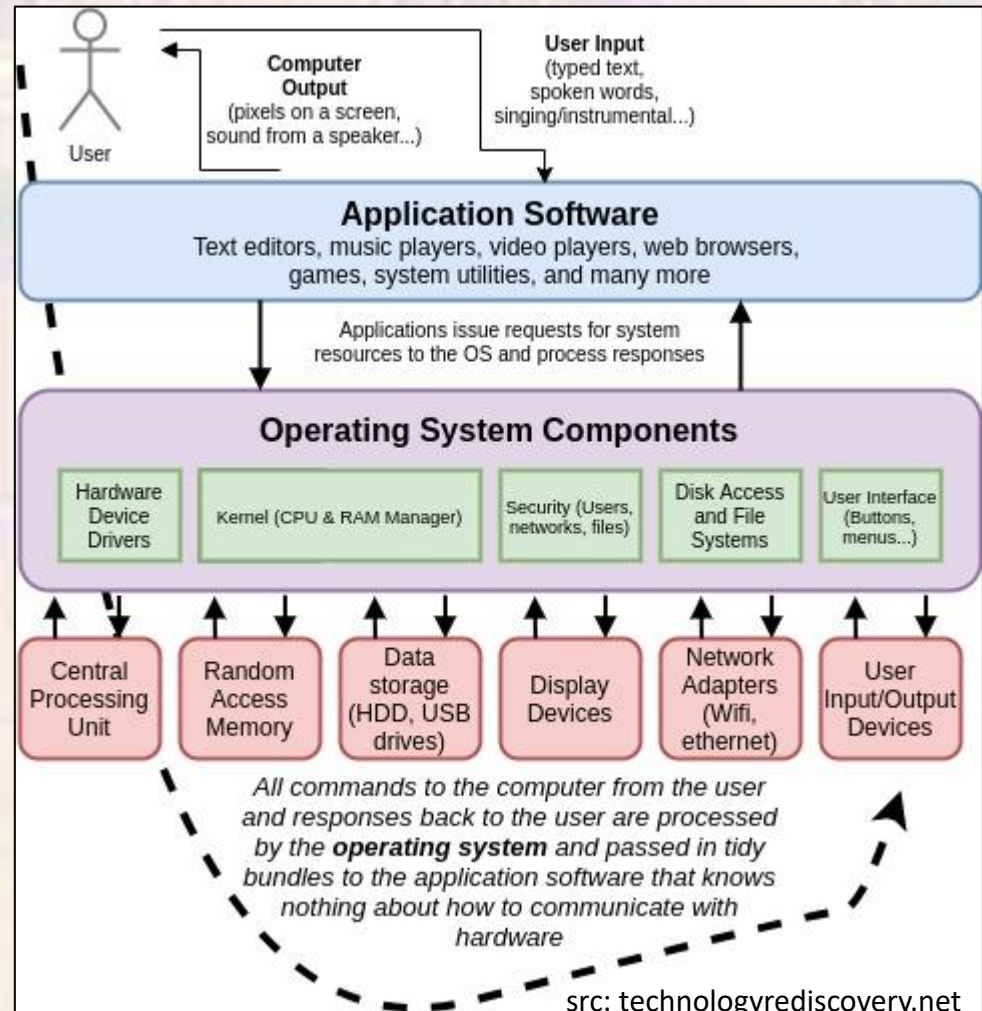
OS Intro

- Basic OS Functions

Coordination between other software
Coordination between users

Security
Control over system performance
Job accounting
Error detecting aids

Processor Management
Memory Management
Device Management (incl. I/O)
File Management



OS Intro

- Types of Operating Systems
 - Batch
 - Old school system – users submit jobs(programs), jobs are prioritized and run in order
 - Still used in super-computer environments
 - Maximizes processor utilization
 - Time-sharing
 - Multiple programs are running in parallel (not really)
 - Each program gets a slice of time on the processor
 - Because of the high speed – each program looks like it is running at the same time
 - Windows, OS-X, Android
 - Distributed / Networked
 - Multiple processors available (locally or in the cloud)
 - Programs are distributed by the OS

OS Intro

- Types of Operating Systems
 - Real Time (RTOS)
 - Provides specific Time or Delay responses
 - Required for systems that must react at a specific time or within a specific delay
 - Wireless communications
 - Requires the receiver to respond at a specific time interval
 - Robots, Industrial control, Automotive
 - Must carefully time actions to sensor inputs
 - Real time OS's are typically small
 - Makes them fast
 - Often used in non-real time situations to save memory

OS Intro

- Processes and Threads

- Process

- A program
 - Many processes can be “active”
 - Word, Email, Music, Clock ...
 - One process is “running” at a time (on one processor)
 - Start/Ready – waiting to be assigned
 - Running – actually running on the processor
 - Waiting – waiting for its next opportunity(turn) to run on the processor
 - Terminated – finished

- Thread

- One or more independent section(s) of code within a program
 - Spawned by a process (the program)
 - Threads share the process’s resources
 - Sometimes called lightweight processes since multiple threads use less resources than multiple processes

OS Intro

- Process Scheduling
 - All active processes put into a waiting queue
- Scheduling Policy
 - Round-Robin – each process takes a turn
 - FIFO – first process ready is the first executed (first in – first out)
 - Priority – highest priority process executes first
 - Combinations
- Blocked process
 - Process that cannot continue
 - Waiting on I/O from user or hardware
 - The process is put in waiting mode and back into the queue

OS Intro

- Process Scheduling
 - Process SWAP
 - Managed by the OS
 - Switching from one process to another
 - Context Switch
 - Managed by SW, HW, or both
 - Actually stopping one process and starting another in the HW
 - Current process
 - Save the current PC
 - Save any register values
 - Finish any memory writes
 - Next process
 - Restore that process's PC
 - Restore any register values

OS Intro

- Memory Management
 - Process Swap
 - If necessary – due to space limitations
 - Moves the current process memory from main memory to disk
 - Moves the new process memory from disk to main memory
 - Multiple processes in main memory
 - If multiple process fit
 - Makes sure each process knows where its memory is located
 - Prevents one process from entering another's space
 - More sophisticated memory management
 - Fragmentation support
 - Virtual memory
 - Segmentation

OS Intro

- Device Management
 - Device Drivers
 - Low level code to manage hardware on the processor
 - A/D, Timers, SPI, ...
 - Low level code to manage hardware attached to the processor
 - External memory, Accelerometer, Power management
 - Setup devices via Control Registers

```
/* Configure IO Direction mode (Input, Output, Alternate or Analog) */  
temp = GPIOx->MODER;  
temp &= ~(GPIO_MODER_MODER0 << (position * 2));  
temp |= ((GPIO_Init->Mode & GPIO_MODE) << (position * 2));  
GPIOx->MODER = temp;
```

OS Intro

- Device Management
 - Resource management
 - Prevent multiple processes from accessing a device while it is actively being used by another process
 - P1 starts an A/D conversion
 - While P1 waits, P2 gets swapped in
 - P2 want to use the A/D – but it is still working on the previous conversion
 - OS – makes P2 wait
 - Resource synchronization methods
 - Mutex
 - Queue
 - Semaphore
 - Mail