Introduction to RTOS (Part 1)

- Operating System is a piece of software that runs on computer or microcontroller that accomplishes some important functions:

General Purpose Operating System (GPOS) (Windows, Mac OS, Linux, iOS, Android)

① In charge of scheduling background tasks and user applications. OS figures out how to give slices of time to each of these processes for everything appears to be happening concurrently.

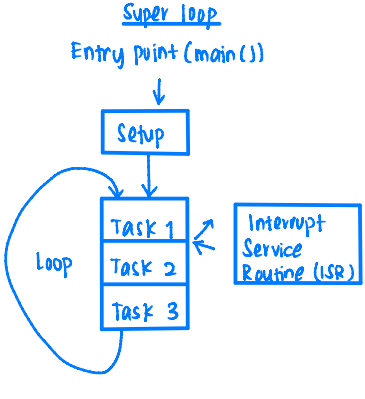
② Manages a number of virtual resources such as files, library and folders and allowing applications and processes to access them when needed.

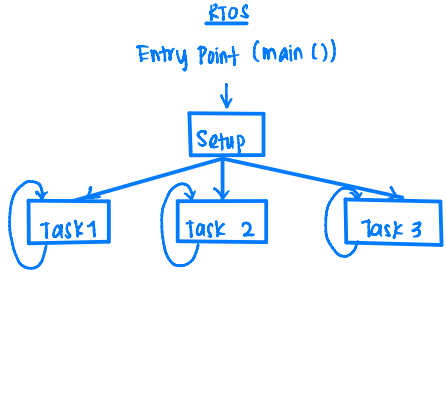
③ Manage or provide device drivers for the system. The drivers allow the system to read and write from an external disk respond to keyboard and mouse input or draw graphics on monitor.

- OS are designed with human interaction as the most important feature
- The scheduler is designed to prioritize such tasks which mean some timing deadlines can be missed or pushed back and a little lag in responsiveness.
- The scheduler is non-deterministic which human don't know which task will execute first.
- RTOS saves a strict timing deadline. RTOS are designed such that the scheduler can guarantee meeting timing deadlines on the tasks. RTOS might provide high level device drivers for microcontroller such as wi-fi, bluetooth stacks or lcd drivers.

Super Loop

- easy to implement, has very little overhead and great to accomplish a handful of tasks on your microcontroller.
- The super loop saves on cpu cycles and memory and easier to debug than using an RTOS.
- We can create interrupts to break the flow of execution to handle an external event such as push button.
- Interrupt Service Routine can be used if you want to meet strict timing deadlines for one or two tasks with precise timing.
- Unable to execute task concurrently and able to cause lag or missed data. To fix that, use the concept of running multiple tasks at same time on multi-core processor





Task : set of program instructions loaded in memory.

Thread : unit of CPU utilization with its own program encounter and stack.

Process : instance of computer program