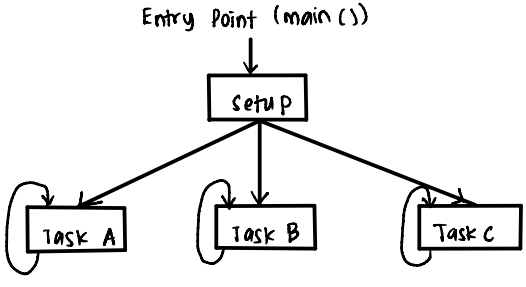
Introduction to RTOS (Task Scheduling)

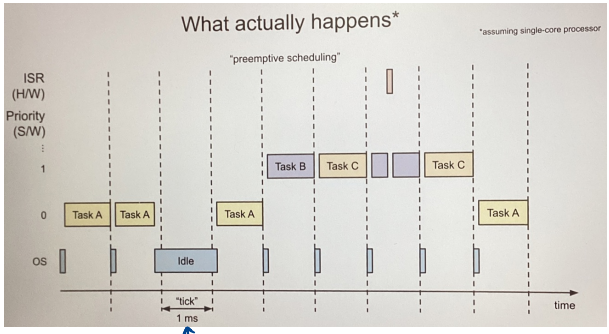
FreeRTOS : set priority task that allows the scheduler to preempt lower priority task with higher priority task.

Scheduler : A piece of software inside the operating system incharge of figuring out which task should run at each tick.



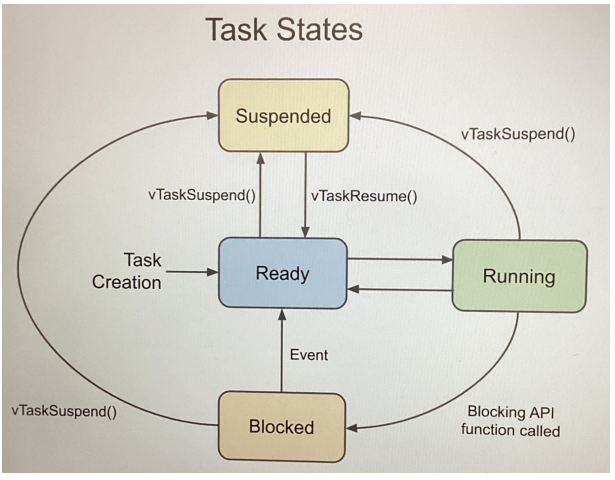
Interrupt Service Routine (ISR)

- Each task appear concurrently in its own loop
- ISR is used to handle hardware timer overflows, pin state changes or new communication on a bus. ISR can be set up in microcontroller to preempt any of the task to execute some code.
- CPU must divide task into time slices so that they can appear to run concurrently in the single core system.



default time slice

- Hardware timer is configured to interrupt every 1ms. The ISR for that timer runs the scheduler and chooses which task to run next.
- The task with the highest priority is chosen to run.
- If the highest priority task have the same priority, they are executed in a round-robin fashion.
- If the task with higher priority that the currently running task is in ready state, it will immediately run without waiting for next tick.
- Hardware interrupt always have a higher priority than any task running in software. Therefore, hardware ISR can interrupt any other task.
- ISR code can be kept short to reduce interruption to the running task.



- When the task is created, it enters the Ready state. (Telling scheduler that it ready to run).
- Scheduler chooses 1 task that is in ready state to run for each tick.
- While running, task that is in running state and can be returned to the ready state by the scheduler.
- vTask Delay() is the function that cause the task to wait by placing the task in Blocked state.
- The task is waiting for some other event to occur or waiting for some resources to be released by another task.
- Task in Blocked state allow other task to run instead.
- vTask Suspend() can put task in suspended mode (sleep). Any task can put any task into suspended mode.
- Task only return to Ready state by an explicit call to vTaskResume() by another task.