Singapore Polytechnic
School of Electrical and Electronics Engineering
ET0104 Embedded Computer Systems DECC 3FT/4EO

Tutorial 11 **Embedded Operating Systems and Multitasking**

1. OS is useful when the application needs standard routines like: Getting input from a user / Output to a display and/or a printer / Creating, reading and writing to files / Load, run and terminate a computer program

   The application might not need an OS when the hardware is too simple to support an OS or the developer does not wish to be bound by the rules/royalties incurred.

2. i) A task is the execution of a set of program instructions that are loaded into memory. A task could be either a process or thread. This is somewhat an abstract entity.

   A process is the execution of program instructions together with the collection of memory areas allocated to the process by OS and the *current* CPU states. A process is normally started by OS. (Refer to Figure 1 in notes)

   A thread is a separate execution path *within* a process. Each thread shares the same code, data, heap and memory areas but has its own stack area. Therefore local variables are private to each thread. (Refer to Figure 2)

   ii) Running state – task is executing.
   Ready state – task is waiting for its turn for execution.
   Block state – task is waiting for something to happen.

3. i) In multitasking environment, the program need not wait for events to happen. While a task is waiting for an event to happen, another task can be run. The program also appears to be more responsive by appearing to be running many different tasks at the same time.

   ii) The job of a scheduler is to transfer the resources (such as CPU and memories) from one task to another task, based on some algorithm, so that all the tasks *appear* to be running at the same time. Two common scheduling algorithms are First-in First-out (FIFO) and Round Robin.

4. Synchronization can be achieved by using synchronization objects such as mutex object, semaphore object and event object.

5. i)

Boot Options	Advantages / Disadvantages
Directly into ROM, use RAM for data	-Uses least amount of memory -Difficult to set up -Hardware dependent
i. User program on system memory map in ROM ii. Start OS on Flash Disk iii. OS runs program from ROM	- OS provides system functions - Some memory used by OS - Hardware dependent - May need OS license-but is low cost

i. User program on Flash Disk ii.Start operating system on Flash Disk iii.Operating system loads program into RAM - NOTE: similar to desktop	- Easy to set up and is flexible - Keeps two copies of program: On Flash Disk and RAM. - May need to license O/S

ii) Embedded Linux, ETS and Palm