an introduction to multi-threaded applications

ESE3025

multi-tasking operating systems (1)

- today, we take for granted the multi-tasking nature of operating systems
- for this introduction the term *task* refers to a *process* or *thread*
  - as we shall see, processes correspond to executing *programs*, while threads are implemented as *functions* within a program
- in the early days of the PC (circa 1980), desktop machines had single-tasking operating systems which:
  - could only run one program at a time
  - acted as a “shell” for a running program
  - handled various file management tasks

multi-tasking operating systems (2)

- a key feature of a multi-tasking operating system is *preemption*
- preemption simply means that the *operating system can switch from one task to another at any time*
- when execution switches from one task to another, we call this a *context switch*
- if an operating system is cooperatively preemptive, it means that the task decides when it can allow a context switch
- if an operating system is fully preemptive, it means that the task can be interrupted at any time by the operating system *kernel* or *scheduler*

multi-tasking operating systems (3)

- what does multi-tasking application look like?
- note to instructor: open multi-tasking_app_preview and describe its execution using a timing diagram live for the students