Lecture 1

- Single vs. Multi
- Threading/Tasking
L1: Tasks/Processes, Threads

• Program running: task or process
  – Resources allocated by the Operating System: process stack, memory address space, registers, program counter, I/O ports, network connections, file descriptors, etc.

• CPU: a resource used by the tasks
  – Given to a task for a short while, then to another task,...
  – Multitasking: CPU passed between tasks quickly enough.
    Simultaneous execution?
  – Persistence vs. Execution time

• States: running (actively executing), suspended (a.k.a. ready or ready waiting), blocked (a.k.a. waiting).
  – Having to share resources is what makes tasks move from one state to the other.
L1: Tasks/Processes, Threads

• Scheduling: when and to which task should the CPU and other resources be given?
  – Priority, time sharing, round robin, etc.
  – Preemption: return control to the OS, each process is given a slice of time.
• Task context: registers, program counter, stack, privileges, etc.
  – Switching context: state of currently active task needs to be saved so it can be restored later on.
L1: Tasks/Processes, Threads

- Thread (Light-weight process): smallest subset of resources necessary for the execution of a program (copy of registers, program counter, stack).
- Threads can also be running, suspended, waiting.
- Thread of execution: sequential execution of a set of instructions.
- Multiple threads
  - Smaller jobs with their own thread of execution
  - Not independent from each other (within a single process).
  - Advantage?
    Context switch between threads: simpler/faster
## L1: Tasks/Processes, Threads

<table>
<thead>
<tr>
<th>Single process – single thread</th>
<th>Single process – multiple threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiprocess – single thread</td>
<td>Multiprocess – multiple threads</td>
</tr>
</tbody>
</table>

- **Resources shared among threads**
  - The code
  - Data
  - CPU and associated physical registers
- **Proprietary to each thread**
  - A stack
  - Status information
L1: Operating System (OS), Real-Time OS

• The OS must provide the following functions:
  – Scheduling: which task will run and when.
  – Dispatching: necessary operations to start the tasks
  – Inter-task communication and synchronization: mechanisms for exchanging data and information between tasks.

• Kernel: smallest portion of operating system that provides these functions.

• RTOS: ensures that time constraints can be met.
  – Is it enough to “respond quickly”?
  – Key: deterministic behavior.
Assignment (class / laboratory)

• Read Chapter 11, sections 11.0 – 11.10.2

• Read the following documents before going to the lab:
  – Lab Guidelines
  – Lab How Tos
  – Lab 1 assignment guide

• You can find the documents on Blackboard, under Course Content -> Laboratory