

Processes

Questions answered in this lecture:

What is a process?

How does the dispatcher context-switch between processes?

How does the OS create a new process?

What is a Process?

Process: An **execution stream** in the context of a **process state**

Execution stream

- Stream of executing instructions
- Running piece of code
- Sequential sequence of instructions
- “thread of control”

Process state

- Everything that the running code can affect or be affected by
- Registers
 - General-purpose, floating point, status, program counter, stack pointer
- Address space
 - Everything process can address through memory
 - Represented by array of bytes
 - Heap, stack, and code

Processes vs. Programs

A process is different than a program

- Program: Static code and static data
- Process: Dynamic instance of code and data

No one-to-one mapping between programs and processes

- Can have multiple processes of the same program
Example: many users can run “ls” at the same time
- One program can invoke multiple processes
Example: “make” runs many processes to accomplish its work