

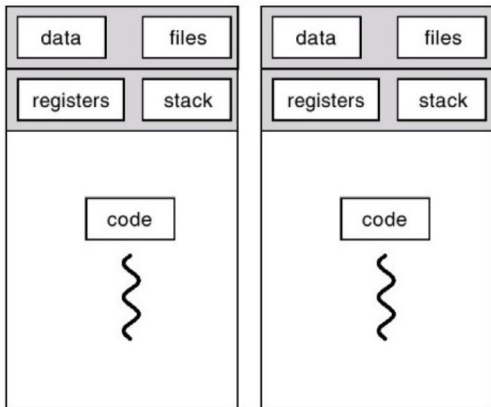
# Multiprocessing

CPSC 328 - Network Programming

# Create a New Process

- `fork()` system call
- Copies existing process
- Two return values – one for each process
  - parent: PID of child
  - child: 0
- Conceptually, the data, code, stack, and heap are copied
  - OS implementation may optimize some of this

# Multiple Processes



# File Descriptors

- The child process inherits the parent's file descriptor table
- Shared file offsets

# Ending Processes

- Orphan process
- Zombie process

# Pointers with `fork()`

- Parent's address space is copied
- OS implementation of the heap is irrelevant

# exec() System Calls

- Replace an existing process
- `execl()`: `l` is a list of arguments
- `execv()`: `v` is a vector (array) of arguments
- `execle()`: `e` is an array of environment variables
- `execlp()`: `p` is a path
- `execvp()`
- `execvpe()`