

# Operating Systems

## COMS W4118

### Lecture 6

Alexander Roth

2015 – 02 – 10

## 1 General Overview

- Mechanisms for two different processes to communicate (Interprocess communication).

## 2 File Sharing

- You can have two file descriptors share a file by calling `dup(1)`.
- When a parent process forks and create a child process, all the file descriptors from the parent process get `duped` into the child's file descriptor table.

## 3 Pipes

- `pipe` is a system call and you pass an array of two integers to it.
- Arguments are `int *fd`.
- When called, the kernel will create a data structure called a `pipe`, that will connect to the two file descriptors together.
- `fd[1]` will be the writing end of the pipe.
- `fd[0]` will be the reading end of the pipe.
- The file descriptor numbers will not be 0 or 1. They will be whatever was available next.
- After you call `pipe`, you immediately `fork` to copy the descriptors into a child process.
- This is a half-duplex connection.

- Whichever file descriptor reads first, then the data will be removed from the pipe.
- You must determine what kind of communication you desire.
- `STDIN_FILENO` is 1.
- `dup(2)` copies a file descriptor into another given file descriptor.
- When using `dup(2)` check that the descriptor you want to copy over another value, you will want to check that the descriptor is not the same number.
- `NULL` is sent when there is only one file descriptor open.
- Write a program that is modeled like the example, but takes another program name, have it run, and have the output fed into the pager.
- `pipe` is the oldest and simplest interprocess method; however, it is one way and only related processes can use it.

## 4 Other Interprocess Systems

- There are three other ways to control interprocess communication.
  1. Message queues
  2. Semaphores
  3. Shared memory
- All are terribly implemented.
- Hard to clean-up because there is no reference counting.
- POSIX refers to the modern Linux standard.

### 4.1 Shared Memory

- Gives you the ability to share a piece of memory (like an array) between two processes.

### 4.2 Message Queue

- Like Socket, handles communications between processes.
- Use single-domain-host sockets instead of message queues as they are easier to use.

### 4.3 Semaphores

- Synchronize actions across processes

## 5 Memory Mapping

- When you compile a program into an executable, it uses a shared object that controls the standard library.
- A static link is a `.o` file.
- A dynamic link holds a link to some larger library to load the code in at run time.
- The advantage is that you have one copy of standard library on your system. and all programs will load the necessary parts at run time.
- This is done by memory-mapping.
- Used for writing and reading to file and also for executing code.
- You call `mmap` when you want the starting address of a mapped region.  
**addr** Usually pass `NULL`, the specific area of the memory.  
**len** Length of the memory  
**fd** File descriptor on where to map.  
**offset** The offset of the file.
- If you want to zero out a file, you read from `/dev/zero`.
- You use `mmap` to create a shared reason by mapping `/dev/zero/` to it to zero out the region.
- `mmap` can be used for unrelated tasks as long as there is a file underneath.