



Pipes and Redirections

CIS 3800 Recitation Sep 19th



File Descriptor Refresh

- File descriptors are integers which represent files
- Every process has its own file descriptor table managed by the OS
- A child created by fork copies its parents file descriptor
- Execve clears the file descriptors (but dup2 redirections are still maintained)
- Stdin, stdout, stderr is 0, 1, 2
- File descriptors get incremented from there

```
#include <fcntl.h>

int open(const char *pathname, int flags);
int open(const char *pathname, int flags, mode_t mode);
```



Redirection

- Uses dup2 system call
- Changes the oldfd to now be the newfd
- Preserved across execve
- Useful for tricking process into thinking it is writing or reading from STDIN or STDOUT without making any changes to the code

```
int dup2(int oldfd, int newfd);
```



Pipes

- A unidirectional data channel that can be used for interprocess communication
- Addressed as a file descriptor, just like any file, socket, etc
- Can be named or unnamed
- Run *pipe* on an array of size 2
- Pipefd[0] becomes read, pipefd[1] becomes write
- **PIPES NEED TO BE CLOSED!!!!** (at least the write end)

```
#include <unistd.h>
```

```
int pipe(int pipefd[2]);
```

Code examples

