CS 208

Mon, 15 May 2023
```c
fd = open("elk.txt", O_RDONLY);

dup2(fd, 0);
close(Fd);
```
parent

fork

wait

fgets(command)...

child

parse command

fd = open("out.txt")...

dup

close

exec

Command: ls -l > out.txt

ls -l > out.txt command
```
int fd[2];
pipe(fd);

char buf[30];
write(fd[1], "hi", 2);
read(fd[0], buf, 2);
```

Dumb. Don't do this.
int fd[2];
pipe(fd);
Fork()
Want

parent writes (stdout)

to write-end of pipe

child reads from (stdin)

dead-end
Digression...

When a process writes to a pipe and fills it up, the process goes to sleep until some bytes get read from the pipe.

When process reads from empty pipe, sleeps until there's enough data to finish the read.
then... exec in both children