CS 208

15 Feb 2023
Wed
char *argv[]

ls -l myFolder
char *argv[2]  ls -l tmp

0. Read user input

input: `ls -l -temp`  

2. Insert `\0`'s

3. Build `argv

```
argv[0] 1  2  3
```

argv[1]  NULL
char*argv[7]    ls -l tmp

1) Read user input
   input ls -l tmp

2) Insert \n's
   ls \n -l \n tmp \n
3) Build argv
   argv[0] 1 2 3

① fgets
② Count the # of CL args (argc)
③ Allocate char*argv[argc+1]
④ Insert \n's
   & set argv pointers
   fork+
⑤ execvp
File descriptors

0   \(\text{(stdin)}\)  keyboard
1   \(\text{stdout}\)
2   \(\text{stderr}\)
3   \(\text{stuff.txt}\)

```c
    int fd = open("stuff.txt", [writing, ]);
```
File descriptors

0: READ
1: WRITE
2: WRITE
3: WRITE

(stdin)
keyboard

(terminal)
Stderr

(stuff.txt)

int fd = open("stuff.txt", [writing]);

To redirect
Stdout to stuff.txt
```c
fd = 3

dup2(fd, STDOUT, ....)

close(fd)
```