



# System Calls and `unistd.h`

CS 2130: Computer Systems and Organization 1  
December 3, 2025

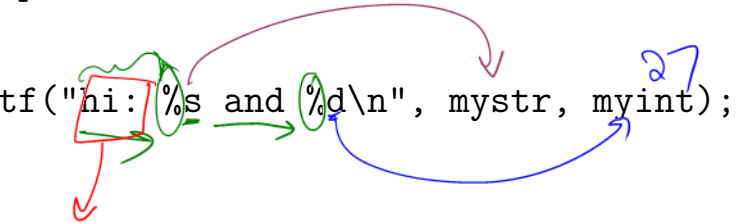
# Announcements

- Homework 10 due Monday
- Final exam: 7pm Dec 12, Wilson 301 (different room!)
  - Cumulative, see practice tests
  - Exam conflict form in email

# printf

```
int printf(const char *format, ...);  
int fprintf(FILE *stream, const char *format, ...);
```

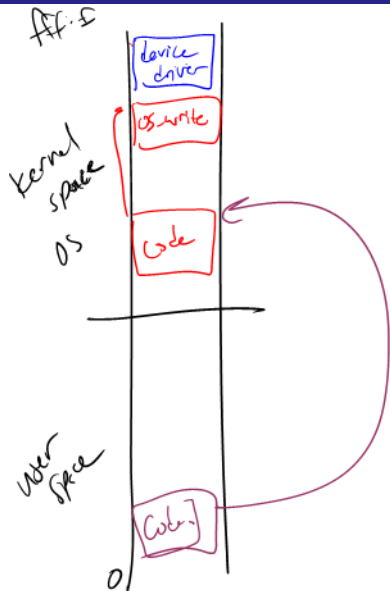
`printf("hi: %s and %d\n", mystr, myint);`



hi: — and 27 ✓

now (str → FILE\*)

# Syscalls



Write:

- arg check
- syscall ←
- ret value check
- ret

# write

`write:`

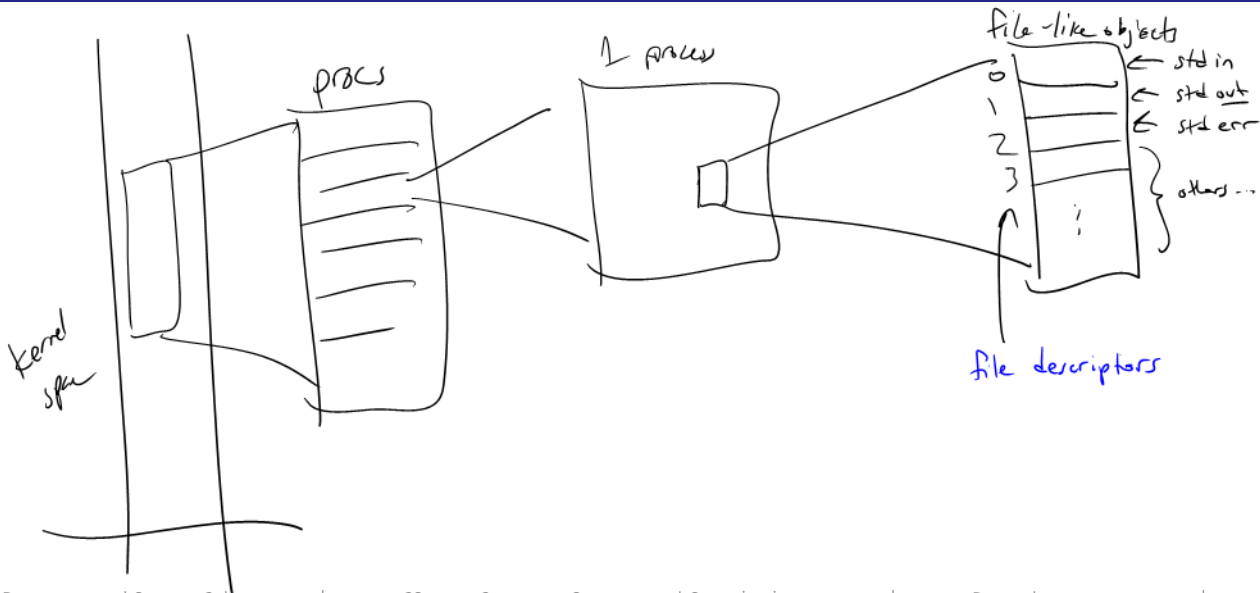
- Argument checking
- `syscall`
- Return value checking
- `ret`

# Processes

Process - approximately what we think of as a “running program”

- Operating System effectively has a giant array of processes started since computer turned on
- Try `ps -A`
- Has access to all memory (but only its own!)
- Operating System maintains data structure about each process
  - What program is running, who ran it, when it started, ...
  - Array of “file like objects”

# Processes



# Using `write`



# Using `write`

# Using `write`