Processes and System Calls

CS 2130: Computer Systems and Organization 1
April 20, 2026

Announcements

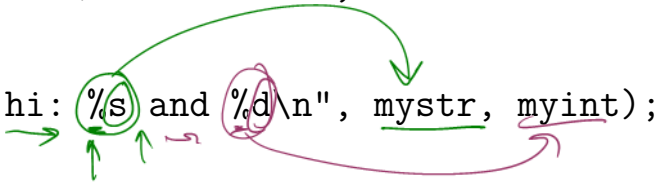
- Homework 10 due next Monday
- Lab ~~12~~ 14 tomorrow
- Final exam: 7-10pm April 30, Gilmer 301 (different room!)
  - Cumulative, see practice tests
  - Exam conflict form in email

Could you write `printf`?

printf

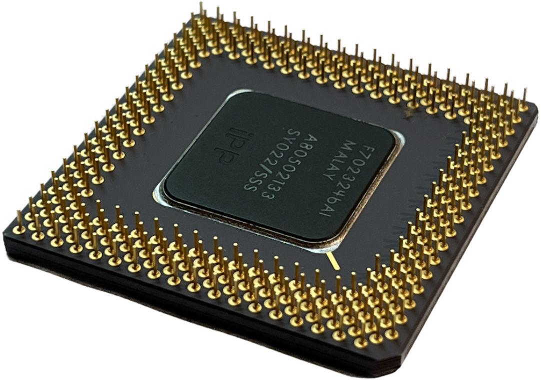
```
int printf(const char *format, ...);
int fprintf(FILE *stream, const char *format, ...);
```

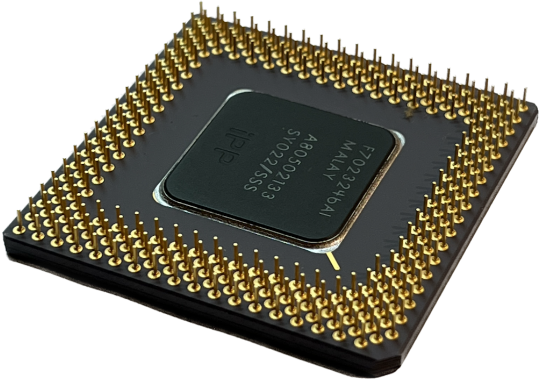
```
printf("hi: %s and %d\n", mystr, myint);
```

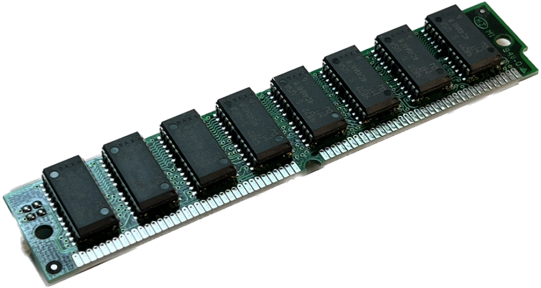


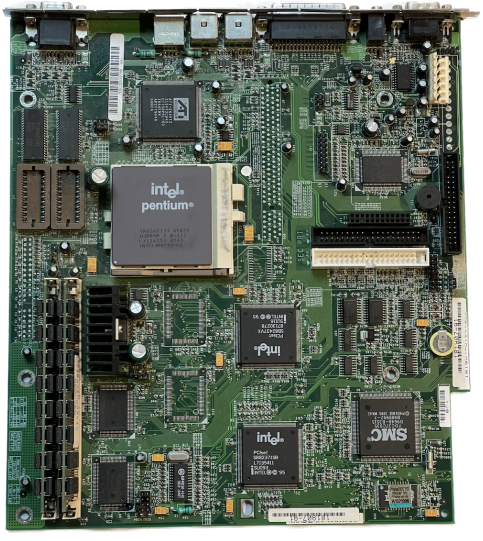
Move: str → FILE*

Backing up…




F7023246AI
MALAY
A80502133
SY022/SSS
iPP




intel
pentium
intel
intel
SMC

Syscalls


```
write:
    [ arg check
    syscall
    [ ret value checks
    retq
```

write

`write:`

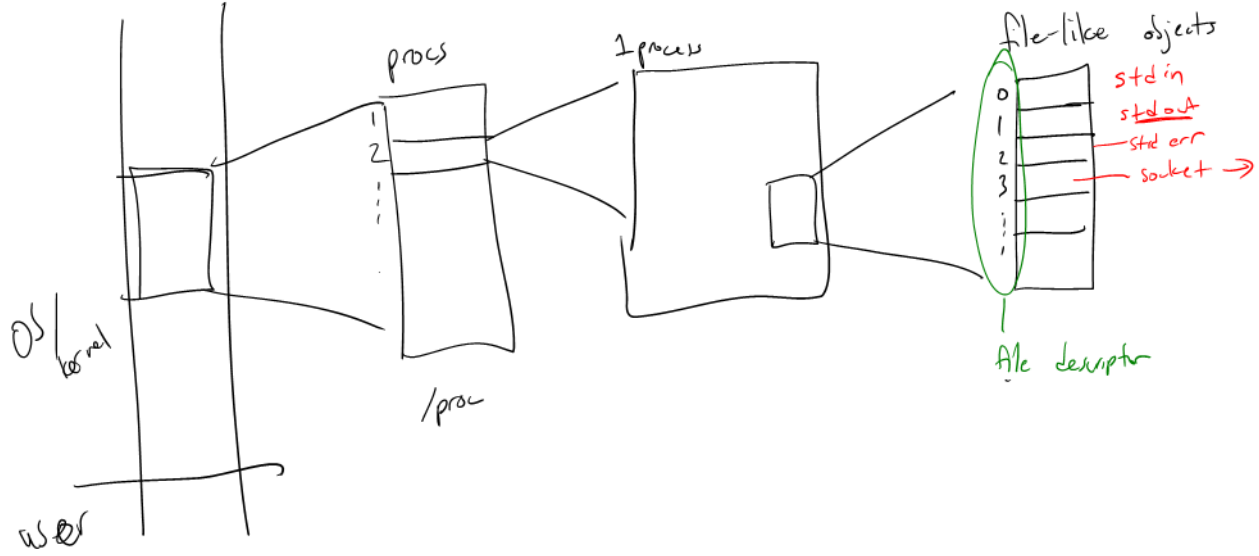
- Argument checking
- `syscall`
- Return value checking
- `ret`

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


procs
1 process
file-like objects
0
1
2
3
std in
stdout
std err
socket
file descriptor
/proc
OS
kernel
user

Using `write`