

C: `stdio.h`, `varargs`, `unistd.h`

CS 2130: Computer Systems and Organization 1

April 24, 2023

Announcements

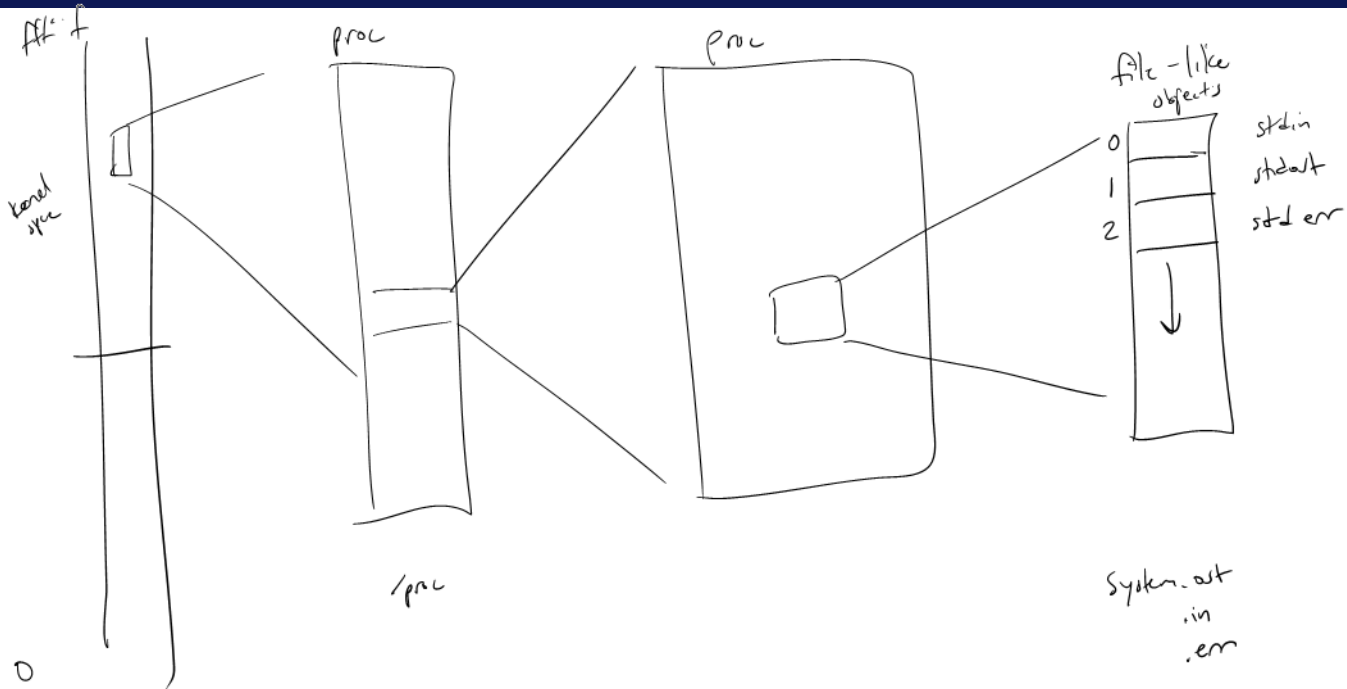
- Homework 9 due tonight at 11pm
 - Limited number of submissions, test your code before submitting
- Homework 10 due next Monday at 11pm
 - Lab tomorrow will be helpful
 - Limited number of submissions, test your code before submitting
- Final Exam: May 4, 7-10pm, Chem 402

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



printf

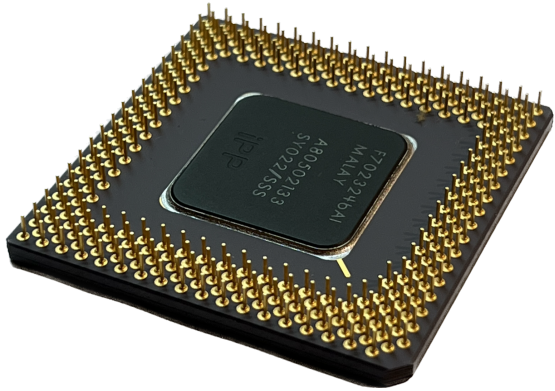
```
int printf(const char *format, ...);  
int fprintf(FILE *stream, const char *format, ...);
```

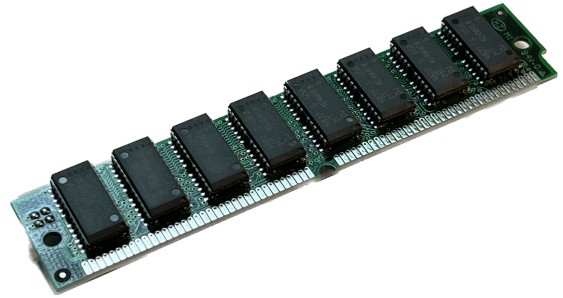
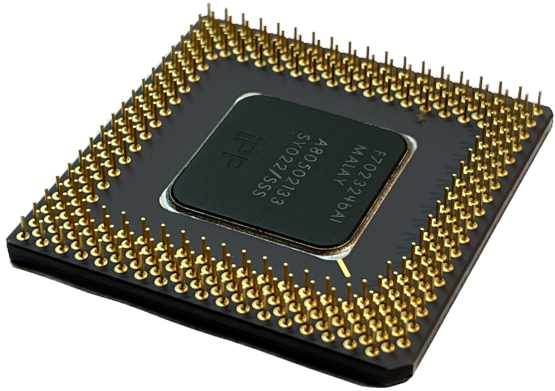
printf

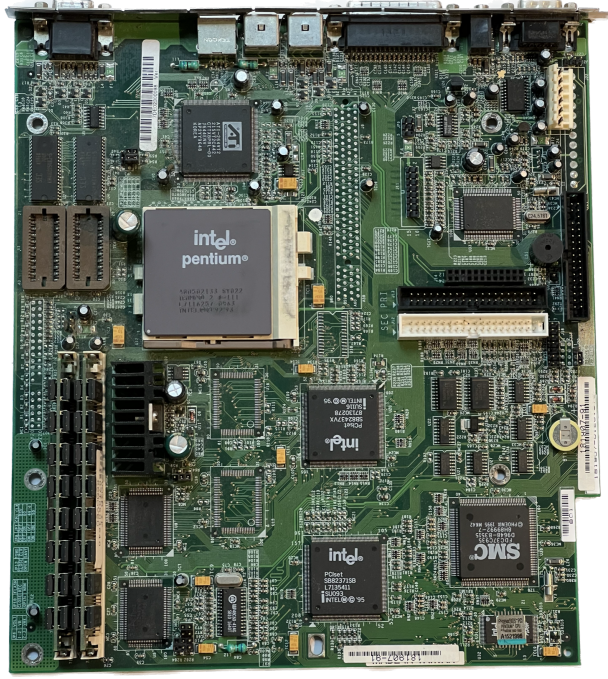
```
int printf(const char *format, ...);
```

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


Backing up...







Syscalls

`write:`

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

Using write