`string.h`, `printf` and `stdio.h`

CS 2130: Computer Systems and Organization 1
November 24, 2025

Announcements

- Homework 9 due tonight on Gradescope
- Lab 11 tomorrow (can check off for full credit by 12/5)
- No homework or quiz over break

`strcat()` warm-up example

printf

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

`printf()` example

Could you write `printf`?

printf

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

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

Backing up...

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

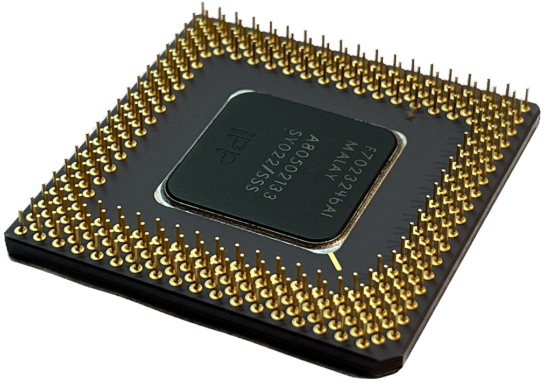
`stdio.h` manual page

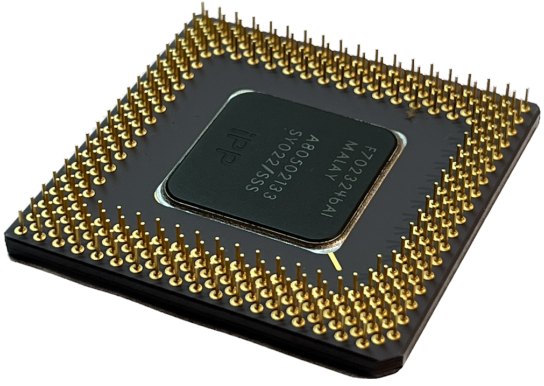
Could you write `printf`?

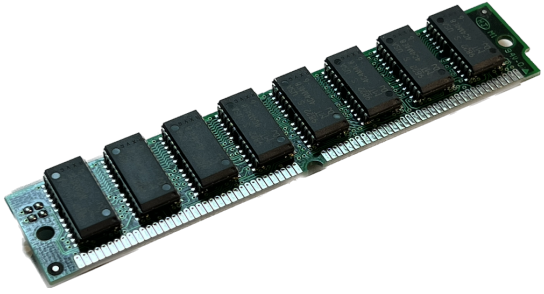
printf

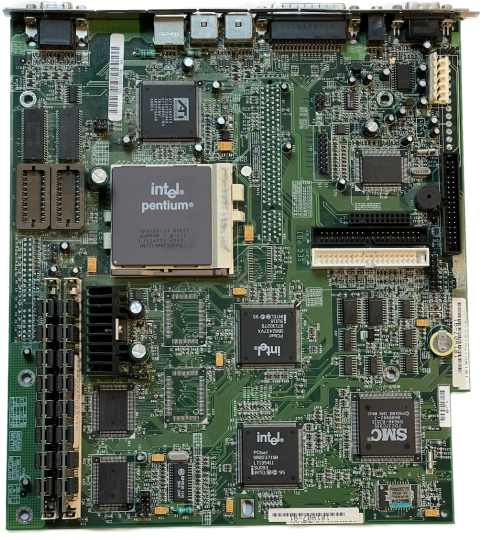
```
int printf(const char *format, ...); int fprintf(FILE
*stream, const char *format, ...);


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

Backing up a little more...


MALAY
A805021333
SY022/SSS


F70232A6AI
MALAY
A805021333
SY022/SSS




intel
pentium
intel
intel
SMC

Syscalls

write

`write`:

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

Using `write`