Lab 1: Booting a PC

Format Printing

`%o` Form

Print `0` first, then call `getuint()` to retrieve the number, then set base to 8, and goto label `number`.

Forced Sign Printing with `+`

Use a variable `snflag` to track whether the sign is forced to be printed, and print the sign according to this flag.

Left Padding and Right Padding

- Implemented a function `num_width()` to calculate the width that a number will occupy when printed (under a given base).
- Implemented a function `printnum_nopad()` to print a number recursively without printing any padding.
- `printnum()` calls the two functions above and thus can print a number with padding specified by `padc`.

Stack Backtrace

- Use `read_ebp()` and `read_eip()` to retrieve current `%ebp` and `%eip` respectively.
- Read the stack structure, retrieve old `%ebp`, return addresses and arguments. Trace though the `%ebp` chain until we meet 0 (set in `entry.S`).
- Use `debuginfo_eip()` to retrieve debug information. Specifically, we should search for `N_SLINE` stabs within the range `[lline, rline]` to get the line number.

The `time` Command

- Add a `time` command with function `mon_time()`, which runs the command specified in `argv[1]` with `runcmd()`.
- Use `read_tsc()` to get elapsed CPU cycles.