INF333 2023-2024 Spring Semester

Hasan Kayra Mike <kayramike@gmail.com>
Kutay Işık <kutay10@windowslive.com>

TP05 Design Document

The following is the sample design document for this TP. We will provide the general layout and the questions, you will provide answers in the `quote` environment.

1 Preliminaries

If you have any preliminary comments on your submission, notes for the TAs, or extra credit, please give them here.

Please cite any offline or online sources you consulted while preparing your submission, other than the Pintos documentation, course text, and lecture notes.

2 Booting Pintos

Q 2.1: Take screenshots of the successful booting of Pintos in QEMU and Bochs, each in both the terminal and the QEMU window. Put the screenshots under 'pintos/src/p0'.

A 2.1:

3 Debugging

3.1 Questions About BIOS

Q 2.2: Your first task in this section is to use GDB to trace the QEMU BIOS a bit to understand how an IA-32 compatible computer boots. Answer the following questions in your design document:

- What is the first instruction that gets executed?

- At which physical address is this instruction located?
- Can you guess why the first instruction is like this?
- What are the next three instructions?

A 2.2:

- First instr: ljmp $0x3630,$0xf000e05b
- Address: 0x000FFFF0
- It jumps from the top of the BIOS ROM to the beginning of the BIOS ROM, since there isn't enough space for instructions at between 0xffff0 and 0xfffff, which is the end address of BIOS ROM.
- cmpw $0xffc8,
- jne 0xd241d0b0
- xor

Q 2.3: Trace the Pintos bootloader and answer the following questions in your design document:

- How does the bootloader read disk sectors? In particular, what BIOS interrupt is used?
- How does the bootloader decides whether it finds the Pintos kernel?
- What happens when the bootloader could not find the Pintos kernel?
- At what point does the bootloader transfer control to the Pintos kernel?

A 2.3:

- BIOS INT 0x19 is used to read 512 bytes from the first sector of the hard drive
- The first sector read in turn allows the reading of other sectors and they combine to be the Pintos kernel, maximum of 1024 sectors.
- Kernel's ELF header contains a pointer to the entry point of the loaded kernel image and the loader extracts the pointer and jumps to the location it points to, ultimately, transfers control to it.

  Q 2.4: Add a screenshot of gdb while tracing the Pintos kernel.

  A 2.4: