Hacking on SPARC

Mohammad Yazdani

CS 850

2018

Where to begin?

The process of booloading a toy kernel on SPARC is partly easy, partly interesting and mostly challenging.

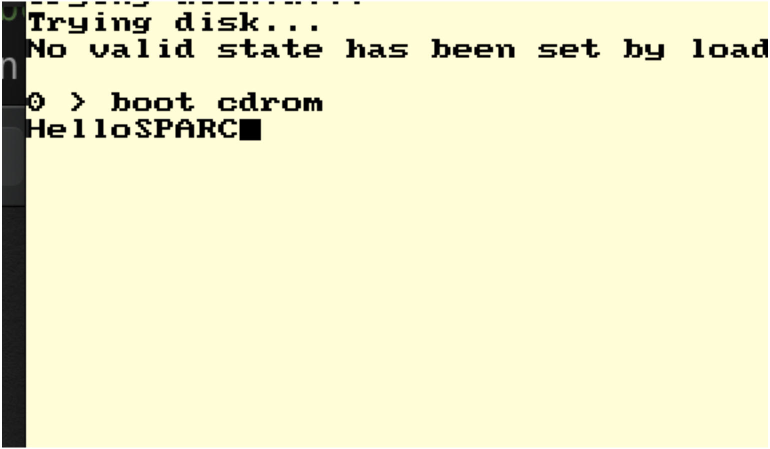
- OpenFirmware: Boot sequence is determined by a 512 byte table in the start of memory space.
- A position-independent binary has to be loaded into memory.
- In the binary, the loader starts with the text section
- Our boot.S determines the position of a string we want to print, and the printing function.
- Sets string pointer and length as 1st and second arguments and then calls the function.

Now: ld

Then we need to tell ld to:

- Kickout the headers
- Add a.out signature
- Size of our text section (comes after header)

After many hours



```
Trying disk...
No valid state has been set by load

0 > boot cdrom
HelloSPARC
```

What I learned and what was interesting:

- Compilers matter: I faced a lot of problems being able to get the right assembler and the right compiler. In the process of figuring out the bootloader I started to get more and more interested in diving deeper on compilers as an integral part of systems dev.
- Compared to what I read on boot loaders for other architectures, the SPARC boot loader is rather more straight forward. The basic process is figuring out your entry point, pointing to something you want to do, return from it and loop.