Lab Report 9
Linux Kernel: Built-in Modules

Christopher Padilla

ECEN 449 - 504

Purpose/Introduction

- Linux is a very configurable operating system, and we can actually boot Linux with device drivers built into it. Similarly, we can remove any unneeded device drivers. The purpose of this lab is to guide students through the process of customizing the Linux kernel by having the device drivers we made in previous labs be built into the operating system, instead of through the use of kernel modules.

Procedure

- The first part of this lab is to configure the operating system and using menuconfig we ensure we have selected our multiplication device driver be built in. In lab 6 we created a device driver for the "multiply" peripheral but, we had to instantiate this driver by first running the "insmod" command on the terminal. After doing this process we should be able to just "mknod" without doing insmod first.
- In part two, we then extend our kernel even more by having both the multiply peripheral from lab 6 *and* the ir_demod peripheral from lab 8 be built into the kernel. We repeat the steps in part one after creating a new appropriate device tree block and BOOT.bin. We are to observe the size of uImage afterwards.
- The final part was to remove some unneeded device drivers that exist in the default configuration of our kernel. These include sound card, network, and multimedia support drivers. We should observe a smaller uImage size.

Results

- During the boot-up process I was able to see the init_module printk() messages on the PICOCOM terminal that I had written back in lab 6 for the multiply periphal. This means that the module was successfully instantiated after being built into Linux. The size of uImage for the first part was 3369 KB.
- For the second part I was able to see the init_module prink() messages on the PICOCOM terminal for both labs 6 and 8 (multiply and ir_demod). The size of my uImage for part 2 was 3370 KB (slightly bigger).
- The final part was to remove some unneeded drivers. The size of uImage after compiling the kernel was 2445 KB (quite smaller).

Conclusion

- This lab gave me a taste of the configurability of Linux and guided me through the process of inserting and removing built-in drivers and understanding the trade-offs (time and size) of using built-in modules in lieu of loadable modules.

Questions

- There can be several advantages and disadvantages to using kernel modules as well as built-in drivers. Loadable kernel modules are great because we can have a reduced kernel size and still have the functionality we want without having the recompile the whole kernel; however, they can require more configuring and maintenance because you have to make sure everything the loadable module needs is present. Built-in device drivers are great because it is always ready to use, and there is minimal configuration needed after a bootup. Similarly, we can reconfigure the operating system to exclude the drivers we don't need if we want a "mean, lean" kernel. The disadvantage is that it takes time to recompile the kernel and the more built-in modules we have, the bigger our kernel is.