Simple Device Driver

UNX511 Week 5 Class 1

John Sellens

September 5, 2025

Seneca Polytechnic

Outline

Simple Device Driver

Simple Device Driver

Devices So Far

- Recall that devices have entries in the /dev directory
- Device files are “block” or “character” devices
  - Indicates whether you read/write per data block, or per character
  - I think the only /dev file types are b and c ...
- Device files have major and minor device numbers
  - e.g. see output of `ls -l /dev/null` – major 1, minor 3
  - They connect a named device to kernel device driver code
- The `mknod(1)` command creates device files

Loadable Kernel Modules

- Recall that the kernel runs in privileged mode and talks to hardware
- The linux kernel supports loadable code modules
  - Most device drivers are implemented as kernel modules
- Primary commands:
  - `insmod(8)` – insert (load) a module into the kernel
  - `rmmod(8)` – remove (unload) a module
  - `lsmod(8)` – list currently loaded modules
  - `modinfo(8)` – information about a module

Kernel Logging

- Kernel code (and modules) can log messages with `printk()`
  - Can't use stdio (and other libraries) from kernel code
- Messages can go to log files, the system console, or the kernel log
- `dmesg(1)` lets you access the kernel log
  - From the message buffer in kernel memory space
- Handy command: `dmesg -T -w`
  - human timestamps, follow output (like `tail -f`)

Peripheral Writer Module

- Simple, sample module – reads and writes “channels”
  - e.g. Multiple processes could pass data through the kernel
  - Perhaps a slighly contrived device
- Comes with `load.sh` and `unload.sh` scripts
- And a `userWriter` sample command to exercise it
- Let’s have a look ...

Summary

- Kernel device drivers can be (much) more complicated
- But careful abstraction and construction helps
- We have only scratched the surface