EMB 2
Device Driver
Manual

by
Torben Alexander Zurhelle
Ferdinand Lucas Bermudez

Table of Contents

How it was tested

Testing the reading and writing

This was done using cat and echo commands.

It is very straight forward so we will not go deep into this topic.

Testing the seek option

For the seek option we made a c program that opens the device node as a file and then used lseek to seek in the buffer.

We also made a case for the whence options that are defined by the lseek function: SEEK_SET, SEEK_CUR, SEEK_END.

*Note: Part of our test function can be seen below in the “Seeking” section.

Device Driver Interface

Getting started

The first step is to compile the module using the makefile.

Example: make all

The result of this will be a module called device_driver.ko.

The next step is to load the module in the kernel.

Example: sudo insmod device_driver.ko

To check if the module was loaded succesfully, use the dmesg command to check the kernel log, you should see a module/device loaded message.

The last step is to create a device node using the mknode command.

The major number for this device is 1337. You can choose between 2 minor numbers: 0 and 1, each have their own buffers.

Example: sudo mknode -m 666 /dev/device c <major> <minor>

Note: the node name should be unique per minor number.

Write to buffer

Writing is done with the help of the echo command, just echo any string to the device node and it will be stored in the buffer.

Example: echo -n <string> > /dev/device

The default maximum buffer size limit is 1024 bytes, this can be changed in the source code.

Read from buffer

Reading is done with the help of the cat command, just cat the device node and you will see the contents of the buffer will be displayed.

Example: cat /dev/device

Seeking

Seeking to a position in the buffer will take some c programming. You have to use a c-language function called lseek to be able to do this.

Example code:

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>

// Name of the device node.
#define DEV_NAME       "/dev/device"

int main (int argc, char* argv[])
{
   FILE *filepointer;
   int filedescriptor;

   // Open the file.
   filepointer = fopen(DEV_NAME,"w+");

   // Get the file descriptor.
   filedescriptor = fileno(filepointer);

   // Use lseek to move the position in the buffer.
   lseek(filedescriptor, 5, SEEK_SET);

   return 0;
}
```