

Betriebssysteme

Kerneltreiber

Praktikum 10

Fachhochschule Bielefeld
Campus Minden
Studiengang Informatik

Beteiligte Personen:

Name	Matrikelnummer
Peter Dick	1050185

29. Juni 2016

Inhaltsverzeichnis

1	Aufgabenstellung	3
2	Aufgabe 10.1	4
2.1	Vorbereitung	4
2.2	Durchführung	4
2.3	Fazit	4

Aufgabe 10 - Kernel Treiber

1 Aufgabenstellung

Kompilieren Sie das Modul lkm.c und machen Sie sich mit den Befehlen zum Laden bzw. Entladen von Modulen vertraut.

1. Vollziehen Sie den Aufbau eines Kernel Moduls nach und ergänzen Sie das im Info-Blatt vorgestellte Modul memory.c:
2. Ändern Sie die Major Nummer.
3. Erweitern Sie das Kernel Modul so, dass 256 Byte im Buffer gespeichert werden können (anstatt 1 Byte).
4. Erweitern Sie das Kernel Modul so, dass der Speicherinhalt des Buffers byteweise ausgelesen werden kann.
5. Schreiben Sie ein Programm, dass sowohl die Schreib- als auch die Lesemethode des Kernaltreibers verwendet.
6. Der zu speichernde Text soll beim Schreiben in den Kernel Space vom Kernel Modul mit Hilfe des printk-Befehls im Syslog-Dämon ausgegeben werden.
7. Unter welchen Bedingungen kann der vorgestellt Treiber verwendet werden, um auf reale Hardware zuzugreifen. Welche Anpassungen sind dazu notwendig?

2 Aufgabe 10.1

2.1 Vorbereitung

Wichtige Headerdateien installieren(siehe p10_info.pdf S. 1 Z1ff).
C-Projekt anlegen.
Makefile schreiben.

2.2 Durchführung

Code schreiben und dann testen bzw debuggen.

2.3 Fazit

Aufgabe 10.1.1

Ein Kernel Modul muss "linux/module.h" includieren. Das Kernel Modul hat eine "init-Methode" die mit dem Marco "module_init(<init-Methode>)" als Start-Methode definiert ist. Die Methode wird beim Einhängen aufgerufen. Auch hat das Modul eine "exit-Methode" die mit dem Marco "module_exit(<exit-Methode>)" als End-Methode definiert ist. Die Methode wird beim Aushängen aufgerufen. Jedes Modul muss das Marco "MODULE_LICENSE(<Lizenz>)" enthalten. Es gibt noch weitere Marcos die zu Informationszwecken da sind.

Aufgabe 10.1.2

```
int memory_major = 61;
```

Die Major Nummer ist in memory.c in Zeile 37 als Globale Variable definiert so ist es einfach die Major Nummer zu ändern.

Aufgabe 10.1.3

```
/* Allocating memory for the buffer */
memory_buffer = kmalloc(256, GFP_KERNEL);
if (!memory_buffer) {
    result = -ENOMEM;
    goto fail;
}
memset(memory_buffer, 0, 256);
```

Es werden in der "init-Methode" einfach 256 Byte alokiert statt 1 Byte.

Aufgabe 10.1.6

```
ssize_t memory_write( struct file *filp , char *buf ,  
                      size_t count , loff_t *f_pos) {  
  
    char *tmp;  
  
    tmp=buf+count-1;  
    printk(KERN_INFO tmp);  
    copy_from_user(memory_buffer , tmp , 1);  
    return 1;  
}
```

Mit "printk(KERN_INFO tmp);" wird der zu speichernde Text ausgegeben.