

Assignment 6 – Device Driver

Description:

This device driver project aims to create a custom driver to handle an associated device. It implements functionalities for reading, writing, and controlling the device, defined through file operations like open, read, write, ioctl (input/output control), and close. This driver interacts with user-applications, allowing translation of single words based on specific rules, such as transforming words starting with vowels and ending with 'y' into an Igpay Atinlay-like format.

Approach / What I Did:

Here's a description of my approach:

Driver Implementation: Developed a device driver in C to manage the associated hardware, enabling essential operations like reading and writing data to the device.

Translation Logic: Included translation logic within the driver to convert words according to specified rules, such as translating words beginning with vowels and ending with 'y' into a specific format.

File Operations: Defined file operation functions like open, read, write, ioctl, and close to enable user-space interaction and device control.

Kernel Interactions: Utilized kernel functions and libraries for memory allocation, user-space communication, and managing the device within the Linux kernel environment.

Issues and Resolutions:

Translation Logic Accuracy: Adjusting translation rules to ensure accuracy in converting words according to the specified conditions, addressing cases where translations were not aligned with expectations.

Memory Management: Addressed memory allocation issues, ensuring efficient usage of memory resources and preventing memory leaks within the driver.

User Interaction: Improved user interaction by refining error handling, providing clearer feedback during data operations, and enhancing overall usability in interfacing with the device.

Error Handling: Resolved issues related to error handling during data read/write operations, ensuring proper handling of errors to maintain stability and prevent system crashes.

Compatibility with C Standards: A complication arose due to the usage of for loop initial declarations, which are only permissible in C99 or C11 modes but might lead to compilation errors in earlier C standards. To resolve this, I had to use the -std=c99 flag during compilation, the code can be compiled in C99 mode, ensuring compatibility with for loop initial declarations. I did incorporate a change within the Makefile to make it work.

Analysis - N/A

Screen shot of compilation:

Testing compilation of Device Driver

```
student@student-VirtualBox: ~/Desktop/operating-systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module$ make all
make -C /lib/modules/uname -r /build/My/home/student/Desktop/operating-systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-150-generic'
  CC [M] /home/student/Desktop/operating-systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module/DeviceDriver.o
/home/student/Desktop/operating-systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module/DeviceDriver.c: In function 'readFromDevice':
/home/student/Desktop/operating-systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module/DeviceDriver.c:82:14: warning: 'test[0]' is used uninitialized in this function [-Wuninitialized]
   if ((test[0] == 'a' || test[0] == 'e' || test[0] == 't' || test[0] == 'o' || test[0] == 'u' ||
       ^
~~~~~
Building modules, stage 2.
MODPOST 1 modules
  CC [M] /home/student/Desktop/operating-systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module/DeviceDriver.mod.o
  LD [M] /home/student/Desktop/operating-systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module/DeviceDriver.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-150-generic'
student@student-VirtualBox: ~/Desktop/operating-systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module$
student@student-VirtualBox: ~/Desktop/Operating-Systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module$ sudo insmod DeviceDriver.ko
insmod: ERROR: could not insert module DeviceDriver.ko: File exists
student@student-VirtualBox: ~/Desktop/Operating-Systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module$ sudo mknod /dev/DeviceDriver c 415 0
mknod: /dev/DeviceDriver: File exists
student@student-VirtualBox: ~/Desktop/Operating-Systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module$ sudo chmod 666 /dev/DeviceDriver
student@student-VirtualBox: ~/Desktop/Operating-Systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Module$
```

Screen shot(s) of the execution of the Test file :

```
student@student-VirtualBox:~/Desktop/Operating-Systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Test$ make run
gcc -c -o Thomas_Sidney_HW6_main.o Thomas_Sidney_HW6_main.c -g -I.
gcc -o Thomas_Sidney_HW6_main Thomas_Sidney_HW6_main.o -g -I. -l pthread
./Thomas_Sidney_HW6_main
*****
Welcome to Igpay Atinlay Translator
*****
***** Language Options *****
0. Translate from English to Igpay Atinlayn
1. Translate from Igpay Atinlay to English
*****
Enter your desired translation direction (0 or 1): 0

**** Menu ****
1. Write data to the device
2. Read data from the device
3. Switch translation language
4. Exit
*****
Enter your choice (1-4): 1
Your Option = 1

Enter the string to write into driver: aio
Data Writing ...Done!

**** Menu ****
1. Write data to the device
2. Read data from the device
3. Switch translation language
4. Exit
*****
Enter your choice (1-4): 4
Your Option = 4

Thank you for using my Igpay Atinlay translator!
student@student-VirtualBox:~/Desktop/Operating-Systems-CSC-415/Assignment-6/csc415-device-driver-siid14/Test$
```