Simple Char driver

**Victor Rodriguez Bahena**

Rafael Lopez Peña A01630163

Ramiro Manuel Rubio Contreras A01223151

Víctor Ramirez Zepeda A01225277

Campus Guadalajara

Operating Systems

Jan-May 17

Introduction

This report covers the solution of the final project of the Operating System course, this project approaches us more to the practical functionality of some topics learned at the course. And let us to define a more clear line between the different kinds of drivers (Char, Block, and Network). Implementing and using one of them (char driver), to solve the problem proposed below.

Problem

Implement in a Linux Embedded board (RaspberryPi2) a char driver able to control a GPIO, the challenge was no using any high level language like python or C++, neither their libraries that simplify the GPIO handling. The solution includes libraries with basic functions to drive high or low the output of an onboard pin. An application of this library needs to be presented to prove the proper functioning of the char driver.

Solution

A char driver was implemented as a loadable Kernel module, since it’s a driver a node needs to be created in order to transmit information between the application and the driver. An option to transmit information to this driver is trough user space using commands like “echo” to channel a string into the driver’s node. The other way is using a library that we made using system calls like “write” to change the value of the GPIO inside of the application.

Conclusions

Kernel versions and Kernel headers must match for a module to build. We actually see a working char driver and also how to communicate an application with a Kernel module. Currently the PWM mode is not exact, because we don’t use a high definition timer. The PWM mode is implemented in a blocking fashion. If we have more time, we surely can implement more precise timers and tools like alarms to avoid the blocking implementation that is actually running.

Latex Link: <https://www.overleaf.com/read/gxpdnfjxygwy>

Git Link: <https://github.com/Aribababa/Sistemas-Operativos>