

Práctica 1: GNU/Linux Microcontrolador

Alberto Acosta López, Francisco Javier López Franco
Luis Fernando Peña Flores

Resumen—En esta práctica ~~introdutoria al laboratorio de Principios de Mecatrónica~~ se comenzó a utilizar el sistema operativo Linux y sus comandos básicos en terminal como entorno de desarrollo de las prácticas. Así mismo, se aprendió lo básico de git para ser usado como controlador de versiones a lo largo del curso. Finalmente, como primer acercamiento a Arduino se implementaron programas básicos con el mismo.

1. INTRODUCCIÓN

El uso de los sistemas operativos basados en Linux es de gran importancia, ya que permite ~~controlar de una manera más específica~~ diversos comandos y acciones necesarias para el desarrollo de sistemas embebidos. Cuando se desarrolla un programa o se tiene un conjunto de instrucciones programadas no solo para sistemas embebidos, sino para cualquier aplicación, es altamente recomendado que se utilicen repositorios, por lo que Linux tiene una integración nativa con los comandos de Git. Para el desarrollo de un sistema embebido, es necesario un dispositivo para que la información sea procesada y ejecute los comandos definidos por el programador, sin embargo, una computadora cualquiera presenta problemas para procesar la distinta información que recibe de los sensores requeridos, por lo que en esta práctica se utilizó un Arduino, el cual es capaz de procesar y recibir la información de distintos sensores y devolver una señal.

Los objetivos propuestos para esta práctica es aprender a utilizar comandos en la terminal de un sistema operativo basado en Linux, el cual fue Ubuntu en este caso; utilizar los comandos básicos para el uso correcto de un repositorio en Git, el cual fue alojado en Github debido a que previamente existía una cuenta del equipo y, por último, aprender el funcionamiento de los comandos y la forma de programarlos en un Arduino para que éste procese la información que se le haga llegar, ya sea directamente del teclado de la computadora o de voltajes y corrientes de un circuito armado en una protoboard.

Este documento tiene como organización un marco teórico, donde se explica la funcionalidad y objetivo de cada herramienta utilizada, un desarrollo, donde se explica la metodología y pasos a seguir de la práctica, los resultados obtenidos al finalizar la práctica, las conclusiones individuales de cada integrante del equipo y, finalmente, el papel que desempeñó cada integrante del equipo para llevar a cabo el correcto funcionamiento de la práctica.

2. MARCO TEÓRICO

La terminal de Linux es el medio estándar de comunicación e interacción con un sistema Linux, ya que es una consola interna del sistema conectado al *kernel* de Linux. [1] En un sistema Linux, el *shell* es una línea de comando que interpreta los comandos del usuario y los archivos de *script* y le comunica al sistema lo que se debe hacer con ellos. Es una vía de comunicación entre el *kernel* y otros procesos para mandarle mensajes de salida al usuario, y para recibir mensajes de entrada del usuario. Cuando se abre la terminal, el sistema muestra la información básica del usuario, el *hostname* del sistema y el directorio en el cual se encuentra actualmente. [2]

Git es el sistema más utilizado de control de versiones en el mundo. A diferencia de software de control de versiones, Git no se fija en el nombre de los archivos, sino que se enfoca en el contenido del archivo. En un repositorio, utiliza una combinación de conversión delta de archivos, compresión y almacenamiento explícito de los contenidos del directorio, así como la versión de los objetos. El contenido de los archivos, así como sus relaciones entre ellos se guardan de manera segura mediante un algoritmo de encriptación SHA1. Uno de sus mayores objetivos es la flexibilidad de almacenar distintos tipos de desarrollo no lineal y es eficiente tanto en pequeños como grandes proyectos. [3]

Arduino es una plataforma *open-source* utilizado para construir proyectos electrónicos, ya que consiste de una parte física programable, referido como un microcontrolador, y una parte de software, utilizado para escribir y mandar el código de la computadora al dispositivo. A diferencia de otros dispositivos programables, Arduino no necesita otro pedazo separado de hardware para mandarle nuevo código, sino que requiere un cable USB. [4]

3. DESARROLLO

Como ya se mencionó anteriormente, la práctica fue dividida en dos secciones. La primera consistió en familiarizarse con comandos de consola en Linux, así como poner en práctica los comandos básicos de git. La segunda se enfocó en habituarse a utilizar lo básico de Arduino.

Para la primera parte, se creó un repositorio en la plataforma de GitHub y se clonó en la computadora. Una vez dentro del repositorio se verificó, al ver que el existía el *archivo .git*, que este fuera en realidad un repositorio de git. Posteriormente, se utilizaron distintos comandos de Linux

para navegar y crear el árbol de directorios como se presenta en la figura 1:

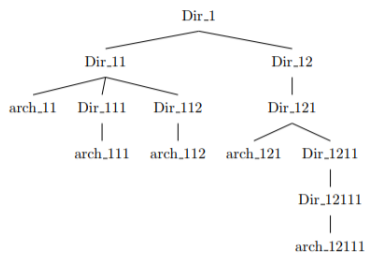


Figura 1. Árbol de directorios

Una vez creado se probaron más comandos para ver el estado de un directorio, borrarlo o añadir un archivo al mismo. Luego, para finalizar la primera sección de la práctica se realizó en el repositorio el primer **commit**, al cual se le hizo un **push** hacía GitHub.

Enseguida se comenzó con el desarrollo de la segunda sección de la práctica. Dicha sección consistió en, primero, cargar un programa de ejemplo en el Arduino y verificar que este último funcionara correctamente. Una vez verificado se procedió a simular un alumbrado que responde a cambios de intensidad luminosa. Para ello se utilizó una fotoresistencia y un led. Con el Arduino se leía el estado actual de la foto resistencia y, con base en ello, se prendía o no el led. Adicionalmente, se incluyó un botón que al apretarlo controlaba el encendido del led. Su funcionalidad es análoga a la implementada con la fotoresistencia, se leía el valor del estado del botón y se decidía si se prendía o no el led.

Finalmente, sé debía implementar un programa que, dado un número hexadecimal por teclado, fuera capaz de desplegarlo en un display de 7 segmentos. Se utilizó un display de ánodo común en combinación con un programa escrito en lenguaje C que, primero, validaba el número hexadecimal recibido como entrada y, finalmente, asignaba los valores lógicos correspondientes a cada uno de los segmentos. En caso de que la entrada no fuera valida no se prendía segmento alguno. La tabla de verdad del display a 7 segmentos se presenta en la figura 2.

4. RESULTADOS

Dentro de la primera sección de la práctica se cumplieron todos los ejercicios planteados dentro de la misma. Se tuvo un ligero inconveniente, al momento de hacer **push**, con una de las carpetas del árbol de directorios. Ya que, al ser una carpeta sin archivos, git la ignoró y no se añadió como parte del commit.

Respecto a la segunda sección de la práctica, también se cumplieron todos los objetivos planteados. Al ejecutar el archivo de ejemplo, 'Blink', se comprobó que el Arduino funcionaba correctamente. Entonces, como pequeño experimento, se procedió a modificar el valor del tiempo de delay y hacer que el led del Arduino prendiera más o

Digit	Display	gfedcba	abcdefg	a	b	c	d	e	f	g
0	0	0x3F	0x7E	on	on	on	on	on	on	off
1	1	0x06	0x30	off	on	on	off	off	off	off
2	2	0x5B	0x6D	on	on	off	on	on	off	on
3	3	0x4F	0x79	on	on	on	on	off	off	on
4	4	0x66	0x33	off	on	on	off	off	on	on
5	5	0x6D	0x5B	on	off	on	on	off	on	on
6	6	0x7D	0x5F	on	off	on	on	on	on	on
7	7	0x07	0x70	on	on	on	off	off	off	off
8	8	0x7F	0x7F	on	on	on	on	on	on	on
9	9	0x6F	0x7B	on	on	on	on	off	on	on
A	A	0x77	0x77	on	on	on	off	on	on	on
b	b	0x7C	0x1F	off	off	on	on	on	on	on
C	C	0x39	0x4E	on	off	off	on	on	on	off
d	d	0x5E	0x3D	off	on	on	on	on	off	on
E	E	0x79	0x4F	on	off	off	on	on	on	on
F	F	0x71	0x47	on	off	off	off	on	on	on

Figura 2. Tabla de verdad de siete segmentos

menos rápido.

En el caso del programa que utilizaba la fotoresistencia, se tuvo que medir de forma empírica el valor de corte que determinaba si el led debía encenderse o apagarse. Una vez que se realizó dicha medición fue sencillo conectar el circuito con la fotoresistencia y el led, el cual se prendía siempre y cuando el valor de la fotoresistencia superara el valor de corte. Al momento de agregar el botón para controlar el prendido del led se procedió de la misma manera que con la resistencia. Se procedió sin más inconvenientes.

Finalmente, al momento de implementar el programa con el display de 7 segmentos, hubo pequeños inconvenientes al momento de asignar pines de de salida utilizando for-loops, así como asignar, segmento a segmento, los valores de salida para el display. Una vez atendido ese detalle, cabe recalcar que la consola de Arduino agrega un salto de línea al momento de ingresar datos por teclado. Dicho efecto hacía que la última entrada al programa fuera un salto de línea y, por lo tanto, no se reconocía como un número válido. Para corregir el problema bastó con modificar la configuración de la consola para que no agregara el salto de línea. Con esto se procedió sin mayores inconvenientes y se finalizó la práctica exitosamente.

5. CONCLUSIONES

5.1. Alberto Acosta

Al finalizar esta práctica, aprendí nuevos comandos de la terminal de Linux, como *touch*, *head*, *tail*, a pesar de tener un conocimiento medio del uso de terminal en Linux. En cuanto a Git, fue interesante utilizarlo en la práctica, aunque ya tenía conocimiento más avanzado del uso más profesional de la herramienta, sin embargo, mi mayor reto fue Arduino, ya que no tenía conocimiento alguno de su funcionamiento ni de la forma de crear proyectos.

Me siento satisfecho por el conocimiento aprendido en la práctica y espero con ansias aprender cosas más difíciles.

5.2. Francisco López

Me familiaricé con la interfaz de Linux y con su terminal, la cual puedo decir que es útil para realizar este tipo de problemas. Igualmente conocí la interfaz de Arduino, y puedo decir que la programación es bastante intuitiva, y con un poco de conocimiento del lenguaje C puedes crear componentes complejos en tan solo unas horas. Tengo buenas expectativas para las prácticas futuras.

5.3. Fernando Peña

Una vez finalizada la práctica aprendí, de forma básica, la manera de operar un sistema operativo Linux a base de consola de comandos. En particular, utilicé comandos para desplazarme dentro de distintos directorios, ver su contenido, su status, así como crearlos y eliminarlos. Respecto a Git y GitHub fui capaz reforzar mis conocimientos acerca de estas útiles herramientas, ya que contaba con conocimiento previo de las mismas. Asimismo, aprendí acerca del funcionamiento del Arduino. Fue importante observar su relación con los temas vistos en la clase teórica. Finalmente, pude comparar al Arduino con otros dispositivos, como el FPGA, e implementar programas básicos dentro del mismo.

6. ROL O PAPEL

1. **Alberto:** Revisar puntos a realizar de la práctica, armar el circuito y checar errores presentados.
2. **Francisco:** Programar el código en Arduino y checar errores presentados.
3. **Fernando:** Armar el circuito y checar errores presentados.

7. FUENTES CONSULTADAS

REFERENCIAS

- [1] Wikipedia. Linux console. [Online]. Available: https://en.wikipedia.org/wiki/Linux_console
- [2] M. Anicas. An introduction to the linux terminal. [Online]. Available: <https://www.digitalocean.com/community/tutorials/an-introduction-to-the-linux-terminal>
- [3] Atlassian. What is git. [Online]. Available: <https://www.atlassian.com/git/tutorials/what-is-git>
- [4] SparkFun. What is an arduino? [Online]. Available: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>