

Instituto Tecnológico de Costa Rica



Área Académica de Ingeniería Mecatrónica

Microprocesadores y Microcontroladores

Tarea 2

Profesor:

Ing. Rodolfo Piedra Camacho

Integrantes:

Elena Cantero Acosta (2018133278)

Mauricio Brenes Siles (2018319132)

Ricardo Castro Barrenechea (2018112991)

Fabián Álvarez Juárez (2018089216)

Jose Luis Mora Solis (2018085016)

Preguntas Teóricas (20 pts, 2pts c/u)

1) ¿Qué es el Shell de Linux?

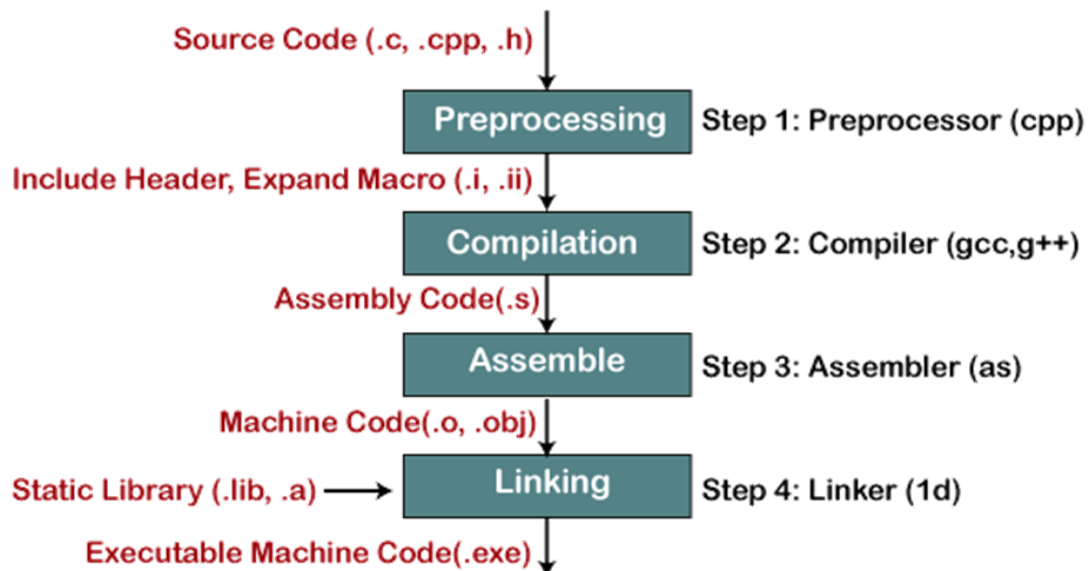
Shell es una interfaz que se encarga de establecer una comunicación entre el “kernel” y el usuario del sistema (el kernel es quién se comunica con el hardware de la computadora). Por medio del Shell se pueden dar diversos comandos que nos permiten realizar diferentes tareas a nivel del sistema operativo. Algunas características interesantes del Shell son las siguientes:

- La Shell puede ser de tipo CLI (Command Line Interface) o GUI (Graphic User Interface).
- En la mayoría de las distribuciones de Linux se ejecuta el Shell “Bash”.
- Las aplicaciones interactúan directamente con el kernel y el Shell es la herramienta que poseen los usuarios para hacerlo.
- El Shell original es se llama “sh”, desarrollado por Steve Bourne

2) ¿Qué es gcc?

GCC significa “GNU Compiler Collection”, en esencia es un compilador “Open-Source” disponible en Git, que funciona en múltiples lenguajes de programación y fue lanzado en 1987. Es utilizado principalmente para compilar C, C++, Objective-C, Fortran, entre otros. Un ejemplo de un programa compilado en GCC es el Linux Kernel.

El proceso de compilación de GCC es el siguiente:



[1]

- Preprocesamiento: El preprocesador GNU crea un archivo intermedio tipo ‘.i’ que posee el código fuente.

- **Compilación:** El código fuente preprocesado es compilado en código de ensamblador para un procesador específico y crea un archivo tipo `‘.s’`
- **Ensamblador:** El código de ensamblador es convertido en “código máquina” y crea un archivo tipo `‘.o’`.
- **Enlazador:** El enlazador es un archivo `‘.exe’` que vincula el código objeto con el código de la biblioteca, creando así el archivo ejecutable que inicializa la aplicación.

3) ¿Cuál es la diferencia entre gcc y g++?

Ambos son compiladores de software compatible con GNU, pero lo compilan de maneras diferentes. Por ejemplo, ambos pueden leer archivos tipo c con terminaciones `.c` o `.cpp`, pero gcc los va a cargar como un archivo tipo c y tipo c++, respectivamente; mientras que g++ lee ambos archivos como tipo c++. También, al trabajar con un documento, g++ va a asociar directamente las librerías que tiene como estandarizadas para c++; mientras que gcc no pero puede hacerse de manera manual. Esto último se traduce a que son básicamente lo mismo pero con una opción de compilación y dos de indexación diferentes, donde g++ es más eficiente en archivos enfocados en c++ mientras que aunque gcc puede diferencias c de c++ ocupa más líneas de código para llegar a esa eficiencia en c++ (lo que a la vez le permite ser más personalizable). [7]

4) ¿Cómo se compila un programa en C utilizando el compilador gcc desde la terminal de Linux? Brinde un ejemplo.

Primero hay que ubicarse en el directorio donde se encuentre el archivo tipo C que se quiera compilar. Para ello, en la terminal se va a escribir el comando `“pwd”` (abreviatura de print working directory) para indicar el directorio donde se encuentra actualmente, luego se coloca `“cd (abreviatura de change directory) dirección/del/directorio/deseado”`, por último se chequea que esté bien ubicado de nuevo con `“pwd”`. [9] Ahora se tiene que compilar el archivo en sí para lo que se coloca el comando `“gcc (del compilador) nombre_del_archivo.c”`. Por default, va a generar un archivo llamado `a.out` que contiene las salidas del archivo tipo C recién compilado y se puede ejecutar con `“./a.out”`. Para más recomendaciones respecto a ello, se recomienda revisar la publicación de sshubhamk1 de Geeks for Geeks. [8]

5) ¿Cómo se ejecuta un programa en C ya compilado desde la terminal de Linux?
Al estar en el directorio donde se encuentra el programa ya compilado con el comando **`./nombre_del_programa`**

6) ¿Qué es el comando ifconfig de Ubuntu? ¿Qué información provee?

Ifconfig se usa para configurar los interfaces de red residentes en el núcleo. Si no se pasan argumentos, ifconfig muestra el estado de los interfaces activos. Si se pasa un sólo argumento de interfaz, muestra sólo el estado de esa interfaz; si se da un solo argumento `-a`, muestra el estado de todos los interfaces, incluso de aquellos que no están activos. De otro modo, configura un interfaz.

7) ¿Para qué sirve el protocolo SSH?

El protocolo SSH es un método para mejorar la seguridad cuando se realiza un login remoto desde una computadora a otra. Utiliza herramientas de autenticación y encriptación de los mensajes para proteger la seguridad e integridad de los mensajes. [7]

Dentro de los usos más comunes de este protocolo se encuentra el acceso remoto seguro a recursos como servidores, bases de datos, routers, firewalls, etc; ejecución remota de comandos, entrega de actualizaciones y parches de software, y en la transferencia automática e interactiva de archivos. [10]

8) ¿Qué es la herramienta openssh-server de Linux?

OpenSSH es una colección de herramientas para el control remoto y la transferencia de datos entre computadoras. Es una versión gratuita de la familia de herramientas del protocolo SSH. OpenSSH proporciona un servidor y herramientas de cliente para facilitar el control remoto seguro, encriptado y operaciones de transferencia de archivos reemplazando las herramientas originales o "legacy". [11]

9) Describa las diferencias entre las tarjetas basadas en microprocesador Raspberry Pi 2 y Raspberry Pi 3.

Característica	Raspberry Pi 3	Raspberry Pi 2
Conectividad	WiFi y Bluetooth	Puerto USB
Hardware	Procesador de 1.2GHz Quad Core	Procesador de 900 Mhz de procesamiento central
Plataforma	Requiere de un buscador web para desplegar el contenido	Requiere un motor Java Script para correr el código
Rendimiento	Tiene un alto rendimiento	Posee un rendimiento alto, pero menor al del modelo Pi 3.
Velocidad del Clock	Posee una velocidad de 400 MHz	Posee una velocidad de clock de 250 MHz

RAM	Velocidad de RAM de 450 MHz	Velocidad de RAM de 900 MHz
-----	--------------------------------	--------------------------------

Otras diferencias:

1. En Raspberry Pi 3, el tiempo de instalación del software es menor, mientras que el tiempo de instalación del software es mayor en la versión Raspberry Pi 2.
2. En Raspberry Pi 3, el SoC a menudo se calienta fácilmente y requiere algunos agentes de enfriamiento para controlar los niveles de temperatura, mientras que en Raspberry Pi 2 no se requiere un radiador ya que las temperaturas en SoC no alcanzan altos niveles fácilmente.
3. En Raspberry Pi 3, los dispositivos periféricos a menudo se pueden conectar mediante conectividad inalámbrica, mientras que en Raspberry Pi 2, los dispositivos periféricos se pueden conectar a través de concentradores USB con alimentación.
4. En Raspberry Pi 3, la capacidad de representación de cuadros por segundo es de 37.9 FPS, mientras que la velocidad de cuadros por segundo en Raspberry Pi 2 es de 25.4 FPS.
5. En Raspberry Pi 3, la tarjeta Micro SD es un mecanismo basado en fricción en lugar de un tipo push-push con resorte que se ha utilizado hasta la versión Raspberry Pi 2.
6. En Raspberry Pi 3, el consumo de energía inactiva es menor, es decir, 2.5 W, mientras que el consumo de energía inactiva en Raspberry Pi 2 es 3.2 W, que es más.
7. En Raspberry Pi 3, la potencia bajo carga es de 3.8 W, que es la misma en el caso de las versiones Raspberry Pi 3 y Raspberry Pi 2.

10) Defina los siguientes conceptos de programación por hilos: Thread, lock, join y deadlock

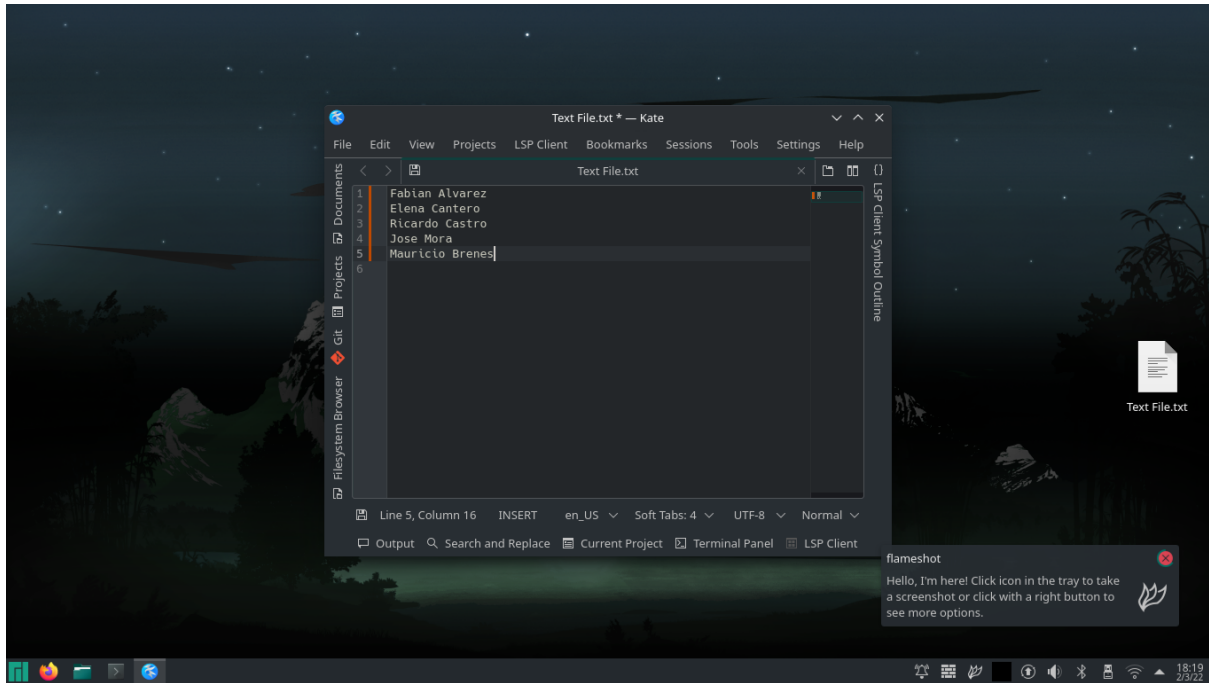
- Thread: según [12], un hilo es la menor de las estructuras lógicas de programación que se ejecuta de forma secuencial por parte del planificador del sistema operativo. Un hilo es, en pocas palabras, una tarea que puede ser ejecutada al mismo tiempo que otra. Un conjunto de hilos que se ejecutan a la vez y comparten los mismos recursos se llama “proceso”.
- Lock: según [13], la instrucción lock consiste en el bloqueo de exclusión mutua de un objeto determinado, ejecuta un bloque de instrucciones y luego libera el bloqueo.

Mientras se mantiene un bloqueo, el subproceso que lo mantiene puede volver a adquirir y liberar el bloqueo. Ningún otro subproceso puede adquirir el bloqueo y espera hasta que se libera.

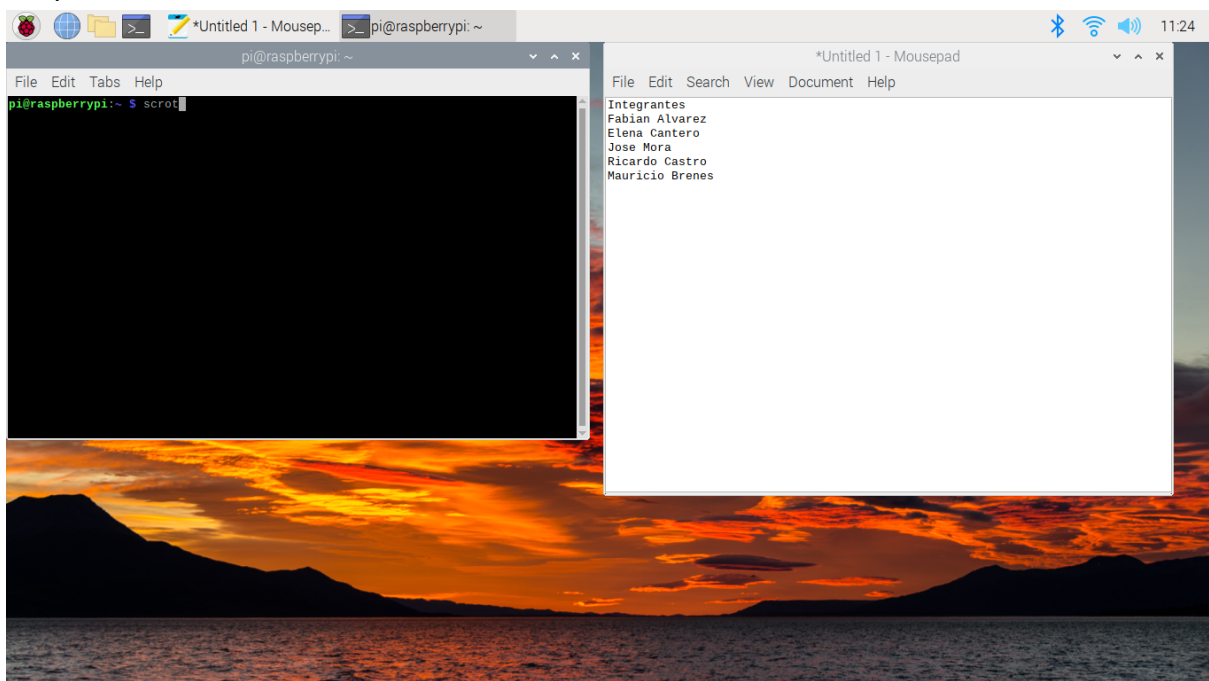
- Join: según [14], el método permite a un hilo quedar a la espera que termine un segundo hilo. El método `join()` suele utilizarse para mantener un orden en la secuencia de los hilos. Así, podemos arrancar una secuencia de hilos llamando a `join()` para que cada uno finalice en el orden que hemos marcado según la llamada a `join()`.
- Deadlock: también conocido como abrazo mortal, ocurre cuando un proceso espera un evento que nunca va a pasar. Aunque se puede dar por comunicación entre procesos, es más frecuente que se dé por manejo de recursos. En este caso, deben cumplirse 4 condiciones para que se dé un "deadlock":
 - Los procesos deben reclamar un acceso exclusivo a los recursos.
 - Los procesos deben retener los recursos mientras esperan otros.
 - Los recursos pueden no ser removidos de los procesos que esperan.
 - Existe una cadena circular de procesos donde cada proceso retiene uno o más recursos que el siguiente proceso de la cadena necesita.

Creación de una imagen y ejecución de un sistema operativo sobre la tarjeta Raspberry Pi
2/3. (8 pts)

Manjaro



Raspbian OS



Bibliografia

- [1] JavaTpoint, "javatpoint.com," javatpoint, [Online]. Available: <https://www.javatpoint.com/gcc-linux>. [Accessed 06 03 2022].
- [2] G. team, "gcc.gnu.org," gcc.gnu.org, 17 01 2022. [Online]. Available: <https://gcc.gnu.org/>. [Accessed 06 03 2022].
- [3] C. Bits, "youtube.com," Contando Bits, 09 03 2021. [Online]. Available: <https://www.youtube.com/watch?v=YUCXzp8n93U>. [Accessed 06 03 2022].
- [4] C. Academy, "youtube.com," Capacity Academy, 23 05 2011. [Online]. Available: <https://www.youtube.com/watch?v=Wsog1hPPipQ>. [Accessed 06 03 2022].
- [5] R. Dev, "youtube.com," Rincon Dev, 31 08 2020. [Online]. Available: <https://www.youtube.com/watch?v=m7-sDw13TAU>. [Accessed 06 03 2022].
- [6] W. Altaqi. "The What, Why & How of SSH Protocol". KEYFACTOR, [Online]. Available: <https://www.keyfactor.com/blog/ssh-protocol/> [Accessed 06 03 2022].
- [7] York, M. "What is the difference between g++ and gcc?", Stack overflow, [Online]. Available: <https://stackoverflow.com/questions/172587/what-is-the-difference-between-g-and-gcc> [Accessed 10 03 2022].
- [8] sshubhamk1, "gcc command in Linux with examples", Geeks for Geeks, [Online]. Available: <https://www.geeksforgeeks.org/gcc-command-in-linux-with-examples/> [Accessed 10 03 2022].
- [9] clissold345, "Using the terminal", Ubuntu documentation, [Online]. Available: <https://help.ubuntu.com/community/UsingTheTerminal> [Accessed 10 03 2022]
- [10] SSH Academy. "SSH Protocol - Secure Remote Login and File Transfer". SSH [Online]. Available: <https://www.ssh.com/academy/ssh/protocol> [Accessed 06 03 2022].

- [11] Ubuntu. "OpenSSH Server". Ubuntu. [Online]. Available: <https://ubuntu.com/server/docs/service-openssh> [Accessed 06 03 2022]
- [12] L. R. Espiritu, "aleph.org," aleph.org, 2021. [Online]. Available: [https://aleph.org.mx/que-son-los-hilos-en-la-programacion#:~:text=Un%20hilo%20\(en%20ingl%C3%A9s%20%E2%80%9Cthread,una%20sentencia%20despu%C3%A9s%20de%20otra\).](https://aleph.org.mx/que-son-los-hilos-en-la-programacion#:~:text=Un%20hilo%20(en%20ingl%C3%A9s%20%E2%80%9Cthread,una%20sentencia%20despu%C3%A9s%20de%20otra).) [Accessed 07 03 2022].
- [13] B. & K. P. Wagner, "docs.microsoft.com," Microsoft, 06 03 2022. [Online]. Available: <https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/statements/lock#:~:text=La%20instrucci%C3%B3n%20lock%20adquiere%20el,adquirir%20y%20liberar%20el%20bloqueo.> [Accessed 07 03 2022].
- [14] C. Garcia, "wordpress," 13 06 2009. [Online]. Available: [https://eaddfsi.wordpress.com/2009/06/13/sincronizacion-de-hilos/#:~:text=Este%20m%C3%A9todo%20permite%20a%20un,la%20llamada%20a%20join\(\).](https://eaddfsi.wordpress.com/2009/06/13/sincronizacion-de-hilos/#:~:text=Este%20m%C3%A9todo%20permite%20a%20un,la%20llamada%20a%20join().) [Accessed 07 03 2022].
- [15] F. Restrepo, "ferestrepoca.github.io," [Online]. Available: https://ferestrepoca.github.io/paradigmas-de-programacion/progconcurrente/concurrente_teor%C3%ADa/index.html. [Accessed 07 03 2022].