



## Instituto Tecnológico de Costa Rica

*Escuela de Ingeniería Mecatrónica*

# Tarea 2

## Manejo de Tarjeta Raspberry Pi

*Respuestas de la sección teórica*

**Profesor:** Ing. Rodolfo Piedra Camacho

**Grupo:** 4

**Estudiantes:**

Adrián Alonso Arguedas Arias  
Selvin Josué Arias Lara  
Kendall Aarón Guerrero González  
Héctor Jiménez Viales  
Mariana Mora Guzmán

II Semestre  
2025

# Respuestas teóricas

## 1) ¿Qué es el Shell de Linux?

El Shell en Linux hace referencia a una interfaz de línea de comandos que le permite a las personas usuarias la interacción con el sistema operativo, de forma que proporciona un entorno para la ejecución de programas, así como la gestión de archivos, directorios y otras funciones. Todo esto se logra gracias a que Shell puede interpretar los comandos que el usuario le indique, ya sea desde la ventana del terminal o desde un archivo script. Los más populares son Bash, Csh y Ksh [1].

## 2) ¿Qué es gcc?

GNU Compiler Collection, conocido popularmente como GCC, es un compilador con la capacidad de recibir un programa de una fuente de lenguaje (C, C++, Objective C y Fortran) y encargarse de generar un programa binario que se pueda ejecutar en el lenguaje de la máquina donde se desea correr [2].

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

Al venir GCC instalado por defecto en Linux, simplemente se debe abrir una terminal y seguir los siguientes pasos [3]:

1. Tener el código en C con su respectivo nombre.
2. En la terminal, para compilar:

```
gcc nombre_archivo.c -o nombre_ejecutable
```

Esto va a generar un ejecutable que se llama `nombre_ejecutable`.

3. Ejecutar el programa:

```
./nombre_ejecutable
```

Por ejemplo, se realiza la suposición de un programa llamado `hola_mundo`, que tiene como código lo siguiente:

```
1 #include <stdio.h>
2 int main() {
3     printf(" Hola , mundo!\n");
4     return 0;
5 }
```

Siguiendo los pasos descritos anteriormente:

1. Nombre: hola\_mundo
2. Compilar en terminal:

```
gcc hola_mundo.c -o hola_mundo
```

3. Ejecutar el programa:

```
./hola_mundo
```

Y esto arrojará en pantalla el mensaje: ¡Hola, mundo!

#### **4) ¿Cómo se ejecuta un programa en C ya compilado desde la terminal de Linux?**

Como se mencionó en el punto 3, una vez que ya se ha compilado el archivo, simplemente debe escribirse lo siguiente [3]:

```
./nombre_ejecutable
```

Y con esto, ya se ejecuta el programa.

#### **5) ¿Qué es el comando ifconfig de Ubuntu? ¿Qué información provee?**

El comando `ifconfig` es aquel que se emplea, de forma general, para mostrar y configurar interfaces de red del sistema, o sea, cuando se necesite consultar acerca de la información que está relacionada con la configuración de red del equipo. Al ejecutarse, se puede obtener información como el nombre de la interfaz, dirección IP, dirección MAC, entre otras [4].

#### **6) ¿Para qué sirve el protocolo SSH?**

Existe gran variedad de protocolos de internet (conjunto de reglas que sirven para enrutar y direccionar paquetes de datos), siendo Secure Shell (SSH) uno de ellos. Este se enfoca primordialmente en la conexión con máquinas a las que se puede acceder por línea de comandos; es decir, con este protocolo se logra conectar con servidores empleando la red internet como medio para las comunicaciones. Es sumamente beneficioso en vista de que siempre se realiza de manera segura, asegurando que la información que viaja a través de la red no es accesible ni legible por terceras personas, gracias a la encriptación de los datos, con lo cual el tráfico de datos se realiza de manera confidencial sin que nadie pueda robar información [5].

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

Esta herramienta en Linux es la que permite que un equipo pueda funcionar como servidor SSH, o sea, que sea capaz de aceptar conexiones remotas seguras mediante el protocolo SSH. Esta ofrece un conjunto amplio de funciones de tunelización segura, diversos métodos de autenticación, cifra todo el tráfico para evitar el robo de información, etc. [6].

## 8) Describa la tarjeta Raspberry Pi 5 en términos de: RAM, Procesador, Periféricos.

Primeramente, es importante conocer qué es una Raspberry Pi; es un ordenador con un tamaño comparable al de una tarjeta de video. Soporta distintos componentes de un ordenador, como procesadores, memoria RAM, entre otros.

Ahora, definiendo esta tarjeta en términos de RAM, es la que almacena datos temporales para tareas que se encuentran en curso, influyendo así en el rendimiento de la tarjeta. La Raspberry Pi 5 utiliza memoria **LPDDR4X-4267 SDRAM**, disponible en configuraciones de **2GB, 4GB, 8GB y 16GB**, que mejora el rendimiento en multitarea y procesamiento intensivo [7].

En términos de procesador, es el cerebro del dispositivo, siendo así el que se encarga de la ejecución de procesos y tareas. La Raspberry Pi 5 cuenta con un **procesador Broadcom BCM2712**, con CPU **quad-core Arm Cortex-A76** de 64 bits a **2.4GHz**, extensiones de criptografía para mayor seguridad, **caché L2 de 512KB por núcleo** y **caché L3 compartida de 2MB**. Además, integra una **GPU VideoCore VII** que soporta **OpenGL ES 3.1** y **Vulkan 1.2**, salida **dual 4Kp60 HDMI con HDR** y decodificación de **4Kp60 HEVC** [7].

Finalmente, en términos de periféricos, permite conectar componentes gracias a sus puertos de salida y entrada, tales como USB, HDMI, GPIO, entre otros. La Raspberry Pi 5 incluye **2 puertos USB 3.0, 2 puertos USB 2.0, Ethernet Gigabit con soporte PoE+** (requiere HAT adicional), ranura **microSD**, interfaces **MIPI de 4 carriles** para cámara/display, expansión **PCIe 2.0 x1** (requiere adaptador), conectividad **Wi-Fi 802.11ac, Bluetooth 5.0/BLE**, alimentación por **USB-C 5V/5A con Power Delivery**, **conector GPIO de 40 pines, RTC con batería externa y botón de encendido** [7].

## 9) Defina los siguientes conceptos de programación por hilos: Thread, lock, join y deadlock. Para explicar deadlock use algún diagrama o código de ejemplo.

A continuación se definen los conceptos solicitados:

- **Thread (hilo):** unidad de ejecución dentro de un proceso, que comparte memoria con otros hilos del mismo proceso [8].
- **Lock (candado):** mecanismo que regula el acceso a recursos compartidos para evitar errores de concurrencia [8].
- **Join:** operación que permite que un hilo espere hasta que otro finalice su ejecución [8].

- **Deadlock:** situación en la que dos o más hilos se bloquean mutuamente al esperar recursos que nunca se liberan [8].

Para ilustrar el concepto de *deadlock*, se presenta el siguiente diagrama, donde puede observarse que:

- El hilo A toma el Lock L1, pero solicita el Lock L2, el cual ya está ocupado por el hilo B.
- El hilo B toma el Lock L2, pero solicita el Lock L1, que ya está en uso por el hilo A.
- Como ambos esperan un recurso que nunca se libera, el sistema entra en un bloqueo circular.

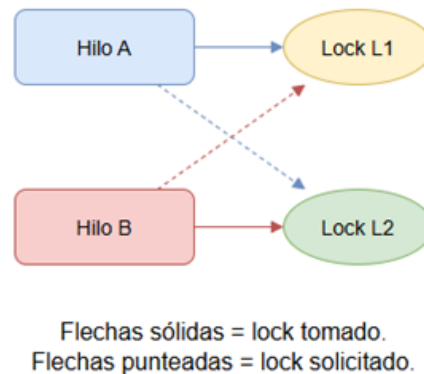


Figura 1: Ejemplo de deadlock entre dos hilos que compiten por dos locks.

## 10) Explique la diferencia entre un mutex lock y un read/write lock.

Un *mutex* es un mecanismo que se emplea en programación para evitar que varios subprocesos accedan a recursos compartidos simultáneamente. Una vez que un proceso ha adquirido un *mutex*, los otros subprocesos tienen que esperar a que libere el bloqueo antes de continuar; en otras palabras, es un bloqueo de exclusión mutua en donde solo un hilo a la vez puede tener el recurso protegido por un *mutex* [9].

En lo que respecta al *read/write lock*, es un bloqueo especial que permite que distintos subprocesos puedan leer recursos compartidos en simultáneo, pero bloquea a todos los demás subprocesos al escribir en recursos compartidos. Es decir, varios hilos pueden adquirir el *read lock* al mismo tiempo, pero solo un hilo puede tener el *write lock* [9].

Sabiendo esto, la diferencia radica en que un *mutex* se usa cuando cualquier acceso, ya sea de escritura o de lectura, requiere exclusión; mientras que, el *read/write lock* se emplea cuando hay muchas más lecturas que escrituras, dado que permite lecturas en simultáneo sin bloqueos.

## Capturas de pantalla solicitadas

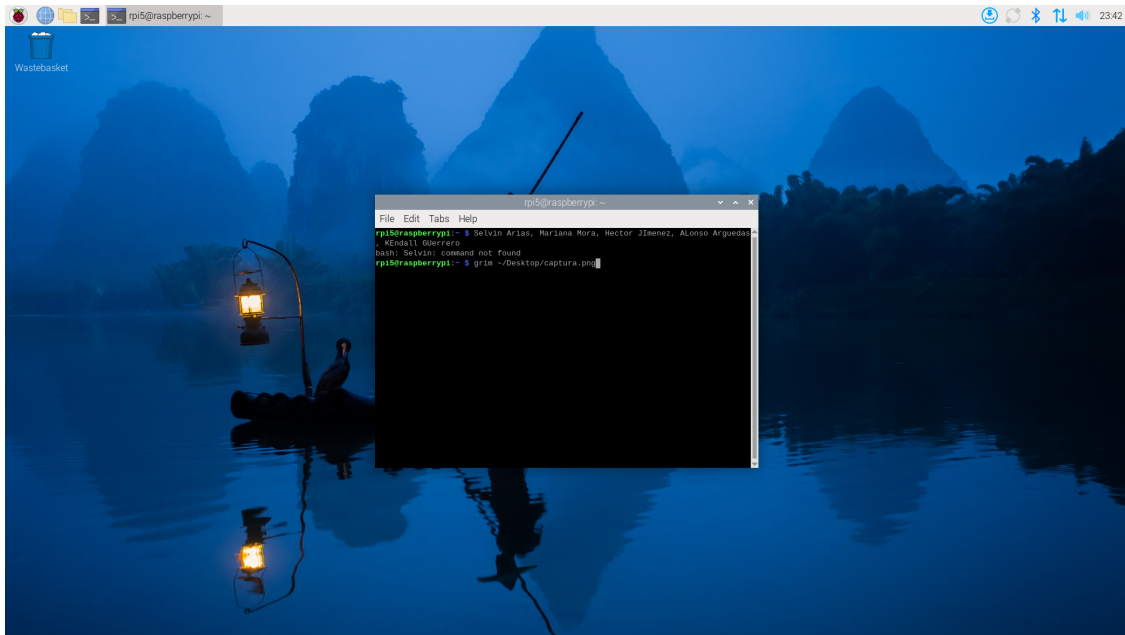


Figura 2: Captura de pantalla mostrando el arranque con Raspberry Pi Imager y los nombres de los integrantes.

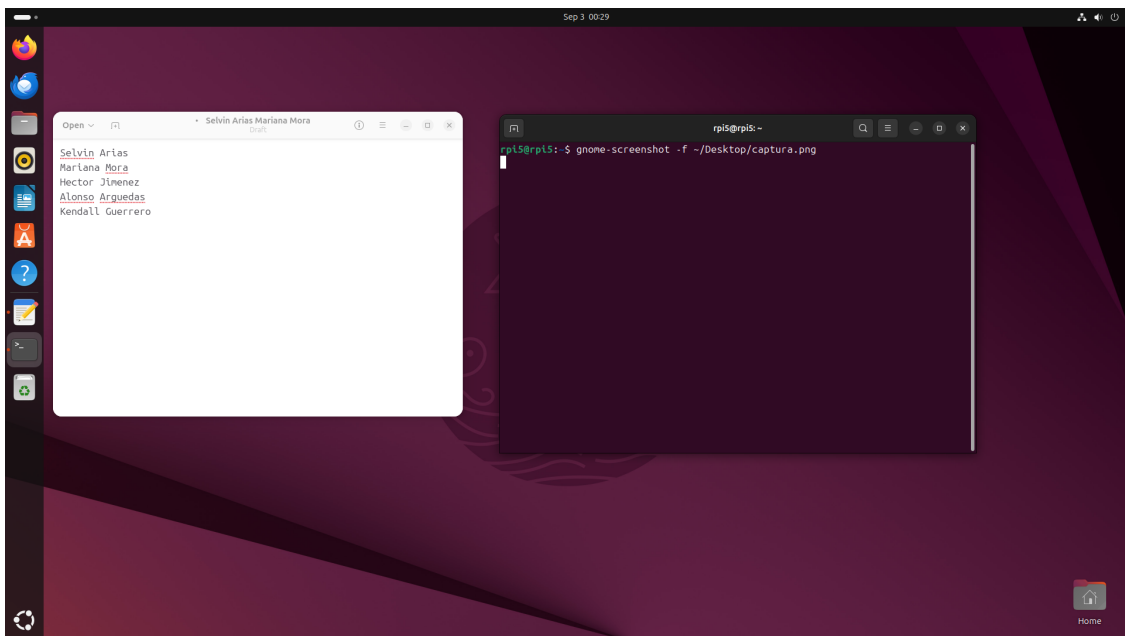


Figura 3: Captura de pantalla mostrando el arranque con la segunda imagen y los nombres de los integrantes.

## Referencias

- [1] Y. Hernández, “¿Qué es Shell en Linux? Guía rápida”, 2023. [Online]. Disponible en: <https://www.dongee.com/tutoriales/que-es-shell-en-linux-guia-rapida/>
- [2] Universidad de La República, “El compilador GCC”, s. f. [Online]. Disponible en: [https://www.fing.edu.uy/tecnoinf/mvd/cursos/eda/material/otros/compilador\\_GCC.pdf](https://www.fing.edu.uy/tecnoinf/mvd/cursos/eda/material/otros/compilador_GCC.pdf)
- [3] M. Dallas, “Cómo compilar código C/C++ con GCC”, 2020. [Online]. Disponible en: <https://medium.com/shared-bytes/c%C3%B3mo-compilar-c%C3%B3digo-c-c-con-gcc-667b70747f1a>
- [4] P. Ruiz, “El comando ifconfig en Ubuntu”, 2018. [Online]. Disponible en: <https://somebooks.es/comando-ifconfig-ubuntu/>
- [5] Cloudflare, “¿Qué es el protocolo Secure Shell (SSH)?”, s. f. [Online]. Disponible en: <https://www.cloudflare.com/es-es/learning/access-management/what-is-ssh/>
- [6] Red Hat Documentation, “Capítulo 12. Uso de comunicaciones seguras entre dos sistemas con OpenSSH”, s. f. [Online]. Disponible en: [https://docs.redhat.com/es/documentation/red\\_hat\\_enterprise\\_linux/8/html/configuring\\_basic\\_system\\_settings/using-secure-communications-between-two-systems-with-openssh\\_configuring-basic-system-settings](https://docs.redhat.com/es/documentation/red_hat_enterprise_linux/8/html/configuring_basic_system_settings/using-secure-communications-between-two-systems-with-openssh_configuring-basic-system-settings)
- [7] Raspberry Pi, “Raspberry Pi 5”, s. f.. [Online]. Disponible en: <https://www.raspberrypi.com/products/raspberry-pi-5/>
- [8] M. Kerrisk, *The Linux Programming Interface*. San Francisco, CA: No Starch Press, 2010.
- [9] Yifan, “What are mutexes and read-write locks? What’s the difference?”, s. f. [Online]. Disponible en: <https://yifan-online.com/en/km/article/detail/6679>