



Ejercicios de repaso y autoevaluación

1. ¿Qué es una llamada al sistema (*system call*)?

es un comando del kernel del sistema, que se ejecute para que realice alguna tarea

2. Marque con una X las características correctas de HTTP.

✓	El conjunto de llamadas al sistema puede variar dependiendo del sistema operativo.
✓	Las llamadas al sistema pueden ponerse a disposición como funciones de bibliotecas.
	Los manejadores del sistema hacen uso de las bibliotecas.
	Toda llamada al sistema se puede ejecutar en modo usuario.
✓	Toda llamada al sistema solo se puede ejecutar en modo <i>kernel</i> .

3. ¿De cuántas formas se puede realizar una llamada al sistema?

- ✓ a. De dos formas: directa o mediante API.
- b. De tres formas: directa, mediante API o por red.
- c. Solo de forma directa.
- d. Solo mediante API.

4. ¿Cuáles de los siguientes tipos no pertenece a una categoría principal de llamadas al sistema?

- ✓ a. Llamadas de información.
- b. Llamadas para la gestión de la memoria.
- c. Llamadas para la gestión de dispositivos.
- d. Llamadas para la gestión de comunicaciones de red.

5. Complete los siguientes enunciados.

Las tres grandes API que envuelven llamadas al sistema son: Win32 API para *Windows*, POSIX API para sistemas basados en *POSIX* (incluyendo todas las versiones *Unix*, *Linux* y *macOS*), y Java API para la máquina virtual de Java.

En sistemas operativos modernos, se ejecutan Múltiples procesos a la vez.

Existe una llamada al sistema muy importante en *Unix* que permite que un programa Cree un nuevo proceso a partir del que se está ejecutando. Esta llamada al sistema es fork() .

6. Ordene la siguiente secuencia de pasos que se produce cuando se realiza una llamada al sistema.

- 7 a. El *kernel* devuelve al manejador de la llamada al sistema, el código almacenado de esa llamada que corresponde al número que se encuentra en el registro del procesador
- 4 b. Después realiza la llamada a la función.
- 1 al 3 c. El proceso de usuario introduce en la pila los valores de los parámetros de la función que envuelve la llamada al sistema.
- 8 d. Se ejecuta la llamada al sistema.
- 10 e. Acaba la ejecución de la función que envuelve la llamada al sistema
- 9 f. Cuando acaba la ejecución, el manejador devuelve el control de ejecución al proceso que realiza la llamada. Conmutando al modo usuario.
- 11 g. Se actualiza la pila de ejecución al estado anterior a la llamada.
- 5 h. Esa función coloca en el registro del procesador el número que identifica la llamada al sistema específica.
- 6 i. Después se ejecuta la interrupción o trap que provoca la ejecución en modo *kernel*.

7. ¿Qué es una API?

Application Program Interface, es una lista de comandos para interactuar con el kernel

8. ¿Cuáles son las tres grandes API que envuelven llamadas al sistema?

- ✓ 1. Win32 API para *Windows*.
- ✓ 2. POSIX API para sistemas *Unix*, *Linux* y *macOS*.
- 3. Cg API.
- ✓ 4. Java API.

9. ¿Cuál es la sentencia que realiza la llamada directa a una *system call*?

- ☒ a. syscall.
- b. syscalls.
- c. recall.
- d. call.

10. De las siguientes afirmaciones, indique cuál es verdadera o falsa.

a. exit es una llamada al sistema que finaliza la ejecución de un proceso en *Linux*.

- ☒ Verdadero
- ☐ Falso

b. CreateProcess es una llamada al sistema que crea un nuevo proceso en *Windows*.

- ☒ Verdadero
- ☐ Falso

c. CreateThread no es una llamada al sistema.

- ☐ Verdadero
- ☒ Falso

d. CreateThread solo crea dos hilos de ejecución cuando se llama desde *Windows*.

- ☐ Verdadero
- ☒ Falso

11. ¿Qué dos llamadas al sistema son capaces de leer y escribir en fichero un número de bytes?

- ☒ a. read y write.
- b. readr y readw.
- c. lseek y close.
- d. Todas las opciones son incorrectas.

12. De las siguientes afirmaciones, indique cuál es verdadera o falsa.

a. Semáforos y tuberías son dos mecanismos de comunicación entre procesos.

- ☒ Verdadero
- ☐ Falso

b. Los *sockets* son un mecanismo de comunicación entre procesos que no permite la comunicación por red.

- ☐ Verdadero
☒ Falso

c. La memoria compartida se utiliza solo por dos procesos como máximo en la comunicación.

- ☐ Verdadero
☒ Falso

13. ¿Qué es *Win32 API*?

Es la Api de windows

14. ¿Qué es *GCC*?

El compilador de C de linux

15. ¿Qué es *GAS*?

El ensamblador de Linux

9. ¿Cuál es la sentencia que realiza la llamada directa a una *system call*?

- a. syscall.
- b. syscalls.
- c. recall.
- d. call.

10. De las siguientes afirmaciones, indique cuál es verdadera o falsa.

- a. exit es una llamada al sistema que finaliza la ejecución de un proceso en *Linux*.
 - ☐ Verdadero
 - ☐ Falso
- b. CreateProcess es una llamada al sistema que crea un nuevo proceso en *Windows*.
 - ☐ Verdadero
 - ☐ Falso
- c. CreateThread no es una llamada al sistema.
 - ☐ Verdadero
 - ☐ Falso
- d. CreateThread solo crea dos hilos de ejecución cuando se llama desde *Windows*.
 - ☐ Verdadero
 - ☐ Falso

11. ¿Qué dos llamadas al sistema son capaces de leer y escribir en fichero un número de bytes?

- a. read y write.
- b. readr y readw.
- c. lseek y close.
- d. Todas las opciones son incorrectas.

12. De las siguientes afirmaciones, indique cuál es verdadera o falsa.

- a. Semáforos y tuberías son dos mecanismos de comunicación entre procesos.
 - ☐ Verdadero
 - ☐ Falso