


Comenzado el martes, 19 de marzo de 2019, 10:23**Estado** Finalizado**Finalizado en** martes, 19 de marzo de 2019, 10:24**Tiempo empleado** 15 s**La puntuación** 8,00/8,00**Calificación** 10,00 de 10,00 (100%)**Pregunta 1**

Correcta

Puntúa 1,00 sobre 1,00

 Marcar pregunta

¿con qué funciones se puede acceder a variables sencillas del espacio de usuario desde el kernel?


Selecciona una:

- ☐ a. write_user() y read_user()
- ☐ b. write y read
- ☒ c. get_user() y put_user() ✓
- ☐ d. trae_paca() y ahi_vaeso()

Pregunta 2

Correcta

Puntúa 1,00 sobre 1,00

 Marcar pregunta

Las llamadas al sistema son:


Selecciona una:

- ☐ a. cualquier funcion que, como programadores, usamos desde nuestro programa para acceder a los recursos del sistema
- ☐ b. cualquier funcion del kernel del sistema operativo
- ☐ c. los mensajes que muestra el sistema en sus logs
- ☒ d. El interfaz para acceder a los servicios del kernel del sistema operativo ✓

Pregunta 3

Correcta

Puntúa 1,00 sobre 1,00

 Marcar pregunta

El comando strace nos sirve para (comprobar su utilidad si no se ha hecho):


Selecciona una:

- ☒ a. ver las llamadas al sistema y señales en la ejecución de un proceso ✓
- ☐ b. ver la traza de accesos a memoria de un programa
- ☐ c. mostrar la traza de accesos a disco de un programa
- ☐ d. ver las llamadas al sistema en la ejecución de un proceso

Pregunta 4

Correcta

Puntúa 1,00 sobre 1,00

 Marcar pregunta

En nuestro ejercicio la tabla de punteros a las diferentes implementaciones de las llamadas al sistema estaba en el fichero llamado:

Selecciona una:

- ☒ a. calls.S ✓
- ☐ b. syscall.h
- ☐ c. inittab
- ☐ d. fstab

Cuando queremos acceder desde el kernel a un buffer en el área de memoria de usuario (está marcado con `*__user` función del kernel) usaremos:

Selecciona una:

- ☐ a. No podemos acceder al espacio de usuario de ninguna manera
- ☒ b. usaremos funciones del estilo `copy_from_user()` y `copy_to_user()` ✓
- ☐ c. Usaremos funciones del estilo `read_proc()` y `write_proc()`
- ☐ d. Usaremos funciones del estilo `read()` y `write()`

Cuando compilamos un módulo de kernel cargable (LKM) con el makefile que hemos usado, obtenemos un archivo

Selecciona una:

- ☐ a. `.lib`
- ☐ b. `.lkm`
- ☒ c. `.ko` ✓
- ☐ d. `.mod`

Selecciona una:

- ☒ a. Con la función `create_proc_entry()` ✓
- ☐ b. Con la función `module_init()`
- ☐ c. Con la función `copy_from_user()`
- ☐ d. Con la función `open(... , O_CREATE)`

Al escribir un módulo de kernel cargable (LKM) ¿qué función o funciones debemos implementar obligatoriamente?

Selecciona una:

- ☐ a. Las funciones `open()` y `close()` para abrir y cerrar el módulo
- ☐ b. Las funciones `read_proc()` y `write_proc()` para comunicarnos con el módulo
- ☒ c. Una función de inicialización y otra para finalización o descarga del módulo ✓
- ☐ d. la función `main()`

¿Qué comandos de linux hemos usado para cargar (instalar) y descargar nuestros módulos?

Selecciona una:

- ☐ a. `mount` y `umount`
- ☐ b. `load` y `unload`
- ☐ c. `module_init` y `module_exit`
- ☒ d. `insmod` y `rmmod` ✓

19/3/2019
Pregunta 5

Correcta

Puntúa 1,00 sobre 1,00

🚩 Marcar pregunta

CUESTIONARIO: Agregar nuevas llamadas al sistema
Si invocamos a una llamada al sistema que no está implementada:
`retorno = syscall(500);`

¿qué valor obtendríamos como retorno (probarlo si no se ha hecho)?

Selecciona una:

- ☒ a. -38 ✓
- ☐ b. 500
- ☐ c. -500
- ☐ d. -1
- ☐ e. -EFAULT

Pregunta 6

Correcta

Puntúa 1,00 sobre 1,00

🚩 Marcar pregunta

la herramienta make sirve para (escoge solo una respuesta):

Selecciona una:

- ☐ a. configurar el kernel
- ☐ b. instalar el kernel
- ☒ c. ejecutar las reglas definidas en el fichero Makefile ✓
- ☐ d. compilar el kernel de linux

Pregunta 7

Correcta

Puntúa 1,00 sobre 1,00

🚩 Marcar pregunta

tras compilar el kernel (make), el fichero con la imagen del kernel se debe instalar en directorio de la raspberry:

Selecciona una:

- ☐ a. /tmp
- ☐ b. /kernel
- ☒ c. /boot ✓
- ☐ d. /root

Pregunta 8

Correcta

Puntúa 1,00 sobre 1,00

🚩 Marcar pregunta

¿Para qué ponemos el modificador `asm` linkage en las funciones que implementan las llamadas al sistema (las que hemos colocado en `sys_arm.c`)?

Selecciona una:

- ☐ a. no lo sé
- ☐ b. para que los parámetros se pasen por la pila (no por registro)
- ☒ c. Cualquiera de las repuestas es cierta ✓
- ☐ d. por que lo ha dicho el profe