

Sistemas Operativos

Departamento de Computación – FCEyN – UBA
Segundo cuatrimestre de 2022

Nombre y apellido: _____

Nº orden: _____ L.U.: _____ Cant. hojas: _____

Segundo parcial – 17/11 - Segundo cuatrimestre de 2022

1	2	3	4	Nota

- Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas entregadas.
- Entregue esta hoja junto al examen, la misma **no** se incluye en la cantidad total de hojas entregadas.
- Cada ejercicio debe realizarse en hojas separadas y numeradas. Debe identificarse cada hoja con nombre, apellido y LU.
- Cada código o pseudocódigo debe estar bien explicado y justificado en castellano. ¡Obligatorio!
- La devolución de los exámenes corregidos es personal. Los pedidos de revisión se realizarán por escrito, antes de retirar el examen corregido del aula.
- Los parciales tienen tres notas: I (Insuficiente): 0 a 59 pts, A- (Aprobado condicional): 60 a 64 pts y A (Aprobado): 65 a 100 pts. No se puede aprobar con A- ambos parciales. Los recuperatorios tienen dos notas: I: 0 a 64 pts y A: 65 a 100 pts.

Ejercicio 1. Sistemas de Archivos (30 puntos)

Se tiene un disco formateado con FAT para el cual se quiere poder devolver el contenido de un archivo a partir de su ruta absoluta. Para ello se debe implementar la función:

```
datos = cargar_archivo(directorios[])
```

donde `directorios` es la lista de los nombres de los directorios de la ruta (ordenados, incluyendo el nombre del archivo, y sin incluir a `root`). Es decir, si el archivo a abrir es `\Documentos\Users\foto.png` entonces `directorios = ['Documentos', 'Users', 'foto.png']`.

Para ello se cuenta con las siguientes funciones auxiliares:

- `FAT_entry(block_address)` que devuelve la entrada de la tabla FAT de la posición `block_address`.
- `raw_data = read_blocks(block_address1, block_address2, ...)` que lee del disco todos los bloques indicados por parámetro, en orden.
- `parse_directory_entries(raw_data)` que devuelve una lista de `struct_entrada_directorio` que representan las entradas de directorio del directorio pasado en `raw_data`.
- `raw_data = root_table()` que devuelve los datos de la tabla de directorios de `root`.

- Enumere tres campos que debe tener, según este tipo de filesystem, la estructura `struct_entrada_directorio`.
- Escribir el pseudo-código de la función `cargar_archivo`.

Ejercicio 2. Sistema de E/S - Drivers (25 puntos)

Nos piden desarrollar el driver para el nuevo *Palopalooza*. El mismo recibirá desde un servidor web las canciones que los clientes piden. Luego el sistema se encargará de buscar el mp3 de la canción y enviar los bytes de la canción hacia el sistema de parlantes. El driver se encargará de enviar dicha música hacia alguna de las pistas disponibles (es decir que no estén pasando música). Se cuenta con 3 pistas (y 3 sistemas de parlantes) y no se quiere que ninguna esté sin música si hay pedidos de usuarios.

Para esto, las canciones se procesan secuencialmente por orden de llegada (i.e. *FIFO*), pudiendo procesarse hasta 3 trabajos en simultáneo, uno por cada pista. Los trabajos consisten en una secuencia de comandos comenzando con la syscall `open` hasta el llamado a la syscall `close`. Si ya hay 3 trabajos abiertos y no cerrados, el próximo quedará bloqueado en la syscall `open`.

Mientras no se llame a la syscall `close`, un proceso puede enviar más bytes al dispositivo (nuevos llamados a la syscall `write`). Al hacer `close`, la canción continuará siendo reproducida hasta su finalización.

Se cuenta con la siguiente API para operar con el dispositivo de E/S, que deberá programar en un driver.

<code>int open(int device_id)</code>	Abre el dispositivo.
<code>int close(int device_id)</code>	Cierra el dispositivo.
<code>int write(int device_id, char* data, int cantidad)</code>	Escribe el valor en el dispositivo <code>device_id</code>
<code>int driver_init()</code>	Durante la carga del SO
<code>int driver_remove()</code>	Durante la descarga del SO

Todas las operaciones retornan la constante `IO_OK` si fueron exitosas o la constante `IO_ERROR` si ocurrió algún error. Además en el kernel se cuenta con la estructura de datos `current`, a partir de la cual puede obtenerse el *PID* del proceso en ejecución llamando a la función `pid = task_pid_nr(current)`.

Para la programación de un driver, se dispone de las siguientes syscalls:

<code>void OUT(int IO_address, char data);</code>	Escribe data en el registro de E/S (1 único byte)
<code>int IN(int IO_address);</code>	Retorna el valor almacenado en el registro de E/S
<code>int request_irq(int irq, void* handler)</code>	Permite asociar el procedimiento handler a la interrupción irq que es llamado por el equipo de música al terminar de escribir el byte enviado. Retorna <code>IRQ_ERROR</code> si ya está asociada a otro handler
<code>int free_irq(int irq)</code>	Libera la interrupción irq del procedimiento asociado.

Se tienen a su vez las siguientes direcciones y valores.

- `EQUIPO_IRQ_X`, con X de 1 a 3, el valor del IRQ del equipo X.
- `ADDRESS_X`. Es la dirección a la que hay que escribir para enviar un byte al equipo.
- `CONTROL_X`. Leyendo este registro (con X, 1 a 3) se puede ver si el equipo X está libre (devuelve 0) u ocupado (devuelve 1).

Puede suponer que los equipos no se utilizan antes de cargar el driver (pero podrían no estar listos al inicio). Además cuentan con el tipo de datos cola, con las funciones típicas (push, pop, peek y length) y su crecimiento y uso de memoria es automático.

Tenga en cuenta que las llamadas a `write` pueden bloquear un tiempo (encolado de operaciones o alguna operación `OUT`) pero no durante la escritura (`OUT`) completa del buffer enviado a `write`.

Ejercicio 3. Sistemas distribuidos: (20 puntos)

En una variante descentralizada del protocolo Two Phase Commit, los participantes se comunican directamente uno con otro en vez de indirectamente con un coordinador

- Fase 1: El coordinador manda su voto a todos los participantes
- Fase 2: Si el coordinador vota que no, los participantes abortan su transacción. Si vota que si, cada participante manda su voto al coordinador y al resto de participantes donde cada uno decide sobre el resultado acorde a el voto que le llega y lleva a cabo el procedimiento.

- a) ¿Qué ventajas y desventajas encuentra esta variante con respecto a la variante centralizada? Hablar con respecto a la cantidad de mensajes, tolerancia a fallos, etc
- b) ¿En qué casos usaría cada versión del protocolo?

Ejercicio 4. Seguridad: (25 puntos)

Dado el código a continuación:

```

1 void imprimir_habilitado(const char *nombre_usuario, const char* clave, const char * imprimir, int tam_imprimir) {
2     char *cmd = malloc(tam_imprimir+5 * sizeof(char));
3     if (cmd == NULL) exit(1);
4     if (usuario_habilitado("/etc/shadow", nombre_usuario, clave)) {
5         snprintf(cmd, tam_imprimir+4, "echo %s", imprimir);
6         system(cmd);
7     } else {
8         printf("El usuario o clave indicados son incorrectos.");
9         assert(-1);
10 }

```

El objetivo de la función es imprimir por pantalla el texto enviado como parámetro por el usuario, siempre y cuando el nombre de usuario y clave provistos por dicho usuario sean correctos.

Para esto se cuenta con la función `usuario_habilitado` que se sabe funciona correctamente y no cuenta con problemas de seguridad. La misma utiliza strings terminados en caracter nulo (`\0`) y lee el archivo provisto de contraseñas (encriptadas) de todos los usuarios del sistema, que puede ser sólo leído por `root`, devolviendo un booleano indicando si el usuario y la clave ingresados se encuentran en dicho archivo.

- a) Indique si es necesario que el programa corra con algún nivel específico de permisos. Justifique en qué líneas y porqué.
- b) Indique dos problemas de seguridad que podrían surgir (hint: tenga en cuenta el ítem anterior).
- c) Indique alguna manera (valuación de los parámetros) de poder explotar cada una de las vulnerabilidades mencionadas.
- d) Indique el impacto de las vulnerabilidades mencionadas, analizándolas según los tres requisitos fundamentales de la seguridad de la información.
- e) Para cada vulnerabilidad, proponga una solución, indicando cómo modificaría el código en caso de corresponder.