

Apartado	1	2	3	4	5	6	7
Puntuación							

No Presentado

EXAMEN DE SISTEMAS OPERATIVOS (Grado en Ing. Informática), Julio 2015.

APELLIDOS Y NOMBRE:

Justificar todas las respuestas. Poner apellidos y nombre en todas las hojas y graparlas a este enunciado. Tiempo total = **2 horas 30 minutos**.

1. **(1.5 puntos)** En un sistema de archivos (tipo `system V`) tenemos un archivo (`datos`) del usuario `juan`, que es su propietario, y cuyo camino absoluto es: `/home/juan/so/practicas/datos`. Responder a lo siguiente (**Cada respuesta solo se puntuará si es correcta**):

- (a) **(0.4 puntos)** Calcular el número mínimo de accesos a disco que serán necesarios en la operación: `open("/home/juan/so/practicas/datos", O_RDONLY)` para obtener el inodo de `datos` si las entradas de los diferentes subdirectorios se encuentran siempre en el segundo bloque del directorio padre, excepto `so` que se encuentra en el quinto y el archivo `datos` que se encuentra en el tercero. Se supone que las cachés de datos e inodos están inicialmente vacías.

Núm. mínimo de accesos a disco:	
Núm. mínimo de accesos al área de datos:	
Núm. mínimo de accesos a la lista de inodos:	

- (b) **(0.4 puntos)** El sistema de archivos tiene un tamaño de bloque de 2 Kbytes e inodos con 12 direcciones directas de bloques, una indirecta simple, una indirecta doble y una indirecta triple. Además, utiliza direcciones de bloques de 8 bytes. Calcula cuántos bloques de disco son necesarios para representar el archivo `datos` cuando su tamaño es de 2563 KBytes. Discriminar cuántos bloques son de datos y cuántos de índices.

Número de bloques de datos: Número de bloques de índices:

- (c) **(0.3 puntos)** Calcular qué número de bloque lógico en el sistema de ficheros corresponde al inodo del raíz (se comienza a contar en el bloque lógico 0), y cuál bloque lógico al inodo del fichero `datos`, suponiendo lo siguiente:

- El número de inodo del `/` es el 2, y al fichero `datos` le corresponde el inodo núm. 263 (los inodos se comienzan a numerar a partir de 1).
- El tamaño de un inodo es de 64 bytes.
- El boot ocupa un bloque y el *superbloque* 8 bloques.

Número bloque lógico inodo `/` =

Número bloque lógico inodo fichero `datos` =

- (d) **(0.4 puntos)** Supongamos la siguiente secuencia de comandos desde el directorio `/home/juan/so/practicas`:

1. Se crea un *hard link* al fichero `datos` (cuyo número de inodo es 11545, tamaño 2563 KBytes y num. de hard links=3):

```
ln datos hlinkdatos /* crea hard link hlinkdatos */
```

2. Posteriormente se crea un *soft link* al archivo `hlinkdatos`:

```
ln -s hlinkdatos slink /* el comando ls -l mostraría slink -> hlinkdatos */
```

3. Finalmente se crea un *hard link* al fichero `slink`:

```
ln slink hlinkslink /* crea hard link hlinkslink */
```

Contestar a lo siguiente:

A. Indicar cuál es el número de (hard) links del fichero `hlinkslink` después de las tres operaciones:

B. Indicar el tamaño del fichero `hlinkslink`:

C. Una vez creados los ficheros `slink` y `hlinkslink`, al borrar el fichero `datos` (`rm datos`) se puede seguir accediendo al contenido del fichero con número de inodo 11545 a través del link simbólico `slink` (Cierto/Falso)

D. Con un sistema de ficheros Unix basado en registro (*journaling file system*), al borrar el fichero `datos`, primero se realiza una copia de su inodo en el registro antes de liberar sus bloques del área de datos. (Cierto/Falso)

2. (0.4 puntos) Que el kernel de UNIX sea *reentrante* significa que ...

- a) que en su código puede haber funciones recursivas
- b) que sus datos no pueden ser compartidos
- c) que no hay pila del kernel
- ⇒ d) que puede ser ejecutado concurrentemente por varios procesos
- e) que no puede ser interrumpido por interrupciones
- f) que las funciones recursivas no accedan a variables globales
- g) ninguna de las anteriores... ¿qué significa entonces?

Reentrante significa que varios procesos puede estar ejecutando concurrentemente la misma o distinta función del kernel; por tanto d)

3. (0.4 puntos) ¿Cuáles de las siguientes condiciones son necesarias para que el kernel de unix sea reentrante?

- a) que se ejecute con IPL muy elevado
- b) que se ejecute con IPL máximo
- ⇒ c) que el código del kernel sea de solo lectura
- d) que los procesos de usuario no tengan mayor prioridad que los del kernel
- ⇒ e) que los datos del kernel estén protegidos de accesos concurrentes
- f) que las prioridades de los procesos estén representadas por núms. menores que las IPLs
- g) que cada proceso tenga su propia copia de los datos del kernel
- ⇒ h) que cada proceso tenga su propia pila del kernel
- i) que la pila del kernel sea de solo lectura
- j) que en lugar de funciones recursivas se usen bucles en la programación del kernel
- k) que los datos del kernel sean de solo lectura
- l) que las funciones recursivas solo tengan variables locales
- m) que las funciones recursivas solo tengan variables locales y punteros
- n) que las funciones recursivas no usen punteros

Condiciones necesarias: código de solo lectura, datos protegidos de accesos concurrentes y cada proceso su propia pila del kernel; por tanto c), e) y h)

4. (0.7 puntos) ¿Qué es un fichero `setuid`? ¿para qué sirve? Póngase un ejemplo

Fichero `setuid`: ejecutable que cuando un proceso hace alguna de las llamadas `exec` sobre él, `exec` cambia la credencial efectiva (y la salvada) de usuario del proceso que ha hecho `exec`, poniéndola igual al propietario de dicho fichero `setuid`.

Se consigue activado el bit `setuid` (04000) en los permisos del fichero. Por ejemplo poniéndole permisos 04755 (`rwsr-xr-x`)

5. (**total 1.2 puntos**) Considere la traslación de direcciones virtuales a físicas en un procesador con direcciones virtuales de 32 bits y páginas son de 4 KB (4 kilobytes). La arquitectura es de paginación multinivel (2 niveles), las tablas de páginas en todos los niveles son de tamaño 1 página y los bits de las direcciones virtuales que componen el número de página virtual se trocean en dos campos del mismo tamaño que se usan para obtener la entrada en la tabla de páginas correspondiente. Conteste a las siguientes cuestiones, si en alguna pregunta cree que no hay datos suficientes para contestar, conteste **FD** (Faltan Datos).

- (**0.4 puntos**) Dibuje un esquema de esta arquitectura.

Consulte las slides estudiadas en clase. Puede estudiar el caso del Intel Core i7 y reducir esa arquitectura a dos niveles y con los datos de este problema.

- (**0.1 puntos**) Tamaño en bytes del espacio de direcciones virtuales:
Tamaño en bytes del espacio de direcciones físicas: No se puede saber el tamaño del espacio de direcciones físicas porque no conocemos el número de bits reservado en las entradas de la tabla de páginas para direccionar el número de página física.
- (**0.1 puntos**) Número de páginas del espacio de direcciones virtuales:
Número de páginas del espacio de direcciones físicas: El mismo comentario que antes con respecto al número de páginas físicas.
- (**0.1 puntos**) Número de entradas la tabla de páginas de primer nivel:
Número de entradas en una tabla de páginas de segundo nivel:
- (**0.1 puntos**) Tamaño en bits de una entrada en la tabla de páginas de 1er nivel:
Tamaño en bits de una entrada la tabla de páginas de segundo nivel:
- (**0.2 puntos**) Considere ahora un proceso que sólo utiliza 12 MegaBytes (MB) de su espacio de direccionamiento en la parte baja de sus direcciones y 4 MB en la parte alta. ¿Cuanto espacio en bytes ocupa este proceso en tablas de página? Cada tabla de páginas de segundo nivel tiene 2¹⁰ entradas con lo que puede direccionar 2¹⁰ x 2¹² = 4MB. Por tanto se requieren 5 tablas de páginas: la de primer nivel y cuatro de segundo nivel, para un total de 20KB en tablas de páginas.
- (**0.1 puntos**) Considere ahora un procesador con direcciones virtuales de 32 bits, páginas de 4KB y entradas de 4 bytes en una tabla de páginas de un nivel. Si a un proceso se le asigna todo el espacio direccionable, ¿Cuál es el tamaño en bytes de la tabla de páginas? Hay 2²⁰ páginas de 4KB, y por tanto 2²⁰ entradas en la TP, cada una de 4 bytes, lo que lleva a una TP de 2²² = 4MB
- (**0.1 puntos**) Para el procesador del apartado anterior considere ahora el proceso que sólo utiliza 12 MegaBytes (MB) de su espacio de direccionamiento en la parte baja de sus direcciones y 4 MB en la parte alta. Considerando que la TP tiene que tener entradas para todo el rango de direcciones virtuales, es decir, es la TP calculada en el apartado anterior, ¿que porcentaje de las entradas de la TP estarían marcadas como válidas para este proceso? 16MB son 2¹² páginas de 2¹² bytes. Por tanto se ocuparían 4096 entradas del total de 2²⁰ entradas, es decir 2⁻⁸ x 100 = 0,39 %

6. (total 0.8 puntos) Considere los algoritmos de reemplazo de página FIFO y LRU, y la secuencia de referencias a páginas 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5. Considere **todos** los fallos de página que se pueden producir aunque no impliquen reemplazo. Muestre una tabla con la asignación de páginas a frames (marcos) en cada referencia e indique el número de fallos de página.

▪ FIFO y 3 frames. Número de fallos:

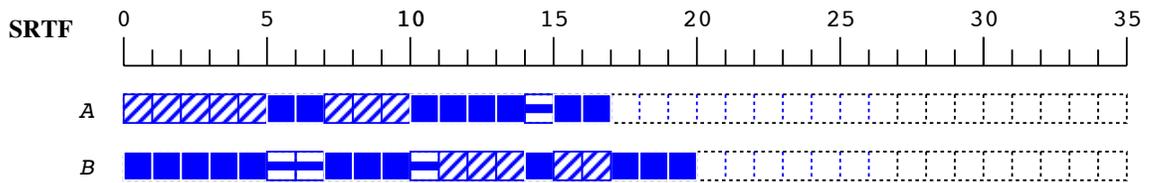
▪ FIFO y 4 frames. Número de fallos:

Es la anomalía de Belady que se comentó en clase y que es una propiedad no deseable de ese algoritmo.

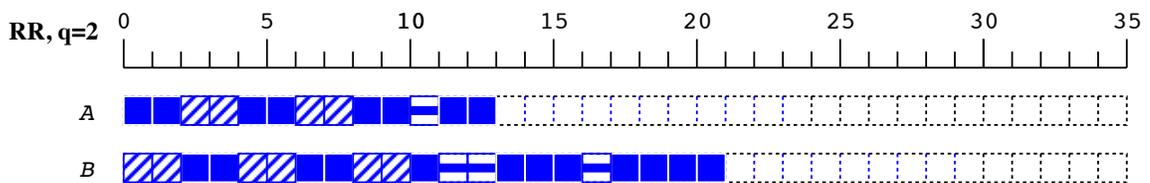
▪ LRU y 3 frames. Número de fallos:

▪ LRU y 4 frames. Número de fallos:

7. (1 punto) Tenemos dos procesos, A y B , que inician ambos su ejecución en el instante $T = 0$, pero A llega a la cola antes que B . Las secuencias de ráfagas de cada proceso son $A = (\underline{6}, 1, \underline{2})$, $B = (\underline{5}, 2, \underline{3}, 1, \underline{4})$ donde los números subrayados representan tiempo de CPU y el resto es E/S (los tiempos se miden en ms). Rellene en las plantillas el resultado de aplicar los algoritmos SRTF y Round Robin con $q = 2 ms$. Cada celda representa $1 ms$ y se debe indicar qué le sucede al proceso en cuestión en cada momento (CPU, E/S o en espera) usando la simbología indicada abajo. Calcule en ambos casos el tiempo de retorno medio.



Tiempo de retorno medio (SRTF) =



Tiempo de retorno medio (RR, q=2) =

8. (1 punto) Hablando de operaciones de entrada/salida...

- a) (0.5 puntos) Indíquese qué sucede (¿Se escribe en pantalla?, ¿Se escribe en el fichero "input.txt"?, ¿otras posibilidades?) cuando el siguiente programa ejecuta `printf("prueba")` (l.7). Justifíquese brevemente su respuesta.

```
1.1: #include "includes.h"
1.2: int main(){
1.3:   int fd1= open("input.txt",O_RDONLY); //existe en el directorio actual
1.4:   close(STDOUT_FILENO);               //equivale a close(1)
1.5:   close(STDERR_FILENO);              //equivale a close(2)
1.6:   dup (fd1);
1.7:   printf("prueba");
1.8:   exit(0);
1.9: }
```

El programa no escribe nada por pantalla ni en el fichero "input.txt". El motivo radica en que el "printf" quiere escribir "prueba" en el fichero "input.txt", que está abierto como de sólo lectura. Nótese que esto se debe a que en las líneas 4 y 6 se trata de redireccionar la salida estándar al fichero "input.txt" (pero no es posible escribir en él por estar abierto en modo de sólo lectura)

- b) (0.5 puntos) Indíquese con qué método(s) de entrada/salida (polling, E/S por interrupciones o E/S por DMA) se relacionan las siguientes sentencias:

- Se genera una interrupción al finalizar una transferencia.

E/S por interrupciones y E/S por DMA.

- Es la CPU la que realiza la transferencia de datos cuando estos están listos.

polling y E/S por interrupciones.

- La CPU debe suministrar al controlador el tipo de operación (lectura o escritura), la dirección de transferencia de datos y cantidad de bytes a transferir.

E/S por DMA

- Durante la transferencia de varios datos, la sincronización se consigue al preguntar la CPU al dispositivo si está listo.

polling