



Presentación

Esta PEC plantea una serie de actividades con el objetivo que el estudiante se familiarice con la temática de los primeros módulos de la asignatura.

Cada pregunta indica una posible temporización para poder acabar la PEC antes de la fecha límite así como el peso de la pregunta en la evaluación final.

Competencias

Transversales:

- Capacidad para la comunicación escrita en el ámbito académico y profesional

Específicas:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo

Enunciado

1. Módulo 2 [Del 31 de octubre al 3 de noviembre] (Peso 30 % = 10 % + 10 % + 10 %)

Responded **justificadamente** a las siguientes preguntas relacionadas con el módulo 2 de la asignatura:

- 1.1 Indicad qué mecanismo de entrada al SO (interrupción hardware, llamada al sistema o excepción) provocan los siguientes eventos:

- Un proceso invoca la rutina `printf`.

Provocarí una llamada al sistema, invocada indirectamente por la rutina `printf` de la biblioteca `libc`.

- Un proceso de usuario intenta ejecutar una instrucción privilegiada de lenguaje máquina.

Provocarí una excepción, por intentar ejecutar una instrucción privilegiada desde modo usuario.

- Un proceso invoca la llamada al sistema `unlink` (la que permite borrar un fichero) para solicitar el borrado de un fichero de la su propiedad.



Provocaría una llamada al sistema para solicitar al SO el borrado de un fichero que, al ser de propiedad del usuario, podrá ser borrado.

- Un proceso invoca la llamada al sistema `unlink` (la que permite borrar un fichero) para solicitar el borrado de un fichero sobre el que no tiene permiso de borrado.

Provocaría una llamada al sistema que, cuando el código de atención a la llamada al sistema detecte que corresponde a una petición no realizable, devolverá error pero no generará una excepción. El proceso continuará con su ejecución sin que el SO haga nada para intentar solucionarlo.

- El controlador de disco notifica que la última petición de lectura ya está disponible.

Provocaría una interrupción hardware para notificarlo al SO y así permitir que el proceso que la esperaba pase al estado *Ready*.

- Un proceso intenta acceder a una posición de memoria que se encuentra en el área de swap.

Provocaría una excepción de tipo fallo de página para que el código de atención traiga a memoria física la página donde se ubica el dato requerido.

1.2 ¿Qué cambio(s) de estado (*Run*, *Ready*, *Blocked*) provocarían los siguientes eventos sobre los procesos del sistema:

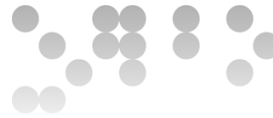
- Un proceso invoca la llamada al sistema que permite leer del disco pero en estos momentos el disco está ocupado atendiendo otra petición.

El proceso pasa al estado *Block* hasta que finalice la entrada/salida. Un proceso que haya sido seleccionado por el planificador pasará de *Ready* a *Run*.

- El planificador decide que un proceso en ejecución ha de dejar de hacer uso del procesador

El proceso en ejecución pasa de *Run* a *Ready*. Un proceso que haya sido seleccionado por el planificador pasará de *Ready* a *Run*.

- El controlador de disco notifica que la última petición de lectura ya está disponible.



El proceso que esperaba esta operación pasará de *Block* a *Ready*.

- Un proceso de usuario intenta ejecutar una instrucción privilegiada de lenguaje máquina.

El proceso será abortado por el SO con lo que desaparecerá del sistema. Un proceso que haya sido seleccionado por el planificador pasará de *Ready* a *Run*.

- 1.3 ¿Cuál es la utilidad de que los procesadores dispongan de instrucciones privilegiadas del lenguaje máquina y de diversos modos de ejecución? ¿Quién puede utilizar las instrucciones privilegiadas?

La utilidad es evitar que los programas de usuario puedan utilizar las instrucciones privilegiadas directamente porque éso podría provocar problemas de estabilidad y de integridad en la máquina. Estas instrucciones sólo se pueden utilizar indirectamente, a través de las llamadas al sistema/interrupciones/excepciones, bajo el control del SO.

Estas instrucciones sólo son utilizables desde el núcleo del SO.

2. Módulo 3: [Del 4 al 10 de noviembre] (Peso 60 % = 10 % + 10 % + 10 % + 30 %)

Responded **justificadamente** a las siguientes preguntas relacionadas con el módulo 3 de la asignatura:

- 2.1 Un usuario ejecuta la orden `ps -u` en un sistema Linux y obtiene el siguiente resultado:

```
prompt$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
enricm    22425  0.0  0.1  25856  5476 pts/20    Ss   13:43   0:00 bash
enricm    22604  0.1  0.2  58908  8972 pts/9     S+   13:53   0:05 vim file.tex
enricm    23325  0.2  2.1 1375448 82500 pts/20    Sl+  14:28   0:02 evince a.pdf
prompt$
```

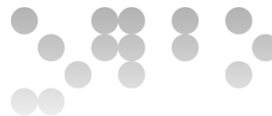
Consultado el manual del sistema, contestad con vuestras propias palabras:

- ¿Qué información aportan las columnas `%MEM`, `VSZ`, `RSS`?

`%MEM` indica el porcentaje de memoria física ocupada por el proceso.

`VSZ` indica el tamaño del espacio lógico del proceso (en Kilobytes).

`RSS` indica la cantidad de memoria física ocupada por el proceso (en Kilobytes).



- Aproximadamente, ¿cuánta memoria física tiene la máquina?

Mediante una regla de tres, podemos concluir que corresponde a $100 \times 82500 \text{ Kilobytes} / 2,1 \approx 3,75 \text{ Gigabytes}$

La información más precisa es la correspondiente al proceso **evince** porque los otros procesos ocupan poca memoria con lo que los redondeos al calcular los porcentajes son muy significativos.

Como constatación, la memoria física total disponible por Linux es de 3908072 Kilobytes que corresponde a 3,727 Gigabytes.

```
prompt$ more /proc/meminfo
MemTotal:      3908072 kB
MemFree:       450840 kB
...
```

- 2.2 ¿Hay algún límite para el tamaño máximo del espacio lógico de un proceso?

Sí, está limitado por el tamaño del bus de direcciones del procesador. En procesadores con buses de 32 bits, el espacio lógico máximo es de 2^{32} bytes, es decir, 4 Gigabytes, y en procesadores con buses de 64 bits el espacio lógico máximo sería de 2^{64} bytes, es decir, 16 Exabytes.

- 2.3 ¿En qué parte del espacio lógico (código, pila, zona de datos estáticos, zona de datos dinámicos-heap) de un proceso se almacenan los siguientes objetos?

2.3.1 Una variable local

Las variables locales se almacenan en la pila del proceso.

2.3.2 Una variable global

Las variables globales se almacenan en la zona de datos.

2.3.3 Un parámetro pasado a una rutina

Los parámetros de las rutinas acostumbran a almacenarse en la pila (aunque en algunos casos se almacenan en registros).

2.3.4 El código máquina de una rutina

El código de las rutinas del programa se almacena en la zona de código.

2.3.5 El resultado de solicitar memoria dinámica con **malloc**



La memoria asignada dinámicamente con `malloc` se almacena en la zona de datos (más concretamente en el heap).

2.4 Sea un sistema de gestión de memoria basado en paginación bajo demanda en el que las páginas tienen un tamaño de 4Kbytes, las direcciones lógicas son de 16 bits y el espacio físico es de 32 Kbytes. Sobre este sistema se crean dos procesos.

- Proceso 1: su fichero ejecutable determina que cada área (código, datos inicializados, datos no inicializados y la pila) ocupan dos páginas cada una.
- Proceso 2: su fichero ejecutable determina que el código y los datos inicializados ocuparán dos páginas cada una, que no existen datos no inicializados y que la pila ocupará dos páginas.

Se pide:

2.4.1 Estimad el tamaño del fichero ejecutable correspondiente al proceso 1.

En el fichero ejecutable es necesario guardar el contenido de la zona de código y de los datos inicializados. Como son necesarias dos páginas para cada una de estas zonas, esto implica un máximo de 16 Kbytes (la fragmentación interna en estas páginas puede causar que el tamaño real sea inferior a 16 Kbytes). Además, hay que sumar el tamaño de las cabeceras del fichero ejecutable.

2.4.2 Suponiendo que las páginas se cargan en memoria física tal y como indica el diagrama de la página 9 de este enunciado, indicad cuál será el contenido de las tablas de páginas de ambos procesos. Indicad por qué motivo ambas tablas de páginas tienen 16 entradas.

El contenido de las tablas de páginas es el que se muestra en la figura (el contenido de las posiciones en blanco es irrelevante al tratarse de páginas inválidas o de páginas no presentes en memoria física).



Proceso 1

Espacio lógico		Tabla de páginas		
pág. lóg.		valida	presente	frame
0x0	C1-1	1	0	
0x1	C1-2	1	0	
0x2	D1-1	1	1	0x2
0x3	D1-2	1	1	0x7
0x4	D1-3	1	1	0x0
0x5	D1-4	1	1	0x3
0x6	-	0		
0x7	-	0		
0x8	-	0		
0x9	-	0		
0xA	-	0		
0xB	-	0		
0xC	-	0		
0xD	-	0		
0xE	P1-2	1	1	0x1
0xF	P1-1	1	0	

Proceso 2

Espacio lógico		Tabla de páginas		
pág. lóg.		valida	presente	frame
0x0	C2-1	1	1	0x5
0x1	C2-2	1	0	
0x2	-	0		
0x3	D2-1	1	0	
0x4	D2-2	1	1	0x6
0x5	-	0		
0x6	-	0		
0x7	-	0		
0x8	-	0		
0x9	-	0		
0xA	-	0		
0xB	-	0		
0xC	-	0		
0xD	-	1	1	0x1
0xE	P2-2	1	0	
0xF	P2-1	1	0	

Espacio físico

frame	
0x0	D1-4
0x1	P2-2
0x2	D1-1
0x3	P1-2
0x4	
0x5	C2-1
0x6	D2-2
0x7	D1-3

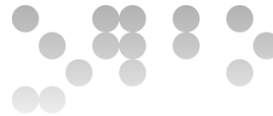
Área de swap

C1-1	D1-2
C1-2	
P1-1	
P2-1	C2-2
D2-1	

Las tablas de páginas son de 16 entradas porque es el resultado de dividir el tamaño del espacio lógico (2^{16} bytes) entre el tamaño de la página (4 Kbytes, es decir, 2^{12} bytes). El resultado es 2^4 , es decir, 16 páginas.

2.4.3 Suponiendo que el proceso en ejecución es el proceso 1, indicad cuáles serán las direcciones físicas correspondientes a las siguientes direcciones lógicas: 0x0123 y 0x2342. ¿Variaría la respuesta si el proceso en ejecución fuera el proceso 2? En caso afirmativo, indicad el motivo y cómo cambiaría.

La respuesta depende del proceso en ejecución porque cada proceso utiliza una tabla de páginas diferente. Como las páginas son de 2^{12} bytes, los 12 bits de menos peso (3 dígitos hexadecimales)



constituyen el desplazamiento dentro de la página. El resultado de las traducciones sería el siguiente:

Dirección lógica	Dirección física Proceso 1	Dirección física Proceso 2
0x0123	Excepción: fallo de página	0x5123
0x2324	0x2324	Excepción: dirección inválida

Se recuerda que realizar divisiones donde el denominador es una potencia de 2 es trivial si trabajamos en base 2 (o en base 16). Dividir entre 2^n no es más que descartar los n bits de menos peso del numerador (expresado en base 2); el residuo de la división entre 2^n son los n bits bajos del numerador.

3. Módulo 4 [Del 11 al 13 de noviembre] (Peso 10 % = 5 % + 5 %)
Responded **justificadamente** a las siguientes preguntas relacionadas con el módulo 4 de la asignatura:

3.1 Indicad si los siguientes dispositivos son físicos, lógicos o virtuales:

3.1.1 El canal de salida estándar

El canal de salida estándar es un dispositivo virtual. Es la abstracción que nos ofrece el SO para que un proceso pueda acceder a un dispositivo de entrada/salida.

3.1.2 El teclado

El teclado es un dispositivo físico.

3.1.3 Una pipe

La pipe es una dispositivo lógico. Es una abstracción ofrecida por SO para poder comunicar dos procesos.

3.1.4 El fichero `/bin/ls`

Los ficheros son dispositivos lógicos. Es una abstracción ofrecida por SO para trabajar más cómodamente sobre el dispositivo físico disco duro.

3.1.5 El fichero `/dev/tty`

`/dev/null` es un dispositivo lógico porque es un fichero ofrecido por el SO.

3.2 ¿En qué consiste la independencia de dispositivos? Poned un ejemplo de su utilidad.



Consiste en que, desde el punto de vista del usuario, todos los dispositivos se acceden de la misma forma. Los procesos trabajan sobre dispositivos virtuales y es el SO quien se encarga de hacer el acceso sobre el dispositivo físico. Desde el punto de vista del proceso, es lo mismo escribir sobre una pantalla que sobre un archivo. Esto simplifica la programación de las aplicaciones y facilita la reutilización del código.

Recursos

- Módulos 1, 2, 3 y 4 de la asignatura.
- El aula "Laboratorio de Sistemas Operativos" (podéis plantear vuestras dudas relativas al entorno UNIX, programación,...).

Criterios de evaluación

Se valorará la justificación de las respuestas presentadas.

El peso de cada pregunta está indicado en el enunciado.

Formato y fecha de entrega

Se entregará un fichero **zip** que contendrá un fichero **pdf** con la respuesta a las preguntas y, si es preciso, los ficheros adicionales que queráis entregar.

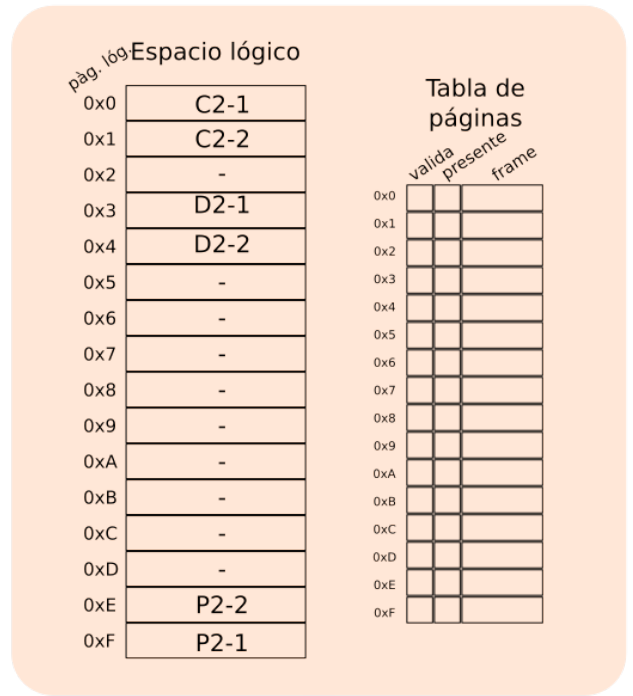
Fecha límite de entrega: 24:00 del 13 de noviembre de 2018.



Proceso 1



Proceso 2



Espacio físico

frame	
0x0	D1-4
0x1	P2-2
0x2	D1-1
0x3	P1-2
0x4	
0x5	C2-1
0x6	D2-2
0x7	D1-3

