



APELLIDOS		NOMBRE		Grupo
DNI		Firma		

- No desgrape las hojas.
- Conteste exclusivamente en el espacio reservado para ello.
- Utilice letra clara y legible. Responda de forma breve y precisa.
- El examen consta de 8 cuestiones, cuya valoración se indica en cada una de ellas.

1. Un sistema de archivos organizado en bloques de 512 Bytes, con punteros a bloque de 2Bytes tiene un tiempo medio de acceso a bloque de 2 mseg. Suponga que en memoria se encuentra la información que contiene el puntero al primer bloque de datos de un archivo de 32KBytes y que el tiempo de acceso a Memoria principal es despreciable. Indique de forma justificada el tiempo medio necesario para acceder a la lectura del Byte 4650 de dicho archivo para cada uno de los siguientes métodos de asignación de bloque:

(1,0 punto = 0,4 + 0,3 + 0,3)

1	a) Asignación Enlazada
	b) Asignación indexada con dos niveles de indexación
	c) FAT (<i>File allocation Table</i>). Considere que la tabla con todos sus punteros a bloque se encuentra en memoria principal y que el tiempo de acceso a memoria es despreciable

2. Un proceso debe escribir el mensaje “*parent message*” en el archivo “*messages.txt*” y a continuación crear un hijo que escriba el mensaje “*child message*” en el mismo archivo. El proceso padre debe esperar la finalización del proceso hijo para leer TODO el contenido del archivo “*messages.txt*” a través de su entrada estándar y escribirlo en su salida estándar. Además el proceso padre debe finalizar cerrando todos los descriptores de archivos abiertos. Complete el programa en código C del apartado a) con las primitivas POSIX necesarias, una en cada línea con número subrayado, para que realice dichas acciones. **NOTA:** Utilice `open()`, `read()`, `write()`, `close()` y `dup2()` cuando lo necesite. No utilice la llamada al sistema `lseek()`.

(1,2 puntos = 0,8 + 0,4)

2	<pre> 1 #include <all_needed> 2 #define SIZE 50 3 int main(int argc, char **argv){ 4 int fd1, fd2, nbytes; 5 char buffer[SIZE]; 6 mode_t fd_mode = S_IRWXU; // file permissions 7 char *parent_message = "parent message \n"; 8 char *child_message = "child messages \n"; 9 fd1 = open(, O_TRUNC O_CREAT O_RDWR, fd_mode); //complete open() 10 write(, parent_message, strlen(parent_message)); //complete write() 11 if (fork() == 0) { // child 12 13 close(fd1); 14 exit(0); } 15 wait(NULL); 16 17 18 while(1) { 19 nbytes = read(, buffer, strlen(child_message)); //complete read() 20 if (nbytes > 0) 21 22 else 23 break; 24 } 25 close(fd1); 26 27 }</pre>																								
b)	<p>Rellene la tabla de descriptores de archivos del proceso hijo tras ejecutar la línea 12 y del proceso padre tras la línea 20. Las tablas deben ser correctas con los requisitos y la implementación de la sección a)</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <caption>Tabla del Proceso hijo en 12</caption> <tr><td>0</td><td></td></tr> <tr><td>1</td><td></td></tr> <tr><td>2</td><td></td></tr> <tr><td>3</td><td></td></tr> <tr><td>4</td><td></td></tr> <tr><td>5</td><td></td></tr> </table> <table border="1" style="border-collapse: collapse; text-align: center;"> <caption>Tabla del Proceso padre en 20</caption> <tr><td>0</td><td></td></tr> <tr><td>1</td><td></td></tr> <tr><td>2</td><td></td></tr> <tr><td>3</td><td></td></tr> <tr><td>4</td><td></td></tr> <tr><td>5</td><td></td></tr> </table> </div>	0		1		2		3		4		5		0		1		2		3		4		5	
0																									
1																									
2																									
3																									
4																									
5																									
0																									
1																									
2																									
3																									
4																									
5																									

3. El programa `/usr/bin/passwd` permite cambiar las contraseñas de los usuarios. Para su ejecución este programa necesita leer y escribir los archivos `/etc/passwd` y `/etc/shadow`. El programa `/usr/bin/chage` permite cambiar el tiempo de expiración de una contraseña (guardado en `/etc/shadow`), lo que implica leer el archivo `/etc/passwd` y leer y escribir en el archivo `/etc/shadow`. Considere el siguiente contenido (parcial) de estos dos directorios:

(1,2 puntos = 0,8 + 0,4)

directorio `/usr/bin`:

```
i-nodo permisos enlaces usuario grupo tamaño fecha nombre
655364 drwxr-xr-x 2 root root 36864 nov 18 14:02 .
655363 drwxr-xr-x 12 root root 4096 jul 20 2018 ..
656249 -rwxr-sr-x 1 root shadow 71816 mar 22 2019 chage
658048 lrwxrwxrwx 1 root root 5 may 20 2019 gcc -> gcc-7
657101 lrwxrwxrwx 1 root root 22 may 8 2019 gcc-7 -> x86_64-linux-gnu-gcc-7
657839 -rwsr-xr-x 1 root root 59640 mar 22 2019 passwd
655397 -rwxr-xr-x 2 root root 2097720 nov 19 2018 perl
655397 -rwxr-xr-x 2 root root 2097720 nov 19 2018 perl5.26.1
657867 -rwxr-xr-x 1 root root 1010624 may 8 2019 x86_64-linux-gnu-gcc-7
```

directorio `/etc`:

```
808275 -rw-r--r-- 1 root root 1812 jul 16 2018 passwd
800878 -rw-r----- 1 root shadow 1041 jul 20 2018 shadow
```

- 3 a)** Indique si funcionaría sin error la ejecución por el usuario especificado de las siguientes órdenes. En caso de éxito justifique cuáles son los permisos que se van comprobando y, en caso de error, cuál es el permiso que falla y por qué.

(UID, GID)	ORDEN	ÉXITO	JUSTIFICACIÓN
(eva, fso)	<code>/usr/bin/passwd</code>		
(eva, fso)	<code>chage -M7 eva</code> (cambio de la expiración a 7 días para el usuario eva)		
(eva, fso)	<code>gcc --help</code> (obtener ayuda del programa gcc)		

- b)** Para el directorio `/usr/bin` justifique número de enlaces de sus archivos “.” y “perl”

Nombre	Enlaces	JUSTIFICACIÓN
.	2	
perl	2	

4. Un disco con capacidad de 64 MBytes, se formatea en una versión de MINIX, cuyas especificaciones son:

- El bloque de arranque y el superbloque ocupan 1 bloque cada uno
- Tamaño del nodo-i de 32 Bytes (7 punteros directos, 1 indirecto, 1 doble indirecto).
- Punteros a zona de 16 bits
- Entrada de directorio de 16 Bytes: 2 Bytes para nodo-i, 14 Bytes para el nombre
- 1 zona = 1 bloque = 1 KByte

(1,6 puntos = 0,8 + 0,8)

4	<p>a) Se formatea reservando espacio en la cabecera para un total de 16384 nodos-i (16 K nodos-i). Calcule el número de bloques que ocupa cada elemento de la cabecera y el área de datos. Justifique su respuesta</p> <table border="1"> <tr> <td>Arranque</td> <td>Super bloque</td> <td>Mapa de bits de Nodos-i</td> <td>Mapa de bits de Zonas</td> <td>Nodos- i</td> <td>Zonas de datos</td> </tr> </table>					Arranque	Super bloque	Mapa de bits de Nodos-i	Mapa de bits de Zonas	Nodos- i	Zonas de datos
	Arranque	Super bloque	Mapa de bits de Nodos-i	Mapa de bits de Zonas	Nodos- i	Zonas de datos					
<p>b) Suponga que partiendo del disco vacío, se va ocupando el disco de acuerdo a las siguientes acciones:</p> <ul style="list-style-type: none"> - Creación del directorio / (<i>raíz</i>) - Creación del archivo regular <i>/fso</i> con un tamaño de 514KBytes - Creación del directorio <i>/Examen</i> - Creación del directorio <i>/Examen/Final</i> - Creación del archivo regular <i>/Examen/Final/Cursoactual</i> con un tamaño de 800 KBytes - Creación del enlace físico <i>/EFinal</i> al archivo regular <i>/Examen/Final/Cursoactual</i> <p>Tras estas acciones, indique de forma razonada para dicho sistema de archivos los siguientes valores:</p> <ul style="list-style-type: none"> - Tamaño en Bytes del directorio / (<i>raíz</i>) - Número de i-nodos ocupados en dicho sistema de archivos - Número de bloques de la zona de datos ocupados con referencias a otros bloques 											

5. Un sistema gestiona una memoria principal de 1000KB con asignación contigua con particiones variables y SIN compactación y cuya ocupación inicial a considerar es:

0		1000KB-1			
SO 100KB	HUECO 150KB	P1 200KB	HUECO 100KB	P2 100KB	HUECO 350KB

El algoritmo de asignación, ubica siempre los procesos dentro de un hueco ajustándose a las direcciones más bajas (izquierda), dejando libre las direcciones más altas del hueco. Indique las bases de los procesos P3 (100KBytes), P4 (400KBytes) y P5 (200KBytes) para cada uno de los algoritmos de asignación de huecos: Mejor Ajuste (Best Fit), Primer Ajuste (First Fit) y Peor Ajuste (Worst Fit). Tenga en cuenta la secuencia de eventos propuesta y el estado inicial de la memoria para cada caso. Si un proceso no cabe indique NO CABE y continúe con el siguiente.

(1,2 puntos = 0,9 + 0,3)

EVENTO	BEST FIT	FIRST FIT	WORST FIT
LLEGA P3 ¿BASE P3?			
TERMINA P1	----	----	----
LLEGA P4 ¿BASE P4?			
TERMINA P2	----	----	----
LLEGA P5 BASE P5?			

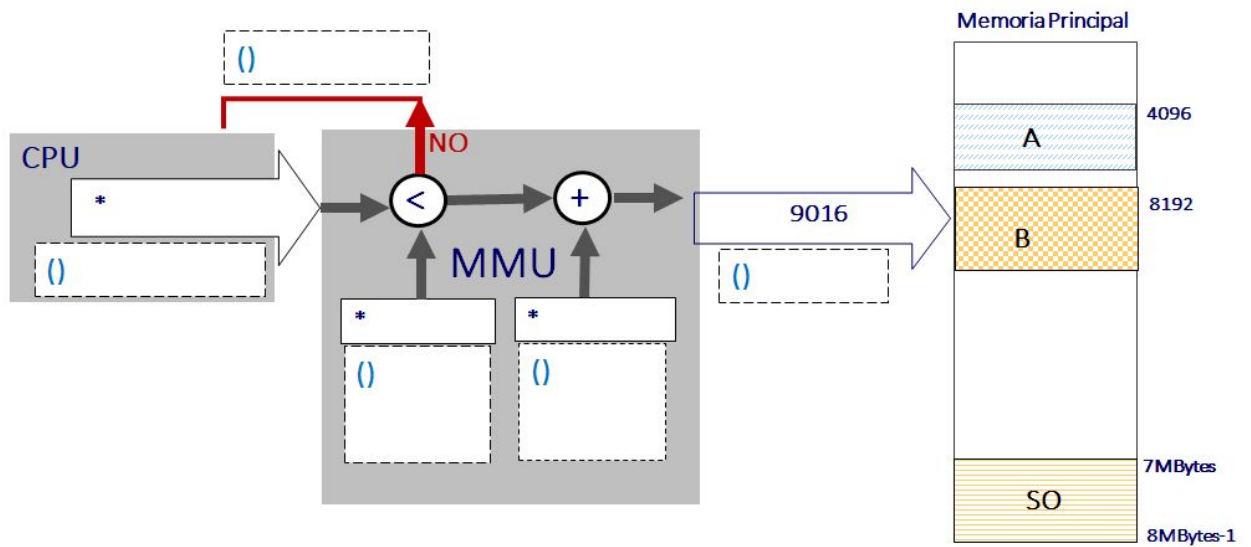
b) Indique para el algoritmo de Worst Fit, los huecos que han quedado (**dirección de inicio y tamaño**) al finalizar la secuencia de eventos anterior, y el tipo de fragmentación que se produce:

--

6. Considere un sistema con una memoria principal de 8MBytes en la que se encuentran ubicados dos procesos A (de tamaño 2KBytes) y B (de tamaño 4KBytes) a partir de las direcciones que se indican en la figura.

(0,8 puntos)

- 6 Suponga que en este instante se está ejecutando el proceso B y se accede a la dirección de Memoria principal 9016. Dado que el sistema dispone de una MMU básica y asignación contigua de memoria, rellene los valores en los recuadros marcados con * para cada uno de los elemento, de manera que se pueda llevar a cabo dicho acceso. Indique también el nombre de cada elemento en los recuadros próximos a ellos marcados con ()



7. Una familia de procesadores de Intel con 32 bits de dirección, trabaja con una arquitectura de memoria paginada con dos niveles de paginación y páginas de 4KBytes. Para ambos niveles de paginación, cada entrada de la tabla de páginas (descriptor) ocupa 4 Bytes y cada tabla de páginas ocupa 4KBytes. Para este sistema indique de forma justificada:

(1.2 puntos = 0,4 x 3)

7	a) Número de páginas que puede llegar a tener, en total, un proceso
	b) Tamaño en Bytes de la tabla de páginas de primer nivel y número máximo de entradas (descriptores) que puede llegar a tener dicha tabla
	c) Para un programa que ocupa 100MBytes, indique de forma razonada el número de descriptores de cada nivel y el espacio que consumen sus tablas de páginas

8. Sea un sistema con paginación por demanda, páginas de 4KBytes y direcciones lógicas y físicas de 24 bits. En dicho sistema se están ejecutando dos procesos A y B, a los que el sistema asigna 5 marcos que comparten con una política de reemplazamiento **de segunda oportunidad con ámbito global**. Ambos procesos tienen un tamaño de 4 páginas (de la 0 a la 3). La relación de los marcos asignados con las páginas de los procesos se muestra en la siguiente tabla:

Marco	Proceso: página	Tiempo de Carga	Tiempo. último acceso	Bit de Referencia	Bit de Validez
0	A:0	1	6	1	1
1	B:1	2	15	1	1
2	A:1	7	7	1	1
3	B:2	8	12	1	1
4	A:3	12	14	1	1

(1,8 puntos = 0,4 + 0,4 + 1,0)

- 8 a) Sabiendo que hasta el instante $t=15$ sólo se han producido fallos de página sin reemplazo, indique el contenido de las entradas de la tabla de páginas del proceso B, para cada uno de sus descriptores, no es necesario poner los bits de permisos, ni los tiempos.

- b) A partir del instante $t=16$ la CPU emite las siguiente direcciones lógicas:

A:0x002345, A:0x002346, B:0x001B72, B:0x000B32, B:0x000B33,

A:0x000111, A:0x001222, A:0x000111, B:0x002ABC, B:0x002ABD

Suponga que se maneja un esquema de área activa, con un tamaño de ventana de 4. Determine las áreas activas de los procesos A y B tras completarse el último acceso.

¿Puede producirse hiperpaginación?. Justifique la respuesta

c) Calcule la serie de referencias correspondientes a la secuencia de direcciones del apartado b) :

Aplique el algoritmo de segunda oportunidad con ámbito global para dicha serie de referencias. Rellene la siguiente tabla (las columnas que sean necesarias, máximo de 8), con la evolución de los marcos de memoria de los procesos A y B a partir del instante $t=16$. La primera columna corresponde al instante $t=15$, en el que el proceso B accedió a su página 1. Indique en cada casilla de la tabla el proceso, la página y el valor de bit R tras realizar el acceso y cuando haya reemplazo marque la casilla elegida como víctima.

marco	B:1 (R)								
0	A:0 (1)								
1	B:1 (1)								
2	A:1 (1)								
3	B:2 (1)								
4	A:3 (1)								

Indique el número de reemplazos: