

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 7 cuestiones, cuya valoración se indica en cada una de ellas.

1. Las bibliotecas o librerías son archivos que contienen código de funciones y puede ser enlazadas de forma estática o dinámica. Indique, poniendo una X, cuáles de las siguientes características corresponden a un tipo u otro de enlace o a ambos: (Nota: Un error penaliza una respuesta correcta)

(1,0 punto)

1	Características	Estático	Dinámico
	Genera archivos ejecutables de menor volumen (tamaño)		X
	Genera archivos ejecutables que contiene el código propio del proceso junto con el de las librerías	X	
	En el mapa de memoria del proceso aparecen regiones propias cuyo soporte es el archivo de código de cada librería enlazada		X
	Varios procesos pueden compartir el código de la biblioteca una vez ubicado este código en memoria principal		X
	Siempre que el SO actualiza la librería, es necesario volver a enlazar la librería	X	
	El enlace se resuelve en tiempo de ejecución, pudiendo ocasionar retrasos en la ejecución		X
	Puede generar múltiples copias en memoria principal de una misma librería	X	
	En el mapa de memoria el código de las librerías tiene como soporte el archivo de código del propio proceso	X	

2. Indique cómo resuelve la MMU (Memory Management Unit) la reubicación de procesos en tiempo de ejecución en la asignación contigua y qué información necesita para llevarla a cabo: (0,5 puntos)

2	<p>La MMU traduce direcciones lógicas (DL) a física (DF), para ello necesita conocer las direcciones de memoria principal donde se ha ubicado el proceso. Cada vez un proceso en memoria cambia de ubicación en memoria principal (se reubica) se debe actualizar la información de dicha ubicación en la MMU para que realice bien la traducción.</p> <p>En asignación contigua la MMU utiliza los registros base y límite, con la DF partir de la cual se ha ubicado el proceso y el tamaño del proceso. El registro base debe actualizarse con cada reubicación del proceso.</p>
---	---

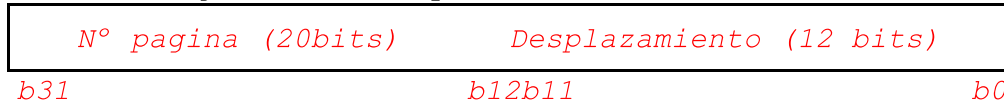
3. Un sistema con paginación pura (sin memoria virtual), direcciones lógicas de 32 bits, dotado de una memoria principal de 64GBytes, utiliza 20 bits para el número de página y 6 Bytes para cada descriptor de página. Responda de forma justificada a los siguientes apartados:

( 1.5 puntos =0.5+0.25+0.25+0.5)

- 3 a) Formato de la dirección lógica y física con nombre y número de bits de cada campo o elemento

DL de 32 bits  $\rightarrow 32 - 20 = 12$  bits para desplazamiento  
 64GBytes =  $2^6 \cdot 2^{30}$  Bytes =  $2^{36}$  Bytes  $\rightarrow 36$  bits para DF  
 DF 36 bits  $\rightarrow 36 - 12 = 24$  bits para número de marco

*Dirección lógica (nombre y número de bits de cada elemento)*



*Dirección física (nombre y número de bits de cada elementos)*



- b) Tamaño de tabla de páginas de un proceso

Si las direcciones lógicas son de 32 bits, con 20 bits para el número de página  $\rightarrow 2^{20}$  es el número máximo de páginas de un proceso. La tabla de páginas necesita un descriptor por cada página  $\rightarrow 2^{20}$  páginas \* 6 Bytes  $\rightarrow 6$  MBytes es el tamaño de la tabla de página

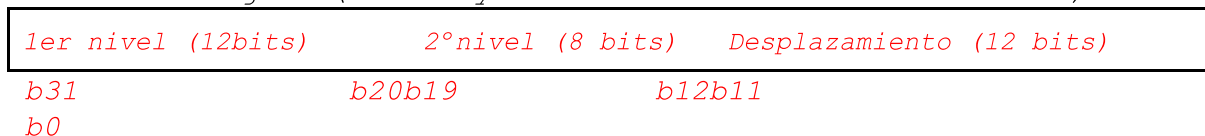
- c) Grado máximo de multiprogramación del sistema considerando procesos de 4GBytes

El grado de multiprogramación máximo viene dado por el número de proceso que pueden estar en memoria simultáneamente. Al ser paginación pura los procesos deben estar ubicados completos en memoria para poder ejecutarse. El tamaño máximo de un proceso es el que viene dado por el tamaño del espacio lógico  $2^{32}$ , mientras que el número de proceso que pueden ser ubicado en memoria vendrá dado por la división entre el espacio físico y el tamaño máximo de los procesos en este sistema  
 $64\text{GBytes} / 2^{32} = 2^{36} / 2^{32} = 2^4 = 16$  procesos

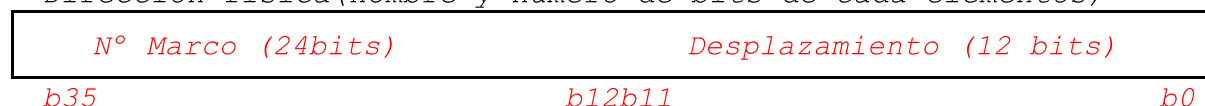
- d) Para una paginación a dos niveles con 256 descriptors de segundo nivel, indique el formato de la dirección lógica y física

256 descriptors =  $2^8$  luego 8 bits, como teníamos 20 bits para el número de página  $\rightarrow 20 - 8 = 12$  para el primer nivel, la dirección física no cambia

*Dirección lógica (nombre y número de bits de cada elemento)*



*Dirección física (nombre y número de bits de cada elementos)*



4. En un sistema con paginación por demanda (memoria virtual) y reemplazo local, el tamaño lógico máximo de un proceso es de 4K páginas, mientras que el tamaño de página es de 64KBytes. La siguiente tabla contiene toda la información relativa a los procesos P y Q en el instante (t)  $t=170$ :

Información de los proceso P y Q en t=170							
Proceso	Marco	Página	Instante de carga	t de última referencia	Bit R (referencia)	Bit M (modificado)	Bits RWX
P	0x4A	0xC72	160	161	1	0	101
P	0x4B	0xC71	120	140	1	1	101
P	0x4C	0xA70	36	152	0	1	110
P	0x4D	0xA73	30	163	1	1	110
Q	0x4E	0xA70	40	167	1	0	101
Q	0x4F	0xA73	42	142	0	1	110

---

Partiendo siempre de la información de la tabla, conteste cada apartado: **( 2 puntos=0.25+0.5+0.5+0.75)**

- | 4  | <p>a) Indique qué páginas de P y Q tienen el bit de validez a 0 y escriba el contenido de los descriptores de páginas con bit de validez (v) a 1.</p> <p>En la tabla de páginas todas las páginas tendrán el bit de validez a 0 excepto las cargadas en memoria principal. El descriptor de cada página tendrá el número de marco donde está ubicada y los bits (V,RWX, etc)</p> <table border="1"> <thead> <tr> <th>Proceso P</th><th>Página descriptor</th><th>Proceso Q</th><th>Página descriptor</th></tr> </thead> <tbody> <tr> <td>...</td><td></td><td>...</td><td></td></tr> <tr> <td>0xA70</td><td>0x4C v=1</td><td>0xA70</td><td>0x4E v=1</td></tr> <tr> <td>0xA73</td><td>0x4D v=1</td><td>0xA73</td><td>0x4F v=1</td></tr> <tr> <td>0xC71</td><td>0x4B v=1</td><td></td><td></td></tr> <tr> <td>0xC72</td><td>0x4A v=1</td><td></td><td></td></tr> </tbody> </table> <p>b) Calcule la dirección lógica que corresponde a la dirección física 0x4D4AB1</p> <p>La DF-&gt; 0x4D4AB1 corresponde al proceso P ya que es la página 0xA73 de este proceso la que está ubicada en el marco 4D y por tanto no tiene correspondencia con una DL del proceso Q</p> <table border="1"> <thead> <tr> <th>Dirección Física --&gt; Dirección de lógica P</th><th>Dirección Física --&gt; Dirección lógica Q</th></tr> </thead> <tbody> <tr> <td>Dirección de lógica P --&gt; Dirección Física</td><td>Dirección lógica Q --&gt; Dirección Física</td></tr> <tr> <td>0x4D4AB1 ----&gt; 0xA734AB1</td><td>0x4D4AB1 ----&gt; NO CORRESPONDE</td></tr> </tbody> </table> | Proceso P                                 | Página descriptor                      | Proceso Q              | Página descriptor      | ...        |            | ... |      | 0xA70      | 0x4C v=1   | 0xA70      | 0x4E v=1   | 0xA73      | 0x4D v=1 | 0xA73 | 0x4F v=1   | 0xC71      | 0x4B v=1   |            |            | 0xC72 | 0x4A v=1 |            |            | Dirección Física --> Dirección de lógica P | Dirección Física --> Dirección lógica Q | Dirección de lógica P --> Dirección Física | Dirección lógica Q --> Dirección Física | 0x4D4AB1 ----> 0xA734AB1 | 0x4D4AB1 ----> NO CORRESPONDE |            |            |            |            |  |
|--|--|---|--|------------------------|------------------------|------------|------------|-----|------|------------|------------|------------|------------|------------|----------|-------|------------|------------|------------|------------|------------|-------|----------|------------|------------|--|---|--|---|--------------------------|-------------------------------|------------|------------|------------|------------|--|
| Proceso P                                  | Página descriptor  | Proceso Q                                 | Página descriptor                      |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| ...  |  | ...                                       |  |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0xA70                                      | 0x4C v=1   | 0xA70                                     | 0x4E v=1                               |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0xA73                                      | 0x4D v=1   | 0xA73                                     | 0x4F v=1                               |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0xC71                                      | 0x4B v=1   |   |  |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0xC72                                      | 0x4A v=1   |   |  |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| Dirección Física --> Dirección de lógica P | Dirección Física --> Dirección lógica Q  |   |  |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| Dirección de lógica P --> Dirección Física | Dirección lógica Q --> Dirección Física  |   |  |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0x4D4AB1 ----> 0xA734AB1                   | 0x4D4AB1 ----> NO CORRESPONDE  |   |  |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
|  | <p>c) En el instante t=171 la CPU emite la dirección lógica 0xB95603A del proceso P y en el instante t=172 la CPU emite la dirección lógica 0xB95603A del proceso Q. Calcule las direcciones físicas correspondientes si se utiliza un reemplazo con <b>segunda oportunidad</b>:</p> <p>Espacio lógico 4K páginas → 12 bits para núm. de página→ 3 dig. hex.<br/> Páginas de 64KBytes → 16 bits desplazamiento → 4 dígitos hexadecimal<br/> 2ª oportunidad → víctima es aquella página con bit de referencia a 0</p> <p>Para P-&gt; pag. 0xA70 que está en marco 4C<br/> Para Q-&gt; pag. 0xA70 que está en marco 4F</p> <table border="1"> <thead> <tr> <th>Dirección de lógica P -&gt; Dirección Física</th><th>Dirección lógica Q -&gt; Dirección Física</th></tr> </thead> <tbody> <tr> <td>0xB95603A ----&gt; 4C603A</td><td>0xB95603A ----&gt; 4F603A</td></tr> </tbody> </table>  | Dirección de lógica P -> Dirección Física | Dirección lógica Q -> Dirección Física | 0xB95603A ----> 4C603A | 0xB95603A ----> 4F603A |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| Dirección de lógica P -> Dirección Física  | Dirección lógica Q -> Dirección Física   |   |  |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0xB95603A ----> 4C603A                     | 0xB95603A ----> 4F603A   |   |  |                        |                        |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
|  | <p>d) Evolución del contenido de los marcos implicados si se utiliza <b>reemplazo LRU (least recently used)</b> y a partir de t=171 la CPU emite direcciones de P cuya serie de referencias es: 0xA71, 0xB40, 0xC72, 0xB51</p> <p>Ponemos entre () los t. de última referencia</p> <table border="1"> <thead> <tr> <th>Marco</th><th>t=170 (Inicio)</th><th>t=171 -&gt;A71</th><th>t=172-&gt;B40</th><th>t=173-&gt;C72</th><th>t=174-&gt;B51</th><th></th></tr> </thead> <tbody> <tr> <td>0x4A</td><td>0xC72(161)</td><td>0xC72(161)</td><td>0xC72(161)</td><td>0xC72(173)</td><td>0xC72(173)</td><td></td></tr> <tr> <td>0x4B</td><td>0xC71(140)</td><td>0xA71(171)</td><td>0xA71(171)</td><td>0xA71(171)</td><td>0xA71(171)</td><td></td></tr> <tr> <td>0x4C</td><td>0xA70(152)</td><td>0xA70(152)</td><td>0xB40(172)</td><td>0xB40(172)</td><td>0xB40(172)</td><td></td></tr> <tr> <td>0x4D</td><td>0xA73(163)</td><td>0xA73(163)</td><td>0xA73(163)</td><td>0xA73(163)</td><td>0xB51(174)</td><td></td></tr> </tbody> </table> <div> <div>FPR</div> <div>FPR</div> <div>FPR</div> </div>   | Marco                                     | t=170 (Inicio)                         | t=171 ->A71            | t=172->B40             | t=173->C72 | t=174->B51 |     | 0x4A | 0xC72(161) | 0xC72(161) | 0xC72(161) | 0xC72(173) | 0xC72(173) |          | 0x4B  | 0xC71(140) | 0xA71(171) | 0xA71(171) | 0xA71(171) | 0xA71(171) |       | 0x4C     | 0xA70(152) | 0xA70(152) | 0xB40(172)                                 | 0xB40(172)                              | 0xB40(172)                                 |   | 0x4D                     | 0xA73(163)                    | 0xA73(163) | 0xA73(163) | 0xA73(163) | 0xB51(174) |  |
| Marco                                      | t=170 (Inicio)   | t=171 ->A71                               | t=172->B40                             | t=173->C72             | t=174->B51             |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0x4A                                       | 0xC72(161)   | 0xC72(161)                                | 0xC72(161)                             | 0xC72(173)             | 0xC72(173)             |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0x4B                                       | 0xC71(140)   | 0xA71(171)                                | 0xA71(171)                             | 0xA71(171)             | 0xA71(171)             |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0x4C                                       | 0xA70(152)   | 0xA70(152)                                | 0xB40(172)                             | 0xB40(172)             | 0xB40(172)             |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |
| 0x4D                                       | 0xA73(163)   | 0xA73(163)                                | 0xA73(163)                             | 0xA73(163)             | 0xB51(174)             |            |            |     |      |            |            |            |            |            |          |       |            |            |            |            |            |       |          |            |            |  |   |  |   |                          |                               |            |            |            |            |  |

5. Al ejecutar el siguiente fragmento de código en C se crean tres procesos:

```
{int fd_pipe[2], fd; /*descriptor tubo, archivo regular*/
int pid;
/** Tabla Inicial *****/
close(1);
fd=open("result",O_WRONLY |O_CREAT|O_TRUNC,0666);
dup2(2,3);
fd=open("datos",O_RDONLY);
close(0);close(2);
pipe(fd_pipe);
pid=fork();
if (pid==0)
{ close(fd);
  dup2(3,2);
  /**Tabla de P2 *****/
  execlp("/usr/bin/wc", "wc", "-l",NULL);
}
pid=fork();
if (pid==0)
{ dup2(2,1); dup2(fd,0); dup2(3,2);
  /**Tabla de P3 *****/
  execlp("/bin/cat", "cat", NULL);
}
close(0); close(2);
/**Tabla de P1 *****/
while(pid != wait(&status));
}
```

	Tabla Inicial
0	STDIN
1	STDOUT
2	STDERR
3	
4	

(1.6 puntos=0.4+1.2)

5

a) Indique el parentesco entre los procesos P1, P2 y P3, y el esquema de comunicación que establecen

Se crean tres proceso un padre (P1) y dos procesos hijos (P2 y P3). Los hijos heredan el tubo del padre y establecen una comunicación entre ellos de manera que: el proceso P3 (lee del archivo "Datos" y escribe en el Tubo) y el proceso P2 Lee del tubo y escribe en el archivo "result"

b) Indique el contenido de la tabla de descriptores para cada uno de los procesos involucrados en los puntos marcados en el código con /\*\*Tabla ...\*\*/

	Tabla de P1
0	
1	"result"
2	
3	STDERR
4	"datos"
5	

	Tabla de P2
0	"fd_pipe[0]"
1	"result"
2	STDERR
3	STDERR
4	
5	

	Tabla de P3
0	"datos"
1	"fd_pipe[1]"
2	STDERR
3	STDERR
4	"datos"
5	

**6. Dado el siguiente listado de un directorio en un sistema POSIX:**

<i>permisos</i>	<i>enlaces</i>	<i>usuario</i>	<i>grupo</i>	<i>tamaño</i>	<i>fecha</i>	<i>nombre</i>
drwxr-xr-x	2	sterr	fso	4096	dec 9 2016	.
drwxrwxr-x	8	sterr	fso	4096	sep 10 2016	..
-r-xr-sr-x	1	sterr	fso	1139706	dec 9 2016	cp1
-r-sr-xr-x	1	sterr	fso	1139706	dec 9 2016	cp2
-r-xr-xr-x	1	sterr	fso	1139706	dec 9 2016	cp3
-r--r-----	1	sterr	fso	9706	dec 9 2016	f1
-r--rw-rw-	1	sterr	fso	4157	dec 9 2016	f2
-rw-r--r--	1	sterr	fso	222	dec 9 2016	f4

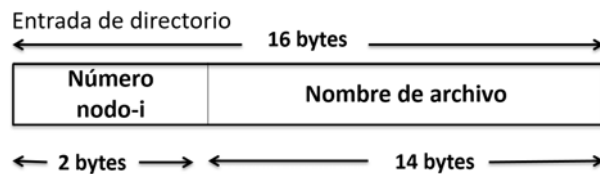
**(1.4 puntos)****6**

Teniendo en cuenta que los programas cp1, cp2 y cp3 son tres copias idénticas del programa del sistema cp que copia el contenido del archivo que recibe como primer argumento en otro cuyo nombre se indica como segundo argumento. Es decir, “cp1 a b” copia el contenido del archivo a al archivo b, si b no existe lo debe crear y después realizar la copia. Rellene la tabla indicando, si la orden funciona, el EUID y EGID (UID y GID efectivos del proceso al ejecutar la orden) y en caso de error, cuál es el permiso que falla.

(UID,GID)	ORDEN	¿FUNCIONA?	(EUID,EGID)	Si falla: permiso que hace que falle
(pepe, fso)	cp2 f1 f2	No	(sterr,fso)	Escritura en f2
(sterr, etc)	cp1 f1 f4	Si	(sterr,fso)	
(ana, etc)	cp1 f1 f2	Si	(ana,fso)	-
(ana, etc)	cp1 f1 f5	No	(ana,fso)	Escritura en .
(ana, etc)	cp2 f1 f5	Si	(sterr,etc)	-
(ana, etc)	cp3 f1 f2	No	(ana, etc)	Lectura f1
(ana, etc)	cp3 f4 f2	Si	(ana, etc)	-

7. Las siguientes figuras hacen referencia a los tamaños y estructuras de los elementos utilizados para formatear una partición MINIX, cuyos bloques son de 1KByte y con 1 zona= 1 bloque. Obsérvese que todos los campos del nodo-i son de 16 bits salvo "Nº de enlaces" y GID que son de 8 bits.

La entrada de directorio tiene el siguiente formato:



(2,0 puntos)



- a) Calcule el tamaño de partición teniendo en cuenta que ha sido formateada para obtener el número máximo de zonas y disponer del número máximo de nodos-i.

Puesto que los punteros de zonas son de 16 bits, podemos tener hasta  $2^{16}$  zonas=64K zonas, como cada zona es de 1KB, el tamaño máximo de la partición es 64Kx1KB=64MB

- b) Número máximo de nodos-i

Puesto que los números de nodos-i son 16 bits, podemos tener hasta  $2^{16}$  nodos-i=64K archivos

Considerando un sistema de archivos Minix con los parámetros descritos anteriormente que tiene un tamaño de partición 32MBytes (Megabytes) y 32 K nodos-i, responda:

- c) Número de bloques que ocupa el Mapa de bits nodos-i, el Mapa de bits Zonas y los Nodos-i

32MB/1KB=32K zonas, como tenemos el mismo número de nodos-i que de zonas (32K) el tamaño de ambos mapas de bits será el mismo 32K/8Kbits=4 bloques, en número de bloques para nodos-i será 32Kx32B/1KB=1024 bloques

- d) Espacio libre en disco después de formatear (incluye la creación del directorio raíz)

El número de zonas es 32K-1-1-4-4-1024=31734 zonas. El directorio raíz vacío ocupa una zona, por tanto el espacio libre es de 31733 KB

- e) Número máximo de enlaces físicos que puede tener un archivo

El campo número de enlaces es de 8 bits, por tanto el número más alto representable es  $2^8-1=255$  enlaces

- f) Número máximo de enlaces simbólicos que puede tener un archivo

Cada enlace simbólico es un archivo, por tanto el máximo número de nodos-i que es 32K (en realidad será del orden de 31K porque cada enlace simbólico ocupa una zona y no tenemos tantas zonas libres)

- g) Número máximo de directorios que puede contener un directorio

El número de enlaces de un directorio es el número de directorios que contiene (pues sus entradas .. le apuntan) más 2 (el . y su nombre en su padre), por tanto el máximo de directorios será  $255-2=253$  directorios.

- h) Número de zonas que ocupan 10 archivos de 100 Bytes

Un archivo de 100 Bytes necesita una zonas de datos, por lo que 10 archivos ocuparán 10 zonas

- i) Número de zonas que ocupan 10 archivos de 100 KBytes

Un archivo de 100 KB necesita 100 zonas de datos, para numerarlas necesitamos un puntero indirecto, por tanto ocupará 101 zonas. 10 archivos ocuparán 1010 zonas.