



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 ejercicio consta de 8 cuestiones, la cuestión 1 vale 1.5 puntos y la 3 vale 2.5 (0.75+ 1+ 0.5+0.25) puntos, el resto de cuestiones valen 1 punto cada una.
- Tiempo de realización 2 horas

Sea un sistema de memoria con particiones de tamaño variable y el siguiente estado de ocupación:

Libre 20K	Ocupado	Libre 10K	Ocupado	Libre 5K	Ocupado	Libre 15K	Ocupado	Libre 25K
--------------	---------	--------------	---------	-------------	---------	--------------	---------	--------------

donde la lista de huecos libres se mantiene ordenada hacia direcciones crecientes:

lista de huecos  $\rightarrow 20K \rightarrow 10K \rightarrow 5K \rightarrow 15K \rightarrow 25K$

A continuación solicitan ser ubicados en memoria cuatro procesos en el siguiente orden P1, P2, P3 y P4. Teniendo en cuenta que sus tamaños son P1(10K), P2(15K), P3(7K) y P4(14K), indique el estado de la memoria (huecos libres y ocupados) así como la lista de huecos si se aplican las técnicas:

- Best fit (mejor hueco)
- Worst fit (peor hueco)
- First fit (primer hueco)

1.5 puntos

1	a) Best fit y lista de huecos
	b) Worst fit y lista de huecos
	c) First fit y lista de huecos



Para un sistema de memoria paginado con direcciones lógicas de 22 bits y un tamaño de página de 2 Kbytes. Conteste de forma justificada:

- Indique de forma justificada cuál será el número máximo de páginas que podrá tener asignadas un proceso en este sistema.
- Si empleamos paginación a dos niveles, donde la tabla de páginas primer nivel contiene únicamente 8 entradas, ¿cuántas tablas de segundo nivel son necesarias para un proceso que requiere 1024 páginas? Razone su respuesta

1.0 puntos

<b>2</b>	a)	
	b)	

Sea un sistema de memoria virtual con segmentación paginada y las siguientes características:

- Direcciones lógicas de 16 bits
- Memoria física de 1024 marcos
- A cada proceso se le asigna un máximo de 4 marcos
- Páginas de 256 palabras
- 16 segmentos máximo por proceso

Actualmente en memoria hay un único proceso P que ocupa los marcos del 0 al 3, como muestra la figura (formato segmento/página):

Marco	Seg, Pág
0	S0,4
1	S1,9
2	S0,5
3	S2,1

- Exponga el formato de las direcciones lógicas y físicas indicando su estructura, distribución de bits y tamaños

0.75 puntos

<b>3</b>	a)	



- b) Utilizando un algoritmo ÓPTIMO de reemplazo de páginas con ámbito LOCAL muestre la evolución del contenido de la memoria para el conjunto de referencias:

0x1358 0x056D 0x1901 0x216E 0x2178 0x3BD8 0x0567 0x1345 0x1367  
0x0590 0x0500 0x0D33 0x0D23 0x1340 0x3B34 0x3B21 0x21AB

1.0 puntos

<b>3</b> <b>b)</b>	
-----------------------	--

- c) Determine la dirección física que se generará al emitir la dirección lógica 0x21AB, si la ocupación de memoria fuese la siguiente

Marco	Seg, Pág
0	S0,4
1	S1,9
2	S0,5
3	S2,1

0.5 puntos

<b>3</b> <b>c)</b>	
-----------------------	--

- d) Razone si para el mismo conjunto de referencias los algoritmos de reemplazo FIFO, LRU o 2ª oportunidad, producirían un mayor o menor número de fallos de página

0.25 puntos

<b>3</b> <b>d)</b>	
-----------------------	--



Dado el siguiente conjunto de referencias a páginas realizados por los procesos A, B y C (indicadas como proceso/nº de página):

A1,A2,A1,A3,B1,B2,B2,B2,B3,B4,B3,B5,C2,C3,C4,A1,C5,C6,C4,  
C3,B5,B4,B3,A2,A3,A1,B4,B5, B3,C2,C3,C4,B2,B1,A5,A6,A7

- a) Calcule el área activa en el instante final para un tamaño de ventana de 6  
b) Asumiendo un modelo de área activa con tamaño de ventana 6, determine si, en el instante final, podría llegar o no a producirse hiperpaginación, en un sistema con 8 marcos de memoria principal

1.0 punto

<b>4</b>	a) Área activa para un tamaño de ventana de 6
	b) ¿puede producirse hiperpaginación?

Considerando el mapa de memoria de un proceso UNIX, diga si las siguientes sentencias son verdaderas o falsas:

1.0 puntos

<b>5</b>	V/F	Sentencias
		Todas las regiones del mapa de memoria tienen soporte en un archivo
		La región de código con soporte en el archivo ejecutable del proceso tiene permisos de lectura y de ejecución, pero no de escritura
		Para poder mapear un archivo en memoria el proceso ha de ejecutar previamente la llamada "open" sobre el archivo
		Al proyectar un proceso P en su mapa de memoria un fichero f con éxito obtiene un descriptor de fichero asociado a f, que permite a P leer/escribir el contenido de f utilizando las llamadas al sistema read() y write()
		Cuando un proceso realiza una llamada a exec() entonces en su mapa de memoria cambia la región de código que tendrá como soporte el nuevo archivo ejecutable.
		Cuando un proceso crea un proceso hijo con fork() el mapa de memoria del hijo justo después de fork() es idéntico al de su padre
		El fichero con el código ejecutable, de un programa que hace uso de funciones de biblioteca, ocupa un mayor espacio en disco, si se enlazan dinámicamente las bibliotecas

Dado el siguiente fragmento de código en C y primitivas POSIX corresponde a un proceso que al ejecutarlo hereda de su padre una tabla de descriptores de archivos con los tres descriptores estándar

```
.....
int tubo[2];
int fd;

pipe(tubo);
/** Rellene tabla del proceso P1

if (!(pid=fork())) {
    dup2(tubo[1],1);
    close(tubo[0]);
    close(tubo[1]);
/**Rellene tabla del proceso P2
    execlp("/bin/cat", "cat", "fich1", NULL);
}

if(!(pid=fork())){
    dup2(tubo[0],0);
    fd=open("result",O_WRONLY |O_CREAT|O_TRUNC,0666);
    dup2(fd,1);
    close(tubo[0]);
    close(tubo[1]);
    close(fd);
/**Rellene tabla del proceso P3
    execlp("/usr/bin/wc", "wc", "-l",NULL);
}
close(tubo[0]);
close(tubo[1]);
while(pid != wait(&status));
}
```

- a) Rellene la tabla de descriptores de archivo para cada uno de los procesos que intervienen en los instantes marcados con el comentario "Rellene tabla..."
- b) Indique cuál sería la línea de ordenes (comandos, tubos, redirecciones, etc..) que se ejecutaría al lanzar la ejecución de este código.

6

a)

*Tabla de P1*

0	
1	
2	
3	
4	
5	
6	

*Tabla de P2*

0	
1	
2	
3	
4	
5	
6	

*Tabla de P3*

0	
1	
2	
3	
4	
5	
6	

b)



Una partición de 32MBytes, de los cuales 2Mbytes están ocupados con estructuras propias del sistema de archivos, está organizada en bloques de 512 bytes, el puntero a bloque es de 32 bits. Calcule:

- Tamaño máximo de un archivo si se utiliza asignación indexada simple, con un solo bloque de índices por archivo
- Tamaño máximo de un archivo si se utiliza asignación enlazada

7	a) Tamaño máximo de un archivo si se utiliza asignación indexada simple
	b) Tamaño máximo con asignación enlazada

Se ha formateado una partición de 16 Mbytes con Minix, utilizando los tamaños estándar y creando un nodo-i por cada 2kbytes. Los tamaños estándar de Minix son los siguientes: nodos-i de 32bytes (7 punteros directos, 1 indirecto, 1 doble indirecto), entradas de directorios de 16bytes, 1zona=1bloque=1024bytes, puntero a zona de 16 bits.

Determine la estructura de la partición indicando cada uno de los elementos que la componen y el tamaño de cada una de ellas en bloques o zonas.

1.0 puntos

8	
---	--