2. [**2 pts**] En un sistema con planificación de colas de realimentación multinivel se ejecutan tres procesos. En la siguiente tabla se muestran los tiempos de llegada y las duraciones de las sucesivas ráfagas de CPU y E/S de los procesos. El tiempo avanza de izquierda a derecha y se expresa en unidades arbitrarias.

	Llegada	CPU	E/S	CPU	E/S	CPU
P1	0	5	3	7		
P2	1	3	6	2	1	2
P3	3	1	9	9		

Características del planificador:

- El sistema multinivel es EXPROPIATIVO POR PRIORIDAD
- Existen dos colas Cola1 y Cola2.
- La Cola1 es la de mayor prioridad. Los tres procesos se lanzan inicialmente en la Cola 1 (alta prioridad).
- La estrategia de planificación es Round-Robin en ambas colas. El *quantum* de Cola 1 es de 2 unidades de tiempo y la de Cola 2 de 4 unidades.
- Observa que no todos se crean en el mismo instante

Las reglas de movimiento de procesos entre colas son las siguientes:

- Cuando un proceso agota su *quantum* baja de nivel.
- Cuando un proceso no agota su *quantum* asciende de nivel.
- Cuando un proceso se despierta del bloqueo y pasa a "listos" obtendrá un *quantum* renovado (completo).
- Un proceso expropiado por prioridad mantiene la prioridad y pasa al final de la cola de la que partió con *quantum* renovado.

En caso de que en el mismo instante varios procesos lleguen a la vez a la misma cola el orden de simulación será:

1. Proceso nuevo
2. Desbloqueo (si varios bloqueados despiertan en el mismo instante, lo harán en el orden que se bloquearon)
3. Proceso en CPU que agota su quantum
4. Proceso en CPU expropiado por prioridad

Simula la situación descrita y rellena el *diagrama de Gantt,* ó bien *el diario de ejecución* que se adjunta.

---

3. [**2 pts**] Considérense los directorios `/drives/f` y `/drives/g` que se encuentran montados sobre dos volúmenes con formato FAT16, y con tamaño de clúster 4KBytes y 7KBytes respectivamente.
Se muestra a continuación el contenido de un fragmento de la FAT y de un directorio de interés (los valores se expresan todos en decimal):

`/drives/f:`

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
...	29	27	25	*0*	26	24	*-1*	*-1*	20	-1	22	30	*0*	31	*0*	*-1*	...

Filename	*Size*	*Start*
foo.txt	27KB	33
goo.txt	8KB	21
moo.txt	4.5KB	20

`/drives/g:`

	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	
...	*0*	*-1*	*-1*	*-1*	162	*0*	157	158	*-1*	*-1*	190	165	163	161	*0*	151	...

Filename	*Size*	*Start*
alpha.txt	?	156
beta.txt	?	154

Todos los clústers previos a los mostrados están ocupados. La marca de fin de archivo es "-1", y la de clúster libre "0". La política de ubicación consiste en ir tomando el primer clúster libre que se encuentre desde el comienzo de la FAT.

Partiendo del estado que se describe, se realiza la ejecución de la siguiente secuencia de comandos en un *shell*:

```
rm /drives/g/beta.txt
cp /drives/f/foo.txt /drives/g/
```

Se pide:

a) Determinar el tamaño máximo posible para cada uno de los volúmenes

b) Evaluar el impacto de la fragmentación interna de los ficheros mostrados en `/drives/f/`

c) Determinar el estado (contenido) de la FAT y del directorio de `/drives/g/` después de la ejecución de los comandos mostrados. Indica claramente cuáles son los cambios.

	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	
...																	...

Filename	*Size*	*Start*

4. [**2 pts**] Se analiza un sistema de memoria virtual, como el estudiado en clase, direccionado a nivel de byte:

- el espacio de memoria físico abarca 512 Kmarcos
- el espacio lógico (virtual) se compone de 32 Mpáginas
- las tablas se hallan paginadas en fragmentos de tamaño de una página
- todas las entradas de todas las tablas de página son de tamaño 4 bytes (incluyendo bits de control)
- el número de entradas que caben en una página de tabla de páginas es de 128
- el sistema carece de TLB.

Responde a las siguientes cuestiones:

a) Determinar el tamaño de página (en bytes) y los espacios lógicos y físico

b) ¿Cuál es la longitud mínima en bits de una entrada de la tabla de página, excluyendo bits de control?

c) Calcular la longitud en bits de cada uno de los campos en que se divide la dirección lógica:

1er nivel	2º nivel	3º nivel	4º nivel	offset
_____ bits	_____ bits	_____ bits	_____ bits	_____ bits

Justificación:

d) En la figura adjunta se muestra parte del contenido de la memoria. La tabla raíz (nivel 1) está en el marco cero (PTBR=0). Se pide completar la siguiente tabla indicando la información que se pide para cada una de las referencias lógicas de memoria. Indíquese con "Desconocido" aquéllas traducciones que no se pueden realizar con los datos proporcionados. Déjense en blanco aquellas celdas de la tabla que no procedan.

Dirección lógica	Desconocida/ Válida/Inválida	¿Fallo de página? (sí/no)	Dirección física en formato **#pág.fisica:offset**	Contenido (en hexadecimal)
0:2:6:0:2				
3:1:1:3:2				
3:1:2:0:6				
3:2:2:3:6				

e) Determinar la división en campos de la dirección física, en el supuesto que el último nivel de tabla de páginas (L4) requiriera entradas del doble de tamaño (8 bytes), debido a bits de control añadidos a este nivel que no son necesario en los otros.
Suponemos que el tamaño del resto de entradas de los otros niveles se mantienen como se describió originalmente, y que el tamaño de los espacios lógicos, físicos y de la página no cambian.

1er nivel	2º nivel	3º nivel	4º nivel	offset
_____ bits	_____ bits	_____ bits	_____ bits	_____ bits

Justificación:

DIAGRAMA DE GANTT (PROBLEMA 2)

Indica en cada ciclo (casilla) el estado de cada proceso: con **R** (*running*) si está en la CPU ó con **B** (*block*) si está realizando una entrada/salida. Deja la casilla en blanco para aquellos ciclos en los que el proceso está listo en cola o ya ha finalizado. En la fila *idle*, indica con una **L** aquellas casillas correspondiente a ciclos ociosos (idle) de la CPU.

P1																														
P2																														
P3																														
idle																														
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	

P1																														
P2																														
P3																														
idle																														
30	31	32	33	34	35	36	37	38	39	30	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	

P1																														
P2																														
P3																														
idle																														
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	

CONTENIDO DEL SISTEMA DE MEMORIA (PROBLEMA 4)

Notas: Bits de control: V=bit válido, P=bit de presencia.
Los marcos ocupados por páginas de la T.P. se indican con la terna *(V, P, marco_destino)*.
En los marcos con páginas del proceso se indica el contenido.
Los valores precedidos de 0x están expresados en hexadecimal y los que no en decimal.
Las celdas sin valor conocido se hayan en blanco, o indicadas con ?.

PTBR=0

Frame 0	
Offset	Content
0	(V=1, P=1, Frame=1)
4	(V=1, P=0, ?)
8	(V=1, P=0, ?)
12	(V=1, P=1, Frame=3)
...	

Frame 1	
Offset	Content
0	(V=1, P=0, ?)
4	(V=1, P=1, Frame=2)
8	(V=1, P=0, ?)
12	(V=0, P=0, ?)
...	

Frame 2	
Offset	Content
0	(V=1, P=0, ?)
4	(V=1, P=0, ?)
8	(V=1, P=0, ?)
12	(V=0, P=1, ?)
...	

Frame 3	
Offset	Content
0	(V=1, P=1, Frame=4)
4	(V=1, P=1, Frame=6)
8	(V=1, P=1, Frame=9)
12	(V=1, P=0, ?)
...	

Frame 4	
Offset	Content
0	(V=1, P=1, Frame=5)
4	(V=1, P=0, ?)
8	(V=0, P=1, ?)
12	(V=0, P=1, ?)
...	

Frame 5	
Offset	Content
0	(V=1, P=0, ?)
4	(V=1, P=0, ?)
8	(V=1, P=0, ?)
12	(V=0, P=1, ?)
...	

Frame 6	
Offset	Content
0	(V=1, P=0, ?)
4	(V=1, P=1, Frame=7)
8	(V=0, P=1, ?)
12	(V=1, P=0, ?)
...	

Frame 7	
Offset	Content
0	(V=1, P=0, ?)
4	(V=1, P=0, ?)
8	(V=0, P=1, ?)
12	(V=1, P=1, Frame=8)
...	

Frame 8	
Offset	Content
0	0x0000
2	0x282a
4	0x1195
6	0x1291
...	

Frame 9	
Offset	Content
0	(V=1, P=1, Frame=10)
4	(V=1, P=1, Frame=13)
8	(V=1, P=1, Frame=18)
12	(V=0, P=0, ?)
...	

Frame 10	
Offset	Content
0	(V=1, P=0, ?)
4	(V=1, P=1, Frame=11)
8	(V=1, P=1, Frame=12)
12	(V=1, P=0, ?)
...	

Frame 11	
Offset	Content
0	0x5620
2	0x1614
4	0x6604
6	0x0020
...	

Frame 12	
Offset	Content
0	0x0000
2	0x0000
4	0xca00
6	0x0702
...	

Frame 13	
Offset	Content
0	(V=1, P=1, Frame=14)
4	(V=1, P=1, Frame=15)
8	(V=1, P=1, Frame=16)
12	(V=1, P=1, Frame=17)
...	

Frame 14	
Offset	Content
0	0x4040
2	0xe0f0
4	0xc044
6	0x0000
...	

Frame 15	
Offset	Content
0	0x0000
2	0x4040
4	0x6810
6	0x6226
...	

Frame 16	
Offset	Content
0	0x4000
2	0x0000
4	0x1532
6	0x4200
...	

Frame 17	
Offset	Content
0	0x4134
2	0x7050
4	0x0022
6	0x022a
...	

Frame 18	
Offset	Content
0	(V=1, P=1, Frame=19)
4	(V=0, P=0, ?)
8	(V=0, P=1, ?)
12	(V=1, P=1, Frame=20)
...	

Frame 19	
Offset	Content
0	0x1400
2	0x9004
4	0x888c
6	0x80a0
...	

Frame 20	
Offset	Content
0	0x7330
2	0x0101
4	0x0600
6	0x0ac9
...	