



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. Conteste de forma breve y justificada a las siguientes cuestiones:

(1.4 puntos= 4 x 0.35)

1	a) Indique a qué se hace referencia con concurrencia entre CPU y E/S.
	b) Justifique por qué los procesadores tienen al menos dos modos de ejecución y describa cuáles son.
	c) Ordene en dos listas, en función de si forman parte del núcleo del sistema operativo o no, los siguientes elementos: <i>gestor procesos / shell / gestor memoria / manejador dispositivos / comando ls / interfaz de llamadas al sistema / navegador internet</i> . <i>Parte del Núcleo SO:</i> <i>No Núcleo SO:</i>
	d) Defina el concepto de utilización de CPU y exponga mediante expresión matemática cómo se calcula.

2. Considere que en el directorio actual de trabajo existe un archivo "hello.txt", que contiene el texto "hello\n", de forma que la orden `$cat hello.txt` imprime una línea con la palabra hello. Suponga que se compila y ejecuta el siguiente programa, para cada uno de los valores $X=1, 2, 3$ y 4 . Exponga para cada uno de los 4 casos (valores de X), qué se muestra en el terminal al ejecutar el programa y justifique cada una de sus respuestas.

```

1 #include <...all headers...>
2 #define X 1    //1,2,3,4
3
4 int main(int argc, char *argv[])
5 { int val= 0;
6   int parent_pid= getpid();
7
8   if (X>=3) val=fork();
9   if (val==0){
10      if (X%2==1) //X impar
11         execl("/bin/cat", "cat", "hello.txt", NULL);
12      else
13         execl("/cat/bin", "cat", "hello.txt", NULL);
14   }
15
16   if (getpid()==parent_pid)
17      printf("parent\n");
18   else printf("child\n");
19   return 0;
20 }
```

Nota: $a \% b$ devuelve el resto de la división entera a / b .

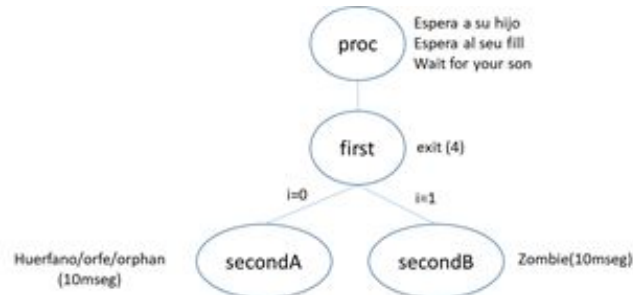
(1.4 punto=0,35+0,35+0,35+0,35)

	a) Caso $X=1$ (#define X 1)
	b) Caso $X=2$ (#define X 2)
	c) Caso $X=3$ (#define X 3)
	d) Caso $X=4$ (#define X 4)

3. Complete el código propuesto para *proc.c* cuyo, ejecutable denominaremos *proc*, de manera que:

- Al ejecutar *proc* se cree un proceso *first* que a su vez crea dos hijos *secondA* y *secondB*.
- El proceso *proc* debe esperar a su hijo *first* y *first* debe finalizar con *exit(4)*.
- Los procesos *secondA* y *secondB*, debe quedarse huérfano y zombi respectivamente durante al menos un tiempo de 10mseg. Para ello se le sugiere utilizar *sleep()*, con *sleep(10)*, *sleep(20)* y *sleep(30)*, cuando lo considere adecuado.

El esquema siguiente muestra el parentesco y las acciones de los diferentes procesos.



(1.2 punto)

```

3. //Programa proc.c
#include <.....los necesarios.h.....>
int main(int argc, char *argv[])
{ int val1, val2, status;
  int i, pid;
  printf("Process proc\n");
  val1=fork();

  while ((pid=wait(&status)) >0)
    printf("hijo esperado %d, estado %d\n", pid, status/256);
  exit(0);
}
  
```

4. En un sistema de tiempo compartido se tiene un único dispositivo de E/S que se gestiona con FCFS. A dicho sistema llegan 3 procesos A, B y C, cuyos instantes de llegada, prioridad (siendo 1 la prioridad más alta, 3 la menor) y esquema de solicitud de ráfagas de CPU y E/S es el siguiente:

Proceso	Llegada	Prioridad	Ráfagas de CPU y E/S
A	0	3 (-)	4 CPU + 1 E/S + 3 CPU + 1 E/S + 4 CPU
B	1	2	2 CPU + 2 E/S + 3 CPU + 1 E/S + 3 CPU
C	2	1 (+)	1 CPU + 5 E/S + 1 CPU + 5 E/S + 1 CPU

(2.0 puntos =1.1+ 0.3+0.3+0.3)

- 4 a) Represente mediante diagrama temporal la ocupación de CPU, del periférico de E/S y de la cola de preparado para un planificador basado **en prioridades expulsivo**. En cada instante de tiempo T ponga entre paréntesis (ej. (A(2))), unidades de tiempo de CPU o E/S pendientes para finalizar dicha ráfaga.

T	Preparados	CPU	Cola E/S-->	E/S	Evento
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					

4 b) Indique los tiempos de espera (de CPU) y de retorno de cada proceso (Tabla PRIO_1)

PRIO 1	T.espera	T.retorno
A		
B		
C		

4 c)	Utilizando el mismo planificador de prioridades expulsivas y carga se obtiene los siguiente valores de tiempos de espera y de retorno. Determine qué prioridades se han asignado a los procesos para obtener estos valores. (Tabla PRIO_2)
------	---

PRIO 2	T.espera	T.retorno
A	0	13
B	7	26
C	14	28

4 d) La siguiente tabla muestra los resultados obtenidos para la misma carga utilizando un planificador Round-Robin con quantum $q=2ut$. (Tabla **RR**)

RR	T.espera	T.retorno
A	8	22
B	8	19
C	6	21

A partir de las tablas PRIO_1, PRIO_2 y RR, calcule la tasa de rendimiento para cada uno de las tres planificaciones

Indique a partir de las tablas PRIO_1, PRIO_2 y RR qué tipo de procesos, limitados por CPU o E/S, son priorizados en cada planificación.

5. Dado el siguiente programa *Hilos.c* cuyo ejecutable es *Hilos*.

(1.5 puntos=0.6 +0.3+0.3+0.3)

<pre>#include <... all headers...> #define NTHREADS 3 pthread_t Th[NTHREADS]; pthread_attr_t atrib; int N=0; void *Func(void *arg) { int i=(int)arg; N=N+i; printf("Hilo %d,N = %d\n",i ,N); pthread_exit(0); }</pre>	<pre>int main (int argc, char *argv[]) { pid_t pid; int i; printf("START MAIN\n"); pthread_attr_init(&atrib); pid=fork(); for (i=0;i<NTHREADS;i++) {pthread_create(&Th[i],&atrib, Func,(void *)i); pthread_join(Th[i],NULL); } if (pid == 0) pthread_exit(0); else exit(0); printf("END MAIN \n"); }</pre>
--	---

5	a) Indique la secuencia (una de las posibles) que imprime el programa en la Terminal al ejecutarlo. Justifique su respuesta.
	b) Indique el número máximo de hilos del programa <i>Hilos</i> que podrían estar ejecutándose concurrentemente. Justifique su respuesta.
	c) Indique si existe riesgo de condición de carrera al ejecutar <i>Hilos</i> . Justifique adecuadamente su respuesta.
	d) Suponga que los hilos de <i>Hilos</i> se soporten a nivel usuario (por el <i>runtime</i>), e indique qué debe gestionar en dicho caso el planificador a corto plazo del sistema. Justifique su respuesta.

6. En un computador con una sola CPU dotado de un sistema operativo con planificador de *turno rotatorio (RR)*, se desea resolver el problema de acceso a sección crítica de **dos** hilos utilizando variantes de *test_and_set()*. Indique **de forma justificada** para cada caso particular propuesto si se cumple la condición de espera limitada y analice si puede ser considerada como espera activa o espera no activa:

(1.2 puntos=0.4+0.4+0.4)

6	a)	Protocolo de entrada: <pre>while (test_and_set(&llave)) /*bucle buit*/ ;</pre>	Protocolo de salida: <pre>llave = 0;</pre>
	Espera limitada		
	Espera activa o no activa		
6	b)	Protocolo de entrada: <pre>while (test_and_set(&llave)) usleep(100) ;</pre>	Protocolo de salida: <pre>llave = 0;</pre>
	Espera limitada		
	Espera activa o no activa		
6	c)	Protocolo de entrada: <pre>while (test_and_set(&llave)) usleep(100) ;</pre>	Protocolo de salida: <pre>llave = 0; usleep(105) ;</pre>
	Espera limitada		
	Espera activa o no activa		

7. Describa una posible secuencia de la ejecución concurrente de los hilos **ThA**, **ThB** y **ThC**

//valores iniciales de las variables int x=0, y=0; Semaphore: S1=0, S2=1, S3=0;		
ThA	ThB	ThC
P(S3); P(S2); x = x + 1; y = 2*x + y; V(S2); V(S1);	P(S1); P(S2); x = 2*x; y = x + y; V(S2);	P(S2); x = x - 2; y = x - y; V(S2); V(S3);

Utilice la tabla siguiente indicando una operación y el hilo al que corresponde en cada línea, así como el valor de las variables y semáforos tras cada operación.

(1.3 puntos)

7	Inicio	ThX-Operación	S1=0	S2=1	S3=0	X=0	Y=0
	1						
	2						
	3						
	4						
	5						
	6						
	7						
	8						
	9						
	10						
	11						
	12						
	13						
	14						
	15						
	16						
	Valores Finales de las variables						