



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

1. Suponga un puesto de trabajo donde debe atender a clientes que le hacen consultas por teléfono y a los que debe responder de forma casi inmediata. Para ello debe llevar a cabo acciones y consultas como usuario en un sistema informático. Escoja el tipo de sistema operativo con el que dotaría a su máquina entre un sistema por lotes multiprogramado, un sistema multiprogramado puro o un sistema de tiempo compartido, justifique su selección. (0,75 puntos)

1	
----------	--

2. a) Indique y justifique las diferencias que existen entre órdenes del Shell y llamadas al sistema operativo, tanto desde el punto de vista conceptual como funcional.
- b) Enumere la secuencia de acciones que se lleva a cabo en un sistema cuando una aplicación de usuario solicita una llamada al sistema. Indique cuales de ellas se realizan en modo núcleo. (0,75 puntos)

2	a)
	b)

3. Dado el siguiente código en C y POSIX correspondiente a un proceso que denominaremos Proc:

```

1 #include ... //los necesarios
2 #define N 3
3
4 main() {
5     int i = 0;
6     pid_t pid_a;
7
8     while (i<N)
9     { pid_a = fork();
10      switch (pid_a)
11      { case -1:
12          printf("Error creating child...\n");
13          break;
14          case 0:
15              printf("Message 1: i = %d \n", i);
16              if (i < N-1) break;
17              else exit(0);
18          default:
19              printf("Message 2: i = %d \n", i);
20              while (wait(NULL)!=-1);
21          }
22      i++;
23  }
24  printf("Message 3: i=%d\n",i);
25  exit(0);
26 }
27
```

- a) Represente el árbol de procesos generado al ejecutarlo e indique para cada proceso el valor de la variable “i” en el instante de su creación.
- b) Indique de forma justificada si existe o no la posibilidad de que los hijos creados queden huérfanos o zombies. (1,0 punto)

3	a)	
	b)	

4. Suponga el siguiente código en C y POSIX correspondiente a un proceso denominado Prueba que se ejecuta con éxito:

```
1 #include ... //los necesarios
2
3 int main()
4 { pid_t val;
5   printf("Mensaje 1: antes de exec()\n");
6   fork();
7   execl("/bin/ls","ls","-la", NULL);
8   val = fork();
9   if (val==0)
10  {execl("/bin/ps","ps","-la", NULL);
11    printf("Mensaje 2: después de exec()\n");
12    exit(1)
13  }
14  printf("Mensaje 3: antes del exit()\n");
15  exit(0);
16 }
```

- a) Indique de forma justificada el número de procesos que se crean al ejecutar Prueba y el parentesco entre ellos.
- b) Indique de forma justificada qué mensajes e información se muestra por pantalla como consecuencia de ejecutar Prueba.

(1,0 punto)

4	a)	
	b)	

5. Dado el siguiente código del proceso Ejemplo:

```
1 #include ... //los necesarios
2
3 int main(void)
4 { int val;
5   printf("Message 1\n");
6   val=fork();
7   /** Aquí deben ir sus modificaciones **/
8   sleep(5);
9   printf("Message 2\n");
10  return 0;
11 }
```



- Indique qué modificaciones serían necesarias introducir en el código anterior para que el proceso hijo se quede huérfano y sea adoptado por el proceso INIT (). (Nota: Utilice el sleep() e instrucciones en C).
- Indique qué modificaciones serían necesarias introducir en el código anterior para que el proceso hijo se quede zombie durante un tiempo. (Nota: Utilice el sleep() e instrucciones en C).
- Indique de forma justificada en qué instrucciones del código propuesto puede asegurarse que ocurrirá un cambio de contexto en la CPU y en cuáles se producirá un cambio de estado para el proceso Ejemplo.

(1,0 puntos)

5	a)	
	b)	
	c)	

6. Dado el siguiente código que intenta solucionar el problema de la condición de carrera, y asumiendo que hay más de un hilo ejecutando concurrentemente el código de la función agrega, indique cuáles de las siguientes sentencias son verdaderas y cuáles falsas (Nota: Un error penaliza una respuesta correcta)

```

1 void *agrega (void *argumento) {
2     long int cont;
3     long int aux;
4     while(test_and_set(&llave)) ;
5     for (cont = 0; cont < REPETICIONES; cont = cont + 1) {
6         V = V + 1;
7     }
8     llave = 0;
9     printf("-----> Fin AGREGA (V = %ld)\n", V);
10    pthread_exit(0);
11 }

```

(1,0 puntos)

6	V/F	
		La línea 4 representa el protocolo de entrada a la sección crítica.
		La solución planteada no garantiza que el código esté libre de condiciones de carrera
		La solución planteada garantiza que el código está libre de condiciones de carrera, ya que, realmente, el acceso a la variable global V se realiza de forma secuencial
		Si intercambiamos las líneas 4 y 5 , y las 7 y 8, la solución propuesta proporciona un código libre de condiciones de carrera
		La función Test_and_Set(&llave) consulta y cambia el valor de llave de forma atómica

7. Dado el siguiente código cuyo archivo ejecutable ha sido generado con el nombre "Hilos1".

```

1 #include "todos los includes necesarios ...."
2 #define DosSegons 2000000
3 pthread_t H1,H2;
4 pthread_attr_t atr;
5
6 void *FaenaB (void *P)
7 { char * texto =(char *) P;
8   usleep(DosSegons);
9   printf("%s\n",texto);
10 }
11
12 void *FaenaA (void *T)
13 { printf("Texto:\n");
14   pthread_create(&H2, &atr, FaenaB, T);
15   usleep(DosSegons);
16   pthread_join(H2, NULL);
17 }
18
19 int main()
20 { pthread_attr_init(&atr);
21   pthread_create(&H1, &atr, FaenaA, "Examen de FSO");
22   usleep(DosSegons);
23   pthread_join(H1, NULL);
24   pthread_join(H2, NULL); /*Hilo creado en FaenaA*/
25   return(0);
26 }
```

- Indique de forma justificada las cadenas que imprime el programa en la Terminal tras su ejecución.
- Indique de forma justificada el tiempo aproximado que tardará en ejecutarse el programa.
- Justifique si en la función main(), sería correcto o no esperar al hilo H2 que es creado al ejecutarse la función FaenaA.
- Durante la ejecución del código, ¿cuántos hilos hay activos cuando escribe "Texto:" en el terminal y cuantos cuando escribe "Examen de FSO"? Justifique su respuesta.

(1,0 puntos)

7	a)	
	b)	
	c)	



d)

8. En un sistema informático conviven tres tipos de procesos: procesos del sistema operativo (PS), procesos de usuario (PU) y procesos en background (PB). Dicho sistema cuenta con un planificador multicola a corto plazo (PCP) con tres colas, una para cada tipo de proceso. La cola PS utiliza un algoritmo de prioridades expulsivas, la cola PU un algoritmo STRF y la cola PB emplea el algoritmo RR con $q=1$. La planificación entre **colas es gestionada con prioridades expulsivas** siendo, como es lógico, la más prioritaria la Cola PS y la menos prioritaria la Cola PB. Las operaciones de E/S se efectúan sobre el mismo dispositivo gestionado con FCFS.

Al sistema llegan 2 procesos de cada tipo en los instantes de tiempo que se indican a continuación:

Proceso	Perfil de ejecución	Instante de llegada	Tipo Proceso
A (-prioridad)	2 CPU	2	Sistema (PS)
B (+prioridad)	3 CPU	3	Sistema (PS)
C	1 CPU + 1 E/S + 1 CPU	6	Usuario (PU)
D	2 CPU + 1 E/S + 1 CPU	17	Usuario (PU)
E	4 CPU + 3 E/S + 2 CPU	0	Background (PB)
F	2 CPU + 1 E/S + 2 CPU	5	Background (PB)

a) Rellene la siguiente tabla indicando en cada instante de tiempo donde se encuentran los procesos.

b) Indique los tiempos medios de espera y retorno y la utilización de CPU para esta carga.

2.0 puntos (1.25+0.75)

8a	T	Cola PS Prio. exp	Cola PU SRTF	Cola PB RR $q=1$	CPU	Cola E/S	E/S	Evento
	0							Llega E
	1							
	2							Llega A
	3							Llega B
	4							
	5							Llega F
	6							Llega C
	7							
	8							
	9							
	10							
	11							
	12							
	13							
	14							
	15							
	16							
	17							Llega D
	18							
	19							
	20							
	21							
	22							
	23							
	24							



8b	<p>Tiempo medio de espera =</p> <p>Tiempo medio de retorno =</p> <p>Utilización de CPU =</p>
-----------	--

9. Dado el siguiente programa, donde se utilizan tres semáforos, con los hilos agrega y resta. Indique de forma justificada para cada una de los valores de x e y propuestos, si puede haber o no condición de carrera, el estado en que quedan los hilos creados y el valor final de V que se escribe en la línea 12:

<pre>#include ... //suponga los includes int V = 100; sem_t sem,sum,res; void *agrega (void *argumento) { int cont; for (cont=0; cont<100; cont++) { sem_wait(&sum); sem_wait(&sem); V = V + 1; sem_post(&sem); sem_post(&res); } }</pre>	<pre>void *resta (void *argumento) { int cont; for (cont=0; cont<100; cont++) { sem_wait(&res); sem_wait(&sem); V = V - 1; sem_post(&sem); } }</pre>
<pre>1. int main (void) { 2. pthread_t hiloSuma, hiloResta, hiloInspeccion; 3. pthread_attr_t attr; 4. int x,y; 5. pthread_attr_init(&attr); 6. sem_init(&sem,0,1); 7. sem_init(&sum,0,x); 8. sem_init(&res,0,y); 9. pthread_create(&hiloSuma, &attr, agrega, NULL); 10. pthread_create(&hiloResta, &attr, resta, NULL); 11. usleep(80000000); //suficiente para ejecutar agrega y resta 12. fprintf(stderr, "-----> VALOR FINAL: V = %d\n", V); 13. exit(0); 14. }</pre>	

- a) Suponga que x=5 e y=1.
b) Suponga que x=500 e y=1.
c) Suponga que x=20 e y=5.

(1.5 puntos)

9	a)
----------	----



	b)
	c)