NIA: _____

Nombre y apellidos _____

Primer Parcial de Sistemas Operativos.
18 de octubre de 2010.

NOTAS:

* La fecha de publicación de las notas, así como de revisión se notificarán por Aula Global.
* Para la realización del presente examen se dispondrá de 1:15 hora.
* No se pueden utilizar libros ni apuntes
* Será necesario presentar el DNI o carnet universitario para realizar la entrega del examen

---

Ejercicio 1 (4 puntos). Responda a las siguientes preguntas:

1) ¿Qué es un proceso ligero?. ¿Qué ventajas proporciona programar con procesos ligeros?

Una unidad de ejecución que comparte el espacio de memoria con otros procesos ligeros del mismo proceso (pesado), pero con diferente pila y estado de procesador.

2) ¿Qué es un proceso? ¿De que tres principales partes se compone la información de un proceso? Describa brevemente otros contenidos del BCP.

Un proceso es un programa en ejecución
Imagen de proceso, registros del procesador y BCP
Otros contenidos del BCP: estado del procesador, identificación de usuario, etc.

3) ¿Qué formas hay de activar el sistema operativo? Ponga, al menos, un ejemplo de cada una.

Por interrupción: Falta papel en impresora.
Por TRAP llamada fork de sistema
Por excepción división por cero.

4) Indicar cuáles de las siguientes transiciones entre los estados de un proceso no se pueden producir en un sistema con un algoritmo de planificación de "Primero el proceso más corto sin expulsión"
   - Bloqueado a listo.
   - Ejecutando a listo.
   - Ejecutando a bloqueado.
   - Listo a bloqueado.
   - Listo a ejecutando

   Explica porqué

Ejecutando a listo, ya que si la planificación es sin expulsión, un proceso solo podrá pasar a estar bloqueado o terminar.

NIA: _____

Nombre y apellidos _____

Listo a bloqueado, un proceso listo, si no pasa a ejecución no tiene motivos para pasar a estar bloqueado, podría pasar a listo en segundo plano.

NIA: _____

Nombre y apellidos _____

Ejercicio 2 (3 puntos)

Se desea implementar un proceso que lea un número desde el teclado y comience a escribir una secuencia de números consecutivos a partir del número introducido. Simultáneamente deberá seguir leyendo del teclado y en el momento en que se produzca otra entrada cambiará la secuencia de números que aparecen por pantalla. Un ejemplo de ejecución sería el siguiente:

Ejecución en el Terminal	Explicación de la ejecución
$> miprograma	
27	Número introducido por teclado
28	Salida proporcionada por el programa
29 27	
30 28	
31 29	
32 30	
5 Número introducido por teclado	Número introducido por teclado
6	Salida proporcionada por el programa
7 15	
8 16	
3	Número introducido por teclado
4	Salida proporcionada por el programa
5 3	
0	Número introducido por teclado
Fin del programa	Salida del programa. Fin del programa
$> -	Espera nueva entrada

Puesto que no se pueden simultanear en un mismo proceso la lectura teclado y escritura en la pantalla, se decide realizar un programa con un proceso padre que continuamente está leyendo por teclado, si la entrada es un número entero lanza un proceso hijo cuya misión consiste en escribir números consecutivos a partir del introducido. Cuando el usuario introduzca otro número el padre mata al proceso que estaba escribiendo y lanza un nuevo hijo con la misma labor. El programa termina cuando el usuario introduce un 0, en este caso el padre mata al proceso en ejecución y termina el programa sacando por pantalla el literal Fin del programa.

NIA: _____

Nombre y apellidos _____

Solución:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

main ( )
{
        int  contador;
        pid_t pid;

        scanf (%d,&contador);
        while (contador)
        {
                pid = fork();
                switch (pid)
                {
                        case 0:                 //estoy en el hijo
                                while (1)       printf(%d,++contador );
                        case 1:
                                perror( error al ejecutar el fork)
                        default:
                                scanf(%d, &contador);
                                kill(pid,9);
                }
        }
        printf(Fin del programa);
}
```

NOTA: Para poder ver la ejecución del hijo correctamente sería necesario incluir un delay.

NIA: _____

Nombre y apellidos _____

Ejercicio 3 (3 puntos)

En un determinado sistema operativo los procesos se ejecutan en función de colas multinivel con las siguientes características:

- El sistema tiene 3 colas:
  - La primera sigue un algoritmo de planificación Round-Robin con quantum de 2ms.
  - La segunda sigue un algoritmo Round-Robin con quantum de 4 ms.
  - La tercera sigue una planificación FIFO.
- Los procesos entran en el sistema por la primera cola.
- Los procesos son degradados de cola si el sistema los expulsa del procesador por vencimiento de quantum.
- La planificación entre colas es por prioridades, siendo la más prioritaria la primera, luego la segunda y después la tercera.

Se pide:

a) Dibujar el cronograma de estados de ejecución de los procesos que se muestran a continuación:

PROCESOS	T. DE LLEGADA	T DE EJECUCIÓN
P1	0	1ms CPU + 6ms E/S + 1ms CPU
P2	1	3 ms CPU
P3	3	5ms CPU + 3ms E/S + 1ms CPU
P4	3	3 ms CPU

Realizar la solución en la siguiente tabla:

Proceso	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P1																
P2																
P3																
P4																

b) Indicar para cada proceso su tiempo de estancia en el sistema y el tiempo de penalización que sufre cada uno de ellos.

c) ¿Cuál es el proceso peor tratado?

Solución:

NIA: _____

Nombre y apellidos _____

a)

Proceso	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P1	CPU	E/S	E/S	E/S	E/S	E/S	E/S	CPU								
P2		CPU	CPU						CPU							
P3				CPU	CPU					CPU	CPU	CPU	E/S	E/S	E/S	CPU
P4						CPU	CPU						CPU			

b)

Proceso	Tiempo de estancia	Tiempo de penalización
P1	8	0
P2	8	5
P3	13	4
P4	10	7

c)
El proceso peor tratado es el proceso 4 ya que tarda 10 periodos en terminar su ejecución, cuando solo tiene 3 periodos de ejecución, sufre 7 periodos de penalización.