

Parte 1 (20 puntos)**Tiempo estimado: 30 min.**

1.- Indica 4 similitudes y 4 diferencias entre los procesos pesados y los threads implementados por el núcleo del S.O. Escribirlo de la forma:

Similitudes

1
2
3
4

Diferencias

1
2
3
4

2.- Un ordenador tiene cuatro marcos de página, con las páginas que se indican cargadas en cada marco. Además se indica para cada página cargada: el instante de carga, el contador para el algoritmo LRU-Aging y los bits de referencia (R) y modificado (M).

Página	Marco	Instante de carga	Cont-Aging	R	M
27	0	126	00011	1	0
50	1	230	11000	1	0
41	2	120	01111	1	1
12	3	160	10111	0	1

Supongamos que hay que cargar una nueva página en memoria, responda a las siguientes cuestiones:

- ¿Qué página es reemplazada por NRU (Not Recently Used)?
- ¿Qué página es reemplazada por FIFO?
- ¿Qué página es reemplazada por segunda oportunidad?
- Si llega una interrupción de reloj, ¿qué valores tomarán los contadores del algoritmo LRU-Aging?

3.- Define los siguientes términos en relación a la gestión de procesos.

- Eficiencia
- Tiempo de retorno (turnaround)
- Tiempo de respuesta (response time)
- Rendimiento (throughput)

4.- Cuando en un disco se utiliza la técnica de *interleaving*, existe un tiempo t_1 que tiene que ser mayor que un tiempo t_2 para que las lecturas se hagan eficientemente. Indique a qué tiempos se refieren t_1 y t_2 .

t_1 : tiempo que tarda ...

t_2 : tiempo que tarda ...

Cuando en un disco se utiliza la técnica de *skew*, existe un tiempo t_1 que tiene que ser mayor que un tiempo t_2 para que las lecturas se hagan eficientemente. Indique a qué tiempos se refieren t_1 y t_2 .

t_1 : tiempo que tarda ...

t_2 : tiempo que tarda ...

5.- Indica tres métodos que se pueden utilizar para que un sistema operativo controle los bloques de disco donde se almacena la información de cada fichero. Explica brevemente en qué consiste cada uno.

	Nombre	Breve descripción
Método 1		
Método 2		
Método 3		

EJERCICIO 1 (35 puntos)**Tiempo estimado: 45 min.**

Considérese el siguiente programa:

```

#define Tam1 1024
#define Tam2 6

int A[Tam1] = {34, 22, ... }; /* inicialización del vector */
int B[Tam2]; /* vector B sin inicializar */
int x;
int register I, J; /* variables almacenadas en registros */

main () {
    x = 7; /* → Punto inicial */
    for (I = 0; I < Tam1; I++)
        B[0] = x + A [I] ;
    for (J = 1; J < Tam2; J++)
        B[J] = B[1] ;
}

```

Supongamos que la imagen en memoria virtual de un proceso tiene 4 secciones disjuntas: Código, Datos Inicializados, Datos Sin Inicializar y Pila. Estas secciones no comparten páginas. El sistema de memoria virtual utiliza paginación por demanda con asignación fija, reservando 4 marcos en memoria para cada proceso. El algoritmo de reemplazo es FIFO de ámbito local. Las páginas tienen un tamaño de 1k.

Un "integer" ocupa 4 bytes de longitud. El código ejecutable ocupa menos de 1k bytes. El único proceso que tenemos en el sistema es el que está ejecutando el programa descrito anteriormente. Las variables I y J son almacenadas en registros internos de la CPU de forma que su utilización solo genera acceso a memoria la primera vez que se utilizan. En el instante inicial no está cargada ninguna página del programa y el sistema se dispone a ejecutar la instrucción "x=7".

Se pide:

1. Escribir un mapa de memoria virtual que muestre qué partes del proceso están cargadas en cada página virtual. Se deberá indicar en hexadecimal la dirección de inicio y final de cada página virtual.
2. Indicar el orden en que se hace referencia a las páginas virtuales durante toda la ejecución de este proceso.
3. Calcular el número de faltas de página que se producen.
4. Dibujar una TLB y una TP que contengan los valores adecuados para el caso en que se produzca una falta de página al acceder a un elemento de la tabla B. En la TP y la TLB deberán figurar, en hexadecimal o binario, los contenidos que hacen que se produzca la falta de página.
5. Dibujar una TLB y una TP que contengan los valores adecuados para el caso en que una dirección virtual se traduzca con éxito al acceder a un elemento de la tabla B. En la TP y la TLB deberán figurar, en hexadecimal o binario, los contenidos que hacen que se produzca la traducción de la dirección virtual.

EJERCICIO 2 (25 puntos)**Tiempo estimado: 30 min.**

Escriba un programa ejemplo_fork2.c con dos procesos, padre e hijo. El hijo escribirá su identificador de proceso y el identificador de su proceso padre. El padre debe esperar a que acaben sus dos hijos. A medida que acaban, el padre escribirá por la salida estándar el identificador del proceso que acaba de morir.