

Problema 1 (2.5 puntos)

En un sistema se ejecutan n procesos con las siguientes características:

- Cada proceso i tiene una duración de i cs.
- Cada proceso i realiza una operación de entrada/salida sobre un único dispositivo en exclusión mutua de 1 ms de duración cada i ms que se haya ejecutado en el procesador.
- Cada proceso i llega al sistema en el instante $i-1$ ms.

El sistema tiene las siguientes características:

- Un solo procesador.
- Algoritmo de planificación RR con quantum de 1 ms.
- Cuando la llegada de un nuevo proceso coincide con la salida de un proceso de ejecución, entonces el proceso que sale de ejecución se coloca detrás del nuevo proceso.
- Cuando la salida de un proceso en estado de espera coincide con la salida de un proceso de ejecución, entonces el proceso que sale de ejecución se coloca detrás de proceso que viene de espera.

Para una carga de $n = 3$ procesos, mostrar la evolución temporal de los procesos del sistema señalando el estado en el que se encuentra cada proceso, así como la ocupación temporal de la CPU y de los dispositivos de E/S. Calcular los tiempos de respuesta, de retorno y de espera para cada uno de los procesos.

Problema 2 (3 puntos)

En una nave industrial existe una máquina de inyectar que deja cada pieza producida en una cinta transportadora de tamaño limitado. Existe un robot que recoge las piezas de la cinta (una cada vez) y las deja en las cajas de embalaje. Finalmente, tenemos un operario que, de vez en cuando, recoge 3 piezas para realizar el control de calidad, si no hay tres piezas en la cinta lo intentará más tarde sin bloquearse.

Resuelve el escenario anterior mediante semáforos.

Problema 3 (2 puntos)

Supongamos que tenemos una máquina con 16 MB de memoria principal y un esquema de gestión de memoria virtual paginado con páginas de 4 KB. Un proceso produce la siguiente secuencia de accesos a direcciones de memoria (mostradas aquí en hexadecimal):

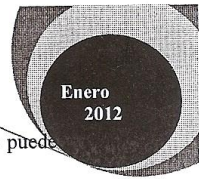
02D4B8, 02D4B9, 02D4EB, 02D4EB, 02D86F, F0B621, F0B815, F0D963, F0B832, F0BA23, D9D6C3, D9B1A7, D9B1A1, F0BA25, 02D4C7, 628A31, F0B328, D9B325, D73425.

El sistema operativo asigna al proceso 4 marcos de memoria principal. Se pide:

- a) Indicar cuál es el formato de una dirección física de memoria.
- b) Dar la cadena de referencia de las páginas accedidas por el proceso.
- c) Si el sistema operativo utiliza 4 marcos de memoria principal, describir el comportamiento del gestor de memoria utilizando cada uno de los siguientes algoritmos de reemplazo de páginas, indicando cuántos fallos de página se producen con los algoritmos FIFO y de la segunda oportunidad.

Problema 4 (1,5 puntos)

Un sistema operativo específico implementa un sistema de archivos híbrido indexado-enlazado con la siguiente estructura para cada archivo: 10 punteros directos, un puntero indirecto simple, un puntero indirecto doble y, finalmente, un puntero a una lista enlazada de bloques hasta un máximo de 1000 bloques. Si tenemos un dispositivo de almacenamiento secundario con una capacidad de 4 Terabytes y los



bloques son de 1 kbyte. Determina el tamaño máximo de los datos de un archivo (en bytes) que puede gestionar el sistema operativo en ese dispositivo.

Problema 5 (1 puntos)

Un planificador de disco que tiene 200 pistas (de 0 al 199). Está inicialmente en la 100 y se mueve en sentido descendiente. La cola de peticiones pendientes es: 13, 149, 88, 191, 93, 150, 101, 183, 134. Indicar cuántos desplazamientos de pista (número de pistas que atraviesa) se producen en total si el planificador está empleando el algoritmo SCAN. Comenta si es cierto que el número de desplazamientos es el mismo que si se emplease el algoritmo LOOK.

Normas

- Es necesario solucionar cada problema en hojas separadas.
- La duración del examen es de 2 h.

Problema 1

Enunciado:

Para $n = 3$

Tiempo ejecución:

$$1 \text{ cs} = 10^1 \text{ ms}$$

$$P1: 1\text{cs} = 10^1 = 10 \text{ ms.}$$

$$P2: 2\text{cs} = 2 \cdot 10^1 = 20 \text{ ms}$$

$$P3: 3\text{cs} = 3 \cdot 10^1 = 30 \text{ ms}$$

Tiempo de llegada:

$$P1: i-1 = 1-1 = 0 \text{ ms.}$$

$$P2: i-1 = 2-1 = 1 \text{ ms}$$

$$P3: i-1 = 3-1 = 2 \text{ ms}$$

Entrada y salida:

P1:

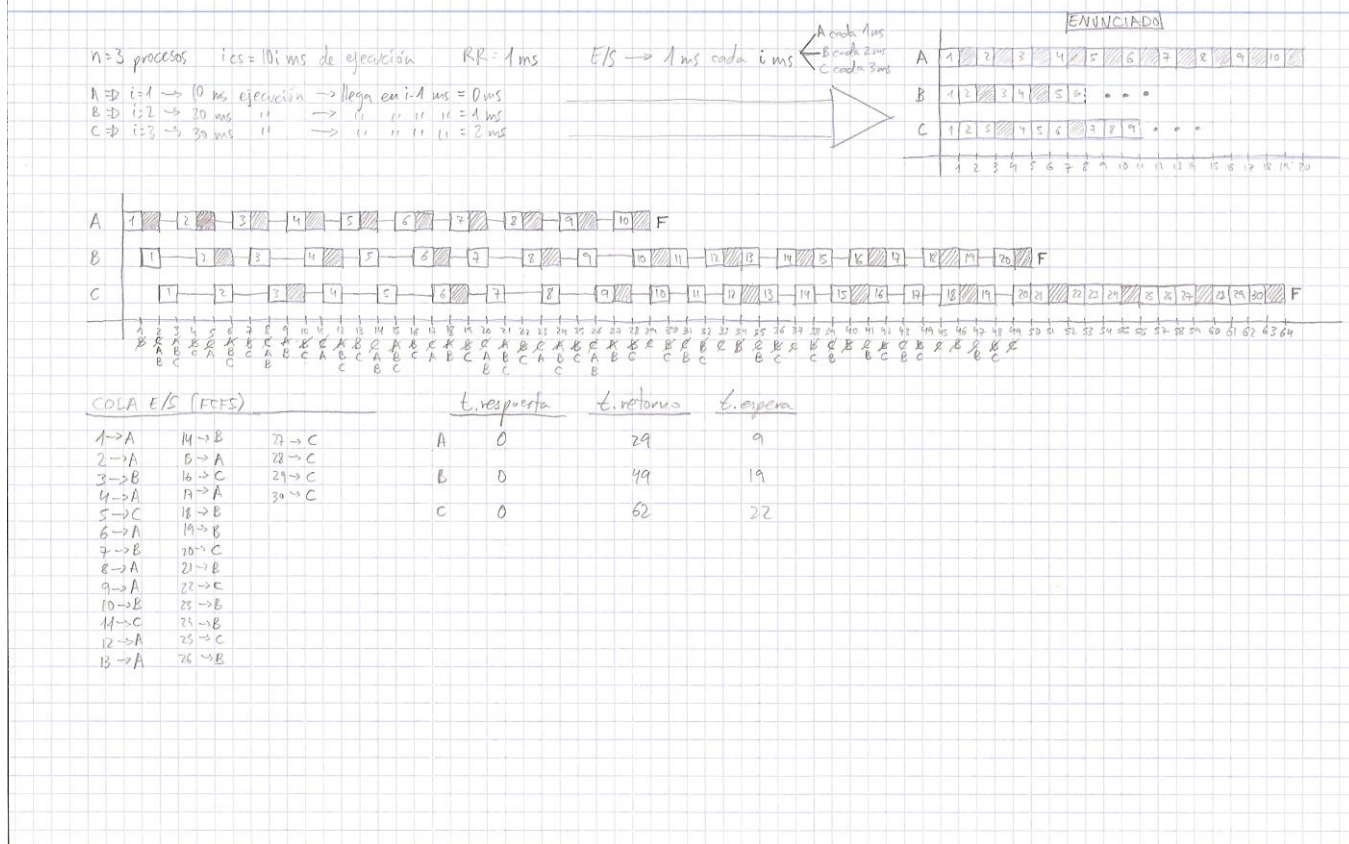
P1	E/S	P1	E/S	P1	E/S
----	-----	----	-----	----	-----

P2:

P2	P2	E/S	P2	P2	E/S
----	----	-----	----	----	-----

P3:

P3	P3	P3	E/S	P3	P3	P3
----	----	----	-----	----	----	----



Tiempos:

	t. retorno	t. respuesta	t. espera
P1	29	0	$29 - 20 = 9$
P2	49	0	$49 - 30 = 19$
P3	62	0	$62 - 40 = 22$

Problema 2

```
#define cinta n
TSemaforo s, n, e;
int piezas;

void inyector() {
    while(true) {
        coger_pieza();
        P(e);
        P(s);
        piezas++;
        inyectar();
        V(s);
        V(n);
    }
}

void robot() {
    while(true) {
        P(n);
        P(s);
        coger_pieza();
        piezas--;
        V(s);
        V(e);
        embalar();
    }
}

void operario() {
    while(true) {
        P(s);
        if(piezas >= 3) {
            piezas = piezas - 3;
            V(e); V(e); V(e);
            control_de_calidad();
        }
        V(s);
    }
}

void menu() {
    inicializar(s, 1);
    inicializar(n, 0);
    inicializar(e, n);
    piezas = 0;

    cobegin
        inyector(); robot(); operario();
    coend;
}
```

Problema 3

Solución:

a) A partir del tamaño y organización de la memoria, se calcula el formato de una dirección física:

16 MB = 2^{24} bytes, es decir, se requieren 24 bits para direccionar la memoria física. Como además se nos indica que el tamaño de página es de 4 KB, tendremos que

4 KB = 2^{12} bytes, por lo que se necesitan 12 bits para direccionar un byte de una página.

Por tanto, el formato de una dirección física de memoria será:

página	desplazamiento
12 bits	12 bits

Dado que la cadena de referencias viene dada en hexadecimal, y para expresar un número hexadecimal se necesitan 4 bits en binario, para direccionar la página se

OJO: El FIFO parece que está mal a partir de la secuencia D9B

necesitan 3 dígitos hexadecimales y otros 3 para el desplazamiento. Por tanto la una dirección física queda expresada según el siguiente formato:

página			desplazamiento		
12 bits			12 bits		
hex	hex	hex	hex	hex	hex
0	2	D	4	B	8

Así, la secuencia de páginas que formará la cadena de referencias es:
02D, F0B, F0D, F0B, D9D, D9B, F0B, 02D, 628, F0B, D9B, D73

b.1) FIFO

02D	F0B	F0D	F0B	D9D	D9B	F0B	02D	628	F0B	D9B	D73
02D	02D	02D		02D	F0B		F0D	D9D	D9B		02D
	F0B	F0B		F0B	F0D		D9D	D9B	02D		628
		F0D		F0D	D9D		D9B	02D	628		F0B
				D9D	D9B		02D	628	F0B		D73
F	F	F		F	F		F	F	F		F

En total se producen 9 fallos de página.

b.2) Segunda Oportunidad

02D	F0B	F0D	F0B	D9D	D9B	F0B	02D	628	F0B	D9B	D73
02D ₁	02D ₁	02D ₁		02D ₁	D9B ₁	D9B ₁	D9B ₁	D9B ₁	D9B ₁		D73 ₁
	F0B ₁	F0B ₁		F0B ₁	F0B ₀	F0B ₁	F0B ₀	F0B ₀	F0B ₁		F0B ₀
		F0D ₁		F0D ₁	F0D ₀	F0D ₀	02D ₁	02D ₁	02D ₁		02D ₀
				D9D ₁	D9D ₀	D9D ₀	D9D ₀	628 ₁	628 ₁		628 ₀
F	F	F		F	F		F	F			F

En total se producen 8 fallos de página.

Problema 4

La capacidad del disco es de 4Terabytes = 2^{42} bytes ($1T = 2^{40}$).

Cada bloque ocupa 1kbyte = 2^{10} bytes.

¿Cuántos bloques podremos tener?

$$2^{42}/2^{10} = 2^{32} \text{ bloque podemos direccionar.}$$

¿Cuántos bits necesitamos para direccionar 2^{32} bloques?

$$32 \text{ bits} = 4 \text{ bytes.}$$

Tenemos bloques de datos, otros de punteros y otros “mixtos”.

Parte indexada

¿Cuántos punteros caben en un bloque?

Tam. Bloque / Tam. Puntero = $1\text{kb}/4\text{bytes} = 2^{10}/4 = 256$ punteros caben en 1 bloque.

Para los punteros: Num. Punteros * Tam. Bloque * Num. Punteros ^ nivel indireccion.

El nivel de indirección para un puntero directo = 0, indirecto simple = 1, indirecto doble = 2 e indirecto triple = 3.

Directo	$10 * 1\text{kb}$	$= 10 \text{ kb}$
Indirecto simple	$1 * 1\text{kb} * 256^1$	$= 256 \text{ kb}$
Indirecto doble	$1 * 1\text{kb} * 256^2$	$= 65536 \text{ kb}$

En total, $10 + 256 + 65536 = 65802\text{kb}$ ó bloques de punteros hacia bloques de datos.

Parte enlazada

Podemos tener máximo 1000 bloques.

Como un bloque tiene 1024 bytes (1kb) entonces, $1024 - 4 = 1020$ bytes se emplean en un bloque para datos. Si tenemos 1000 bloques tendremos $1020 * 1000 = 1020000 \text{ bytes}$ para datos en total.

Sumando las dos partes: $65802 \text{ kb} + 1020000 \text{ bytes} = \underline{\underline{68401248 \text{ bytes.}}}$

Problema 5

SCAN

Pistas a las que se accede

93 88 13 101 134 149 150 183 191

Numero de pistas que se atraviesan

7 5 75 114 33 15 1 33 8 = **291 pistas**

Si empleamos el algoritmo LOOK no obtenemos el mismo número de pistas (obtendríamos menos), ya que, el LOOK recorre las pistas (265 pistas) a acceder en una dirección hasta llegar a la última petición. En este punto se invierte el sentido y se prosigue de la misma forma.