

Material permitido: Solo calculadora no programable	Aviso 1: Todas las respuestas deben estar debidamente razonadas.
Tiempo: 2 horas	Aviso 2: Escriba con buena letra y evite los tachones.
N2	Aviso 3: Solución del examen y fecha de revisión en http://www.uned.es/71902048/

1. Conteste **razonadamente** a las siguientes preguntas:

- a) (1 p) Explicar la diferencia entre un *enlace duro* y un *enlace simbólico*.
- b) (1 p) ¿Bajo que condiciones el algoritmo de planificación de peticiones de E/S a disco se implementa en el driver del disco?

2. Enumerar las ventajas y los inconvenientes de:

- a) (1 p) Los hilos a nivel de usuario.
- b) (1 p) Los hilos a nivel de núcleo.

3. (2 p) Enumerar los pasos del *algoritmo del banquero*. ¿Qué inconvenientes presenta su uso?

4. (2 p) El sistema operativo en colaboración con el hardware gestiona la memoria principal usando la técnica de demanda de página con un tamaño de página de 4 KiB. La memoria principal del computador tiene una capacidad de 256 MiB con un tamaño de palabra de 16 bits. La unidad direccionable es la palabra. Por otra parte el espacio de direcciones virtuales de un proceso A ocupa 128 MiB. Determinar el tamaño en bits de cada uno de los campos en que se descompone una dirección física y una dirección virtual del proceso A.

Nota: 1 KiB=1024 bytes.

5. (2 p) Una persona tiene en su casa una jaula llena de canarios en la que hay un plato de alpiste y un columpio. Todos los canarios quieren primero comer del plato y luego columpiarse, sin embargo sólo tres de ellos pueden comer del plato al mismo tiempo y solo uno de ellos puede columpiarse. En la Figura 1 se muestra un pseudocódigo basado en C de un programa que usando un **monitor** de nombre `jaula` coordina la actividad de los canarios. Este monitor considera la solución de Hansen en el comportamiento de la operación `signal_mon`. Explicar **razonadamente** si este pseudocódigo permite la máxima concurrencia de procesos, es decir, que al mismo tiempo tres canarios puedan estar comiendo y otro cuarto pueda estar columpiándose.

```

#define N 3 /* Número de puestos en el plato */
monitor jaula /* Definición del monitor */
    condición puesto_plato_disponible, columbia_disponible;
    int contadorP, contadorC;

    void obtener_puesto_en_plato() /* Procedimiento del monitor */
    {
        if (contadorP == N) wait_mon(puesto_plato_disponible);
        contadorP=contadorP+1;
        comer();
    }

    void dejar_puesto_plato() /* Procedimiento del monitor */
    {
        contadorP = contadorP - 1;
        signal_mon(puesto_plato_disponible);
    }

    void obtener_columbia() /* Procedimiento del monitor */
    {
        if (contadorC == 1) wait_mon(columbia_disponible);
        contadorC=contadorC+1;
        columpiarse();
    }

    void dejar_columbia() /* Procedimiento del monitor */
    {
        contadorC = contadorC - 1;
        signal_mon(columbia_disponible);
    }

    { /* Inicialización del monitor */
        contadorP=0, contadorC=0;
    }
end monitor

void canario() /* Proceso canario */
{
    jaula.obtener_puesto_plato();
    jaula.dejar_puesto_plato();

    jaula.obtener_columbia();
    jaula.dejar_columbia();
}

main() /* Ejecución concurrente */
{
    ejecución_concurrente(canario,...,canario);
}

```

Figura 1