

Material permitido: **Solo calculadora no programable**

Tiempo: **120 minutos**

N

Aviso 1: Todas las respuestas deben estar debidamente razonadas.

Aviso 2: Escriba con buena letra y evite los tachones.

Aviso 3: Solución del examen y fecha de revisión en <http://www.dia.uned.es/71902048/>

1. Conteste razonadamente a las siguientes preguntas:

- a) (1 p) ¿Qué es un sistema operativo? ¿Cuáles son sus principales objetivos?
- b) (1 p) ¿Qué es un proceso? ¿Qué tipos de procesos se pueden distinguir de forma general?
- c) (1 p) ¿Qué función realiza el planificador a corto plazo de un sistema operativo? ¿Cuáles son sus componentes?
- d) (1 p) Describir el funcionamiento del algoritmo de reemplazamiento de páginas WS-Clock.
- e) (1 p) Describir el método de asignación de espacio de disco conocido como asignación indexada, ¿Cuáles son sus ventajas e inconvenientes?

2. (1 p) En una oficina municipal de atención al ciudadano existen 3 ventanillas. Cuando un ciudadano entra en la oficina para realizar alguna gestión debe guardar una única cola hasta que alguna ventanilla queda libre. Escribir el pseudocódigo de un programa que coordine la actividad de los ciudadanos en la oficina usando **semáforos binarios**.

Nota 1: Antes de escribir el pseudocódigo se debe explicar adecuadamente el significado de cada una de las variables globales y semáforos binarios que se van a utilizar en el mismo.

Nota 2: En la resolución de este ejercicio se deben utilizar las operaciones para semáforos binarios definidas en el libro base de la asignatura.

3. Supóngase un sistema con diez recursos en el que se ejecutan los siguientes procesos:

- Proceso A. Necesita como máximo cinco recursos.
- Proceso B. Necesita como máximo seis recursos.
- Proceso C. Necesita como máximo cuatro recursos.

Utilizando el *algoritmo del banquero* determinar si los siguientes estados son estados seguros:

- 1) (1 p) Estado 1. El proceso A usa dos recursos, el proceso B usa un recurso y el proceso C usa tres recursos.
- 2) (1 p) Estado 2. El proceso A usa tres recursos, el proceso B usa cuatro recursos y el proceso C usa dos recursos.

Material permitido: Solo calculadora no programable	Aviso 1: Todas las respuestas deben estar debidamente razonadas.
Tiempo: 120 minutos	Aviso 2: Escriba con buena letra y evite los tachones.
N	Aviso 3: Solución del examen y fecha de revisión en http://www.uned.es/71902048/

4. En un sistema x86-64 bits la unidad direccionable es un byte y se suelen usar 48 bits para las direcciones lógicas. Supuesto que la memoria principal tiene un tamaño de 16 GiB y un tamaño de página de 4KiB:
- (1 p) Determinar el tamaño en bits de cada uno de los campos en que se descompone una dirección física y una dirección lógica.
 - (0.5 p) Determinar el tamaño máximo que puede tener el espacio de direcciones lógicas de un proceso y el número de páginas en que se divide.
 - (0.5 p) Supuesto que una entrada de una tabla de páginas tiene un tamaño de 8 bytes, determinar el tamaño que tendría una tabla de páginas completa. De acuerdo, con el resultado anterior determinar si se podría almacenar una tabla de páginas en la memoria principal de este sistema.

SISTEMAS OPERATIVOS (Cód. 71902048)

Solución Examen Septiembre 2025

Solución Ejercicio 1

- a) Un *sistema operativo* es una capa de software que gestiona de forma eficiente todos los dispositivos hardware de un computador y además suministra a los usuarios una interfaz cómoda con el hardware.

De la definición anterior se concluye que los principales objetivos de un sistema operativo son:

- *Gestionar o administrar eficientemente los dispositivos hardware de un computador.* El sistema operativo se encarga de decidir cuánto tiempo de ejecución en el procesador se asigna a los programas de los usuarios. El sistema operativo dispone de un módulo denominado planificador que se encarga de esta tarea. Por otra parte, el sistema operativo también se encarga de administrar, junto con el hardware, la ocupación de la memoria principal. También el sistema operativo se encarga de gestionar la memoria secundaria. Finalmente, el sistema operativo gestiona el acceso a los restantes dispositivos de E/S.
- *Ofrecer a los usuarios una interfaz cómoda con el hardware que les facilite el uso del computador.* El sistema operativo proporciona al usuario una máquina extendida, también denominada máquina virtual, que es más fácil de programar que el hardware desnudo.

- b) Un programa es un archivo ejecutable que típicamente se encuentra almacenado en memoria secundaria. Considerado como archivo, un programa es una entidad pasiva o estática. Cuando un programa es ejecutado se convierte en una entidad activa o dinámica denominada *proceso* que va pasando por diferentes estados y utilizando distintos recursos del computador. En conclusión, *un proceso es un programa en ejecución.*

También se define el término *proceso* como la entidad que se puede asignar y ejecutar en un procesador, es por tanto la unidad básica de trabajo de un sistema informático.

De forma general se pueden distinguir tres tipos de procesos:

- *Procesos de usuario.* Son procesos asociados a la ejecución de programas invocados por los usuarios. Se ejecutan en modo usuario excepto cuando realizan llamadas al sistema que pasan a ser ejecutados en modo supervisor (realmente se ejecuta el sistema operativo pero en el nombre del proceso). Además se pueden ejecutar en primer plano o en segundo plano.
- *Procesos demonio.* Son procesos no asociados a ningún usuario que realizan tareas periódicas relacionadas con la administración del sistema, como por ejemplo, la administración y control de redes, y la administración de trabajos de impresión. Los procesos demonio al igual que los procesos de usuario se ejecutan en modo usuario excepto cuando realizan llamadas al sistema que se ejecutan en modo supervisor (realmente se ejecuta el sistema operativo pero en el nombre del proceso). Además se suelen ejecutar en segundo plano.
- *Procesos del sistema operativo,* también denominados simplemente como *procesos del sistema.* Son procesos que realizan tareas de administración del sistema operativo, como por ejemplo, el intercambio de procesos desde memoria principal a memoria secundaria. Se ejecutan normalmente en modo supervisor. Además se suelen ejecutar en segundo plano.

- c) El *planificador a corto plazo* es un componente del sistema operativo que se encarga de decidir que proceso perteneciente a la cola de procesos en el estado preparado será ejecutado a continuación en el procesador.

Desde un punto de vista lógico, todo planificador se puede dividir en tres componentes o elementos:

- *Encolador* (enqueueer). Cuando un proceso entra en el estado preparado, el encolador se encarga de incluir al proceso en la cola de procesos preparados mediante la configuración de los punteros necesarios en el bloque de control del proceso. También puede asignarle una prioridad de ejecución a dicho proceso.
- *Conmutador de contexto* (context switcher). Se encarga de guardar el contexto del proceso que va a ser desalojado del procesador y cargar el contexto del proceso que ha sido planificado para ser ejecutado.
- *Distribuidor o despachador* (dispatcher). Se encarga de seleccionar un proceso de la cola de procesos preparados de acuerdo con un determinado algoritmo de planificación. También se encarga de cederle el control del procesador, para ello el distribuidor debe invocar al conmutador de contexto.

d) El *algoritmo WSClock* es una versión mejorada del algoritmo del reloj para reemplazamiento de páginas.

Para implementar el algoritmo WSClock es necesario que en cada entrada i de la tabla de página de un proceso se mantenga un campo denominado *tiempo de último uso* que contiene el tiempo virtual de último uso de la página i . Cuando se produce un fallo de página se examina el bit referenciada r para saber si la página i fue referenciada en el último intervalo Δ . Si $r = 1$ entonces se escribe en el campo tiempo de último uso el tiempo virtual del proceso. Nótese que la diferencia entre el tiempo virtual actual y el tiempo de último uso, establece la *edad de la página*, es decir, el tiempo que hace que la página no fue referenciada.

Si la edad de la página es menor o igual que Δ entonces la página pertenece al conjunto de trabajo del proceso. Por el contrario, si la edad de la página es mayor que Δ entonces no pertenece al conjunto de trabajo del proceso y es, por lo tanto, una buena candidata para ser seleccionada para ser reemplazada.

El algoritmo WSClock mantiene en una cola circular los números de páginas del conjunto de páginas candidatas y un puntero que apunta a un número de página de la cola. Cuando se produce un fallo de página consulta si la página apuntada por el puntero tiene su bit $r = 1$, es decir, si la página ha sido recientemente referenciada. En ese caso la página no es una buena candidata para ser seleccionada, su bit r se pone a 0 y el puntero apunta a la siguiente página de la lista.

Si una página tiene el su bit $r = 0$, entonces se consulta su bit modificada m . Si $m = 0$ la página no ha sido modificada. En dicho caso se calcula la edad de la página y se compara con la ventana de tiempo Δ . Si la edad es mayor que Δ , la página no pertenece al conjunto de trabajo del proceso y es seleccionada como página víctima para ser reemplazada. Si la edad es menor o igual que Δ , la página pertenece al conjunto de trabajo, por lo que el puntero pasa a apuntar a la siguiente página.

En el caso de que $r = 0$ y $m = 1$, significa que la página no ha sido referenciada recientemente en el último intervalo Δ pero fue modificada en algún intervalo anterior, entonces se planifica para ser escrita en el área de intercambio y el puntero pasa a apuntar a la siguiente página.

En definitiva, el algoritmo WSClock busca páginas con $r = 0$, $m = 0$ y que no pertenezcan al conjunto de trabajo del proceso. Si durante una vuelta completa se da la situación de que todas las páginas pertenecen al conjunto de trabajo del proceso, entonces seleccionará la primera página que encuentre con $r = 0$ y $m = 0$, aunque pertenezca al conjunto de trabajo.

e) El método de asignación de espacio en disco conocido como *método de asignación indexada* consiste en almacenar en un nodo- i los atributos de un archivo y las direcciones físicas de los ocho o diez primeros bloques de un archivo. También se almacenan las direcciones físicas de uno o varios bloques de indirección simple, doble o triple. Un *bloque de indirección simple* es un bloque físico que almacena direcciones físicas de bloques del archivo. Por su parte un *bloque de indirección doble* es un bloque físico que almacena direcciones de bloques de indirección simple.

Mientras que un *bloque de indirección triple* es un bloque físico que almacena direcciones de bloques de indirección doble. El tamaño de un nodo-i suele ser pequeño, unos pocos bytes, por lo que dentro de un bloque físico de discos se pueden almacenar múltiples nodos-i.

Cada archivo tiene asociado un nodo-i que queda identificado mediante un número entero positivo denominado *número de nodo-i*. Al principio de la partición asociada al sistema de archivos se mantiene una lista con todos los nodos-i existentes. El número de nodo-i asociado a un archivo se almacena en la entrada del directorio que contiene el archivo. Cuando se abre un archivo, su nodo-i se lee en la lista de nodos-i y se carga en memoria principal para poder conocer las direcciones físicas de los bloques del archivo y sus atributos.

Las ventajas de este método son:

- Se puede usar tanto para archivos de acceso secuencial como para archivos de acceso aleatorio.
- No produce fragmentación externa.
- Permite que el espacio de los bloques físicos que contienen datos del archivo pueda ser utilizado en su totalidad.
- Su implementación requiere el uso de menos espacio en memoria principal que el método de asignación enlazada con FAT

La principal desventaja del método de asignación indexada es que el tiempo de acceso a un bloque del archivo depende de la estructura interna del nodo-i, del tamaño del archivo y de la posición del bloque al que se desea acceder.

Solución Ejercicio 2

En la Figura 1 se propone una solución que utiliza las siguientes variables globales y semáforos binarios:

- contador. Variable global de tipo entero para llevar la cuenta del número de ciudadanos en la cola o en la ventanilla.
- S1. Semáforo binario que se utiliza para garantizar la exclusión mutua en el uso de la variable global contador.
- S2. Semáforo binario que se utiliza para sincronizar el acceso a las ventanillas de la oficina.

```
int contador=0; /* Definición e inicialización de la variable global */
semáforo_binario S1, S2; /* Definición semáforos binarios */
void ciudadano() /* Proceso ciudadano */
{
    wait_sem(S2); /* Esperar si todas las ventanillas están ocupadas */

    wait_sem(S1);
    contador = contador + 1;
    if (contador < 3) signal_sem(S2); /* Existen ventanillas libres */
    signal_sem(S1);

    realizar_gestión();

    wait_sem(S1);
    contador = contador - 1;
    if (contador == 2) signal_sem(S2); /* Queda una ventanilla libre */
    signal_sem(S1);
}

main() /* Inicialización semáforos y ejecución concurrente */
{
    init_sem(S1,1);
    init_sem(S2,1);
    ejecución_concurrente(ciudadano, ...,ciudadano);
}
```

Figura 1 – Solución del apartado a) del Problema 4.1

Solución Ejercicio 3

NOTA: Aunque en el enunciado se indica que hay que usar el *algoritmo del banquero* (ver páginas 179 y 180 de la bibliografía básica) realmente no es necesaria su utilización en su totalidad ya que simplemente hay que limitarse a comprobar si los estados indicados son seguros, es decir, se estaría únicamente en el paso 1 del algoritmo.

a) Del enunciado se deduce que el estado en que se encuentra el sistema es el siguiente:

$$S_1 = \left\{ \mathbf{N} = \begin{pmatrix} 5 \\ 6 \\ 4 \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} \quad \mathbf{R}_E = 10 \quad \mathbf{R}_D = 4 \right\}$$

El número de instancias del recurso que todavía necesita cada proceso se obtiene restando \mathbf{N} y \mathbf{A} :

$$\mathbf{N} - \mathbf{A} = \begin{pmatrix} 5 \\ 6 \\ 4 \end{pmatrix} - \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 5 \\ 1 \end{pmatrix}$$

Para saber si este estado es seguro simplemente hay que comprobar si es posible completar la ejecución de todos los procesos. Para ello en primer lugar hay que comprobar si existe alguna fila i de $\mathbf{N} - \mathbf{A}$, es decir, algún proceso P_i que cumpla la condición

$$(\mathbf{N}_i - \mathbf{A}_i) \leq \mathbf{R}_D \quad i = 1, 2, 3$$

En este caso, la condición toma la forma

$$\begin{pmatrix} 3 \\ 5 \\ 1 \end{pmatrix} \leq 4$$

Se observa que la primera fila asociada al proceso A si que cumple la condición. Luego al proceso A se le pueden conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez.

Supóngase que el proceso A se ha completado, la condición pasaría a ser:

$$\begin{pmatrix} 0 \\ 5 \\ 1 \end{pmatrix} \leq 6$$

Se observa que la segunda fila asociada al proceso B si que cumple la condición. Luego se le pueden conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez.

Supóngase que el proceso B se ha completado, la condición pasaría a ser:

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \leq 7$$

Finalmente se observa que la última fila asociada al proceso C. Luego se le pueden conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez.

En conclusión, como se ha podido completar la ejecución de los tres procesos, el estado S_1 es **seguro**.

b) Del enunciado se deduce que el estado en que se encuentra el sistema es el siguiente:

$$S_2 = \left\{ \mathbf{N} = \begin{pmatrix} 5 \\ 6 \\ 4 \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix} \quad \mathbf{R}_E = 10 \quad \mathbf{R}_D = 1 \right\}$$

El número de instancias del recurso que todavía necesita cada proceso se obtiene restando \mathbf{N} y \mathbf{A} :

$$\mathbf{N} - \mathbf{A} = \begin{pmatrix} 5 \\ 6 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$$

Para saber si este estado es seguro simplemente hay que comprobar si es posible completar la ejecución de todos los procesos. Para ello en primer lugar hay que comprobar si existe alguna fila i de $\mathbf{N} - \mathbf{A}$, es decir, algún proceso P_i que cumpla la condición

$$(\mathbf{N}_i - \mathbf{A}_i) \leq \mathbf{R}_D \quad i = 1, 2, 3$$

En este caso, la condición toma la forma

$$\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \leq 1$$

Se observa que ningún proceso cumple la condición, luego el estado S_2 es **inseguro**.

Solución Ejercicio 4

- a) Cuando se utiliza la técnica de paginación una dirección física se descompone en los campos número de marco de página de f bits y desplazamiento dentro del marco de d bits. Por su parte una dirección lógica se descompone en los campos número de página de p bits y desplazamiento dentro de la página de d bits.

- *Dirección Física:*

Para obtener el tamaño del campo número de marco de página se va a calcular en primer lugar el número de marcos N_{MP} en que se descompone la memoria principal si se considera un tamaño de página $S_P = 4 \text{ KiB}$:

$$N_{MP} = \text{floor} \left(\frac{C_{MP}}{S_P} \right) = \text{floor} \left(\frac{16 \text{ GiB}}{4 \text{ KiB}} \right) = \text{floor} \left(\frac{2^{34}}{2^{12}} \right) = 2^{22} \text{ marcos}$$

Conocido N_{MP} el tamaño de este campo se obtiene resolviendo la siguiente desigualdad:

$$\min_f \{ N_{MP} \leq 2^f \}$$

Luego como $N_{MP} = 2^{22}$ entonces $f = 22$ bits.

Para obtener el tamaño del campo desplazamiento, puesto que la unidad direccionable es un byte, hay que resolver la siguiente desigualdad:

$$\min_d \{ S_P \leq 2^d \}$$

Luego como $S_P = 2^{12}$ entonces $d = 12$ bits.

Marco j	Desplazamiento d
$f = 22$ bits	$d = 12$ bits

- *Dirección lógica:*

El tamaño del campo desplazamiento de una dirección lógica es igual al de una dirección física, en este caso, $d = 12$ bits. Por otra parte, en el enunciado se indica que una dirección lógica tiene un tamaño de 48 bits, luego la suma del tamaño p del campo número de página y del tamaño d del campo desplazamiento debe ser igual a 48 bits:

$$p + d = 48$$

Luego $p = 48 - d = 48 - 12 = 36$ bits.

Página i	Desplazamiento d
$p = 36$ bits	$d = 12$ bits

- b) La unidad direccionable es un byte y se suelen usar 48 bits para las direcciones lógicas. En ese caso, el espacio de direcciones lógicas de un proceso puede tener un tamaño máximo de

$$C_X = 2^{48} \text{ B} = 256 \text{ TiB}$$

Supuesto un proceso de este tamaño, si el tamaño de página es $S_P = 4 \text{ KiB} = 2^{12} \text{ B}$, entonces el número de páginas N_P en que se divide el espacio de direcciones lógicas de este proceso es:

$$N_P = \text{ceil} \left(\frac{C_X}{S_P} \right) = \text{ceil} \left(\frac{2^{48}}{2^{12}} \right) = \frac{2^{48}}{2^{12}} = 2^{36} \text{ páginas}$$

- c) La tabla de páginas de este proceso tiene una entrada por cada página, es decir, 2^{36} entradas. Como cada entrada de la tabla de páginas ocupa $S_E = 8$ bytes, entonces la tabla de páginas completa tendría un tamaño de

$$C_{TP} = S_E \times N_P = 8 \times 2^{36} = 2^{39} \text{ B} = 512 \text{ GiB}$$

Como la memoria principal tiene un tamaño de $C_{MP} = 16 \text{ GiB}$, entonces la tabla de páginas del proceso no podría almacenarse en la memoria principal.