

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

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

- I) (1 p) ¿Qué información contiene un marco de la *región de pila* del espacio de direcciones de *memoria lógica* de un proceso?
  - II) (1 p) ¿Qué función realiza el *planificador a corto plazo* de un sistema operativo? ¿Cuáles son sus componentes?
  - III) (1 p) Explicar la *gestión de la entrada de un ratón*.
  - IV) (1 p) Enumerar y describir brevemente las diferentes *áreas* que se distinguen de forma general en la *estructura de un sistema de archivos*.
2. (2 p) Enumerar y describir brevemente las *estructuras de datos* que utiliza el sistema operativo para implementar la *paginación*.

3. (2 p) Supóngase un sistema con cinco procesos (P0, P1, P2, P3 y P4) y tres recursos diferentes (R0, R1 y R2). En un instante determinado el sistema se encuentra en el siguiente estado:

Proceso	Asignados			Máximos necesarios			Disponibles		
	R0	R1	R2	R0	R1	R2	R0	R1	R2
P0	1	0	0	4	4	4	3	3	2
P1	0	1	2	1	4	6			
P2	0	0	3	3	2	5			
P3	1	0	3	4	4	8			
P4	1	1	0	5	1	10			

Determinar si el estado es seguro.

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

4. En la Figura 1 se muestra un **fragmento** del pseudocódigo en C de un cierto programa que hace uso de un semáforo S1 inicializado a 1 para implementar la exclusión mutua en el uso de una variable compartida contador.

- (1.5 p) Explicar **razonadamente** si el pseudocódigo es correcto o por el contrario contiene algún error. En este último caso indique como se corregiría dicho error.
- (0.5 p) ¿Qué sucedería con este pseudocódigo si el semáforo se hubiera inicializado con el valor 0?

```
wait_sem(S1);
contador = contador + 1;
signal_sem(S1);
if (contador > N)
{
    enviar();
    x=x/2;
}
```

Figura 1

## SISTEMAS OPERATIVOS (Cód. 71902048)

### Solución Examen Septiembre 2023

#### Solución Ejercicio 1

- I) El *espacio de direcciones de memoria lógica o virtual de un proceso* se divide en varias *regiones*, cada una de las cuales delimita un área de direcciones contiguas de memoria lógica. De forma general se suelen distinguir al menos tres regiones: la región de código, la región de datos y la región de pila.

La *región de pila* se crea automáticamente y su tamaño es ajustado dinámicamente en tiempo de ejecución por el sistema operativo. La pila está constituida por marcos de pila lógicos. Un marco se añade a la pila cuando se llama a una función y se extrae cuando se vuelve de la misma.

Un *marco de pila* contiene usualmente la siguiente información: los parámetros de la función, sus variables locales y las direcciones almacenadas en diferentes registros especiales de la máquina, como por ejemplo, el contador del programa y el puntero de la pila. Salvar el contenido del contador del programa permite conocer la dirección de retorno donde debe continuar la ejecución una vez que se ha ejecutado la función. Mientras que salvar el contenido del registro de pila permite conocer la ubicación del marco de pila anterior o del siguiente libre.

- II) El *planificador a corto plazo*, más conocido como *planificador del procesador* o simplemente *planificador (scheduler)* es el componente del sistema operativo que se encarga de seleccionar al proceso perteneciente a la cola de procesos en el estado preparado que pasará al estado ejecutándose, es decir, que 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.

- III) Un *ratón* es un dispositivo modo carácter de entrada. Cada vez que el usuario desplaza el ratón o pulsa un botón del mismo se transmite un mensaje al registro del controlador de E/S que lo supervisa. El mensaje suele tener un tamaño de unos 3 bytes y contiene la siguiente información: desplazamiento horizontal (eje  $x$ ), desplazamiento vertical (eje  $y$ ) y estado (pulsado o sin pulsar) de los diferentes botones del ratón. Debe tenerse en cuenta que los desplazamientos son relativos, es decir, se miden en relación a la posición del ratón indicada en el último mensaje enviado.

Cada vez que recibe un mensaje, el controlador de E/S que supervisa al ratón genera una interrupción. El manejador de la interrupción despierta el driver del ratón que lee el mensaje y lo coloca en una cola de mensajes para que sea leído y procesado por el proceso adecuado.

- IV) La estructura de un sistema de archivos depende de cada tipo de sistema de archivos en particular, pero de forma general se suelen distinguir en la misma las siguientes áreas:

- *Bloque de arranque*. Se sitúa al comienzo de la partición y puede contener el código necesario para arrancar un sistema operativo. La tabla de particiones indica la *partición activa*, es decir, aquella que contiene el código de arranque del sistema operativo que se quiera cargar al arrancar el computador.
- *Estructura de datos con metadatos del sistema de archivos*. Contiene información administrativa y estadística del sistema de archivos, como por ejemplo: un identificador del tipo de sistema de archivos, el número de bloques del sistema de archivos, el número de bloques libres, etc. Algunos sistemas operativos, como por ejemplo Linux, denominan a esta estructura *superbloque*.
- *Estructura de datos con información sobre los bloques libres en el sistema de archivos*. Generalmente denominada como *lista de bloques libres*. Puede implementarse como un mapa de bits o como una lista enlazada.
- *Estructura de datos para localizar los bloques asignados a los archivos*. Como por ejemplo una *lista de nodos índice* o una *tabla de asignación de archivos* o FAT (*File Allocation Table*).
- *Área de datos*. Contiene los bloques libres y los bloques de datos asignados a los archivos y a los directorios.

## Solución Ejercicio 2

Un sistema operativo para implementar la paginación generalmente utiliza las siguientes estructuras de datos:

- *Tablas de páginas*. Cada proceso tiene asignada una tabla de páginas. Cada entrada  $i$  de la tabla de páginas de un proceso  $X$  contiene, entre otras informaciones, el marco  $j$  donde se encuentra almacena la página  $i$  del proceso  $X$  y los permisos de acceso a la página. Además puede contener información sobre la ubicación de la copia de la página en memoria secundaria, aunque a veces esta información se implementa en una tabla diferente.
- *Tabla de marcos de página*. Esta tabla tiene tantas entradas como marcos de página tiene la memoria principal. Cada entrada  $j$  de la tabla de marcos de página contiene, entre otras, las siguientes informaciones relativas al marco: su estado, es decir, si está libre o ocupado, punteros para crear una lista de marcos libres, y la ubicación en memoria secundaria de la copia de la página  $i$  contenida en el marco.
- *Lista de marcos libres*. Es consultada cuando hay que asignar espacio a los procesos que deben ser cargados en memoria principal. Cuando hay que asignar un marco se selecciona al primero de la lista. Si un marco queda libre es colocado al final de la lista.

**Solución Ejercicio 3**

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

$$S = \left\{ \mathbf{N} = \begin{pmatrix} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 3 & 2 & 5 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix} \quad \mathbf{R}_E = (6 \ 5 \ 10) \quad \mathbf{R}_D = (3 \ 3 \ 2) \right\}$$

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  $P_i$   $i = 1, \dots, 5$  que cumpla la condición

$$(\mathbf{N}_i - \mathbf{A}_i) \stackrel{e}{\leq} \mathbf{R}_D$$

donde el símbolo  $\stackrel{e}{\leq}$  indica que la comparación de cada fila se debe realizar elemento a elemento, es decir, se compara el elemento  $N_{ij} - A_{ij}$  de  $\mathbf{N}_i - \mathbf{A}_i$  con el elemento  $R_{Dj}$  del vector  $\mathbf{R}_D$  siendo  $j = 1, 2, 3$ .

En este caso, la condición toma la forma

$$\begin{pmatrix} 3 & 4 & 4 \\ 1 & 3 & 4 \\ 3 & 2 & 2 \\ 3 & 4 & 5 \\ 4 & 0 & 10 \end{pmatrix} \stackrel{e}{\leq} (3 \ 3 \ 2)$$

Se observa que solo la tercera fila, asociada al proceso P2, cumple la condición. Luego al proceso P2 se le pueden conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez.

Supóngase que el proceso P2 se ha completado, con lo que los recursos que tenía asignados quedan libres y se añaden al vector de recursos disponibles. La condición pasaría a ser:

$$\begin{pmatrix} 3 & 4 & 4 \\ 1 & 3 & 4 \\ 0 & 0 & 0 \\ 3 & 4 & 5 \\ 4 & 0 & 10 \end{pmatrix} \stackrel{e}{\leq} (3 \ 3 \ 5)$$

Se observa que la segunda fila, asociada al proceso P1, cumple la condición. Luego al proceso P1 se le pueden conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez.

Supóngase que el proceso P1 se ha completado, la condición pasaría a ser:

$$\begin{pmatrix} 3 & 4 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 3 & 4 & 5 \\ 4 & 0 & 10 \end{pmatrix} \stackrel{e}{\leq} (3 \ 4 \ 7)$$

Se observa que la primera fila, asociada al proceso P0, cumple la condición. Luego al proceso P0 se le pueden conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez.

Supóngase que el proceso P0 se ha completado, la condición pasaría a ser:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 3 & 4 & 5 \\ 4 & 0 & 10 \end{pmatrix} \leq^e (4 \ 4 \ 7)$$

Se observa que la cuarta fila, asociada al proceso P3, cumple la condición. Luego al proceso P3 se le pueden conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez.

Supóngase que el proceso P3 se ha completado, la condición pasaría a ser:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 4 & 0 & 10 \end{pmatrix} \leq^e (5 \ 4 \ 10)$$

Se observa que la quinta fila, asociada al proceso P4, cumple la condición. Luego al proceso P4 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 cinco procesos, el estado S es **seguro**.

#### Solución Ejercicio 4

- a) El error que presenta este pseudocódigo es que la variable compartida `contador` no se encuentra protegida por el semáforo para garantizar su uso con exclusión mutua cuando se consulta su valor en la sentencia `if`. En la Figura 1 se muestra el pseudocódigo corregido. Se observa que la operación `signal_sem(S1)` que establece el fin de la zona con exclusión mutua, es decir, de la sección crítica, se coloca después del `if`, es decir, cuando ya se ha terminado de usar la variable compartida `contador`. También se ha añadido la sentencia `else signal_sem(S1)` para que la operación se aplique también en el caso de que no se cumpla la condición del `if`, es decir, que `contador` sea menor o igual que N.

```
wait_sem(S1);
contador = contador + 1;
if (contador > N)
{
    signal_sem(S1);
    enviar();
    x=x/2;
}
else signal_sem(S1);
```

**Figura 1**

- b) Si el semáforo S1 se hubiera inicializado a 0 en lugar de a 1, entonces todos los procesos que ejecutasen la operación `wait_sem(S1)` se quedarían bloqueados y no podrían avanzar en su ejecución.