

Si comparamos desde el punto de vista de la **eficiencia** la reserva de disco (también conocida como caché de disco) de escritura diferida con reserva de disco de escritura directa,...

-

La reserva de **escritura diferida** es **más eficiente** que la reserva de escritura directa, pero **sólo en las escrituras**

2.

Las entidades bancarias lanzan a primero de mes cientos o incluso miles de procesos desde sus oficinas que realizan las transferencias bancarias programadas. Para evitar que se pueda dar un interbloqueo entre algunos de esos procesos, un programador propone que **todos** los procesos utilicen el siguiente protocolo para realizar las transferencias:

```
while(hayan_transferias_por_procesar())
{
    // Obtenemos los números de las cuentas y la
    cantidad a transferir...
    obtener_datos_transferencia(&cuenta_deudora, &cuenta_acreadora, &can
    tidad);
    //Comparamos entre sí los números de
    cuenta...
    if (cuenta_deudora < cuenta_acreadora)
    {
        bloquear(cuenta_deudora);
        bloquear(cuenta_acreadora);
    }
    else
    {
        bloquear(cuenta_acreadora);
        bloquear(cuenta_deudora);
    }

    transferir(cuenta_deudora, cuenta_acreadora, cantidad);
    desbloquear(cuenta_deudora);
    desbloquear(cuenta_acreadora);
}
```

La solución propuesta funciona perfectamente sin riesgo de interbloqueo.

En un sistema en el que hay cuatro procesos P1..P4 compartiendo recursos se utiliza el método de matrices binarias de relación para la detección de interbloqueos. En un momento dado tras una petición los cuatro procesos se cuelgan, y el cierre transitivo que se obtiene es el siguiente:

	P1	P2	P3	P4
P1	0	1	1	1
P2	0	0	1	1
P3	0	0	1	1
P4	0	0	1	1

- Se libera P1, pero el interbloqueo entre P3 y P4 se mantiene

4. ¿Cuándo se deben ejecutar los algoritmos de **predicción** de interbloqueo?

-

Cuando se solicita un recurso que está libre

5 La fragmentación del espacio libre de memoria en procesadores que soportan paginación...

-no constituye problema alguno

6-En un sistema basado en microcontrolador la memoria se asigna en unidades de asignación (*ticks*) de 16 bytes, y la gestión de la memoria se efectúa mediante un mapa de bits, encontrándose en un momento el mapa en el estado que se muestra a continuación, y comenzando la memoria gestionada en la dirección 4000H. Si la asignación de memoria se realiza por el método del **peor ajuste**, ¿qué dirección se asignaría si se solicitase un bloque de 90 bytes?

NOTAS: Por simplicidad, las direcciones están en hexadecimal, ya que 16D=10H, 32D=20H, 48D=30H, etc.

-4100H

7-En un sistema basado en microcontrolador la memoria se asigna en unidades de asignación (*ticks*) de 16 bytes, y la gestión de la memoria se efectúa mediante un mapa de bits, encontrándose en un momento el mapa en el estado que se muestra a continuación, y comenzando la memoria gestionada en la dirección 4000H. Si la asignación de memoria se realiza por el método del **mejor ajuste**, ¿qué dirección se asignaría si se solicitase un bloque de 90 bytes?

NOTAS: Por simplicidad, las direcciones están en hexadecimal, ya que 16D=10H, 32D=20H, 48D=30H, etc.

-41B0H

8-Ante la no disponibilidad de otros mecanismos de coordinación, un programador decide utilizar un cerrojo para asegurar la exclusión mutua en el uso de un recurso. Como recuerda el problema de la atomicidad que le explicaron en clase de sistemas operativos, decide deshabilitar las interrupciones para impedir la conmutación durante la comprobación del cerrojo, escribiendo el siguiente código (el valor inicial de cerrojo es ABIERTO, indicando que el recurso está inicialmente libre):

```
deshabilitar_interrupciones();
while (cerrojo==CERRADO) ;
cerrojo=CERRADO;
habilitar_interrupciones();
    [sección crítica]
cerrojo=ABIERTO;
```

¿Qué cree que ocurrirá?

- Si un primer proceso encuentra el recurso libre conseguirá entrar en la sección crítica, pero si estando aún en la sección crítica se produce una conmutación y otro proceso que necesita usar el recurso ejecuta el protocolo de entrada, se colgará no solo dicho procesos, sino también el sistema al completo.

9-En un sistema varios procesos reciben mensajes a través de un mismo buzón B. Dado que estos procesos además de recibir y procesar los mensajes deben realizar otras actividades, no deberían bloquearse al intentar leer del buzón si este se encuentra vacío. Para ello, los procesos utilizan una primitiva que permite consultar el número de mensajes que hay en el buzón, y realizan siempre la siguiente comprobación antes de leer un mensaje:...

```
if (B.numero_de_mensajes() > 0)
{
    Mensaje msg= B.receive();
    procesar_mensaje (msg);
}
else
{
    // hacer lo que quiera que se haga si no hay mensaje que procesar
    ...
}
```

¿Se puede garantizar que nunca se bloqueará un proceso por encontrarse vacío el buzón al ejecutar "Mensaje msg= B.receive();" ?

-No

10- ¿Qué ocurre si durante una fase de ejecución un proceso tiene más marcos que páginas hay en su conjunto de trabajo? ¿Y si tiene menos?

-

Si tiene más marcos que páginas hay en el conjunto de trabajo, el proceso se ejecutará sin producir fallos de página una vez que haya cargado el conjunto de trabajo, aunque habrán marcos que se quedan sin usar. Si tiene menos marcos que páginas hay en el conjunto de trabajo se puede producir hiperpaginación.

11- Ejecutamos un proceso que sabemos que hará **A** accesos a **P** páginas diferentes durante su ejecución. Si dicho proceso lo ejecutamos con **M** marcos, ¿cuántos fallos de página provocará como máximo? ¿Y como mínimo?

- Como máximo producirá **A** fallos de página, y como mínimo, **P** fallos de página.

12- ¿Con cuál de las siguientes afirmaciones sobre el uso de monitores en la práctica está de acuerdo?

- Cuando un hilo está ejecutando un método de un monitor tiene garantizada la exclusión mutua sobre el monitor.

13- En un motor de bases de datos, los procesos pueden solicitar la apertura de archivos en exclusividad. Cuando se intenta abrir un archivo en exclusividad, si el archivo está libre, se realiza la apertura exclusiva. Pero si el archivo ya está bloqueado por otro proceso, la apertura devuelve un código de error, y además todos los archivos abiertos en exclusividad hasta el momento se cierran y es necesario volver a abrirlos.

En este sistema, ¿qué condiciones de Coffman no se cumplen?

+ Ni retención y espera, ni no apropiación

14- Un programador construye una herramienta de depurado para sus aplicaciones concurrentes que le muestra en pantalla el grafo de relación entre procesos y recursos. En un momento dado, todos sus procesos se le cuelgan, con lo que usa su herramienta para visualizar el grafo, y lo que ve es esto:

- Desbloquear P1

15- En un sistema industrial múltiples procesos comparten múltiples dispositivos físicos de forma que para determinadas actividades necesitan la asignación en exclusividad de varios dispositivos simultáneamente. Para cada actividad, los procesos deben construir una lista con los recursos que necesitan y hacer una llamada a un gestor de recursos pasándole dicha lista, y el gestor los asigna todos a la vez cuando están todos disponibles. Una vez utilizado cada dispositivo, los procesos los van devolviendo conforme los van dejando de necesitar.

En este sistema... ¿qué condición de Coffman no se cumple?

- Retención y espera

16- Supongamos un sistema con memoria paginada en el que las direcciones de memoria son de 16 bits, siendo los 4 bits de mayor peso el número lógico de página (NLP). La traducción a la dirección física se hace mediante una tabla de páginas en la unidad de gestión de memoria. En dicha tabla se tiene para cada página, además del número de marco, un bit P que indica si la página está presente (P=1) o ausente (P=0).

A la vista de la tabla de página, establezca la asociación entre cada página y su traducción (o fallo de página, si procede)

NLP	Marco	P	NLP	Marco	P	NLP	Marco	P	NLP	Marco	P
-----	-------	---	-----	-------	---	-----	-------	---	-----	-------	---

0	0	1	4	1	1	8	F	0	C	F	0
1	2	1	5	3	1	9	F	0	D	F	0
2	4	1	6	5	1	A	F	0	E	F	0
3	6	1	7	6	1	B	F	0	F	F	0

**** comentario :** No. Debe usar el dígito superior de la dirección para indexar la tabla de página, y de ahí obtener el número de marco, siempre y cuando el bit de presente sea P=1. Si el bit de presente es P=0, eso significa que la dirección produce un fallo de página.

17- En un sistema basado en microcontrolador la memoria se gestiona mediante lista de control. Considere los nodos de la lista que se muestran a continuación, sabiendo que los nodos que siguen representan **bloques ocupados**. Por simplicidad, la memoria se asigna en *ticks* de 16 bytes, por lo que el tamaño de cada bloque está expresado en *ticks* de dicho tamaño. En esta situación...

- Un proceso solicita 90 bytes de memoria.
- A continuación, el proceso que tiene asignado el bloque de memoria de la dirección 180H lo libera.

Sabiendo que el criterio de asignación que se utiliza es **siguiente ajuste**, y que el bloque que comienza en la dirección 60H fue el último que se asignó, ¿Cómo quedaría la lista tras ambas operaciones?

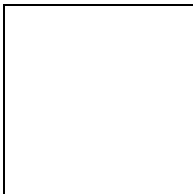
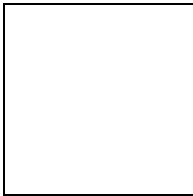
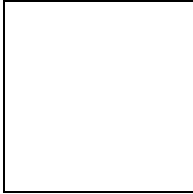
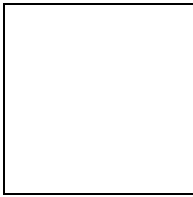
NOTA: Por simplicidad, las direcciones están en hexadecimal, ya que 16D=10H, 32D=20H, 48D=30H, etc.

****Comentario** Pues me temo que no. La asignación por siguiente ajuste realiza la asignación localizando el siguiente bloque, empezando a buscar desde la última asignación que se hizo, con el que se puede satisfacer la petición actual. En nuestro caso, en que necesitamos $90/16=5, x=6$ ticks de memoria. Dado que el último bloque que se asignó es el bloque que empieza en la dirección 60, el bloque con el que podemos satisfacer la petición es justo el siguiente, el que comienza en la dirección C0H. Dicho bloque se divide un bloque ocupado de 6 ticks y un residuo libre de otros 6 ticks, el cual, cuando se libera el bloque ocupado que va justamente detrás (dirección 180H) se fusiona con dicho bloque, y con el siguiente bloque libre, el que empieza en la dirección 1C0H. Los bloques ocupados en ningún caso se fusionan.

18- El algoritmo de asignación de memoria por mejor ajuste tiene como propósito...

- ... intentar minimizar la fragmentación del espacio libre de memoria

- Sistema de sustitución de paginas



18 Las entidades bancarias lanzan a primero de mes cientos o incluso miles de procesos desde sus oficinas que realizan las transferencias bancarias programadas. Para evitar que se pueda dar un interbloqueo entre algunos de esos procesos, un programador propone que **todos** los procesos utilicen el siguiente protocolo para realizar las transferencias:

```
while(hayan_transferias_por_procesar())
{
    // Obtenemos los números de las cuentas y la
    cantidad a transferir...
    obtener_datos_transferencia(&cuenta_deudora, &cuenta_
    acreedora, &cantidad);
    //Comparamos entre sí los números de
    cuenta...
```

```

if (cuenta_deudora < cuenta_acreadora)
{
    bloquear(cuenta_deudora);
    bloquear(cuenta_acreadora);
}
else
{
    bloquear(cuenta_acreadora);
    bloquear(cuenta_deudora);
}

transferir(cuenta_deudora, cuenta_acreadora, cantidad);
desbloquear(cuenta_deudora);
desbloquear(cuenta_acreadora);
}

```

¿Con cuál de las siguientes afirmaciones está de acuerdo?

La solución propuesta funciona perfectamente sin riesgo de interbloqueo.

19 -En un sistema se ejecutan cuatro procesos que comparten cuatro tipos de recursos. En un instante dado, la matriz A representa los recursos de cada tipo asignados a cada proceso, el vector D representa los recursos de cada tipo disponibles y la matriz M representa las necesidades máximas de recursos por parte de cada proceso. En esta situación, el proceso P4 solicita dos instancias del recurso R4. Aplicando el algoritmo del banquero, ¿se debería satisfacer dicha petición?

M=

	R1	R2	R3	R4
P1	2	2	1	2
P2	0	2	1	0
P3	2	2	1	0
P4	2	0	1	2

A=

	R 1	R 2	R 3	R 4
P 1	2	0	0	0
P 2	0	0	1	0
P 3	0	2	0	0
P 4	0	0	1	0

D=

R 1	R 2	R 3	R 4
0	2	0	2

--- No, puesto que el estado al que lleva no es seguro. No existe garantía de que se puedan satisfacer las peticiones futuras de P1 ni P4.

20-

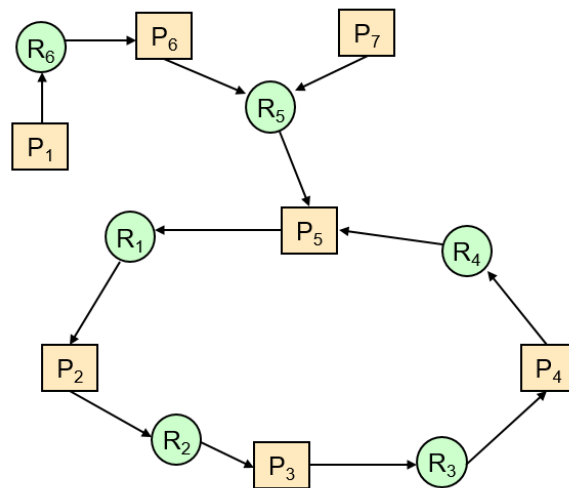
Si un proceso realiza un petición de un recurso que está disponible, y aplicando el algoritmo del banquero llegamos a la conclusión de que atender dicha petición lleva el sistema a un estado **no seguro**, eso quiere decir que...

-No se debe atender la petición porque si lo hacemos nos lleva a un estado a partir del cual se puede terminar llegando a un interbloqueo en el futuro.

21- ¿Cuándo se deben ejecutar los algoritmos de **detección** de interbloqueo?

-Cuando se solicita un recurso que está ocupado

22-Un programador construye una herramienta de depurado para sus aplicaciones concurrentes que le muestra en pantalla el grafo de relación entre procesos y recursos. En un momento dado, todos sus procesos se le cuelgan, con lo que usa su herramienta para



visualizar el grafo, y lo que ve es esto:



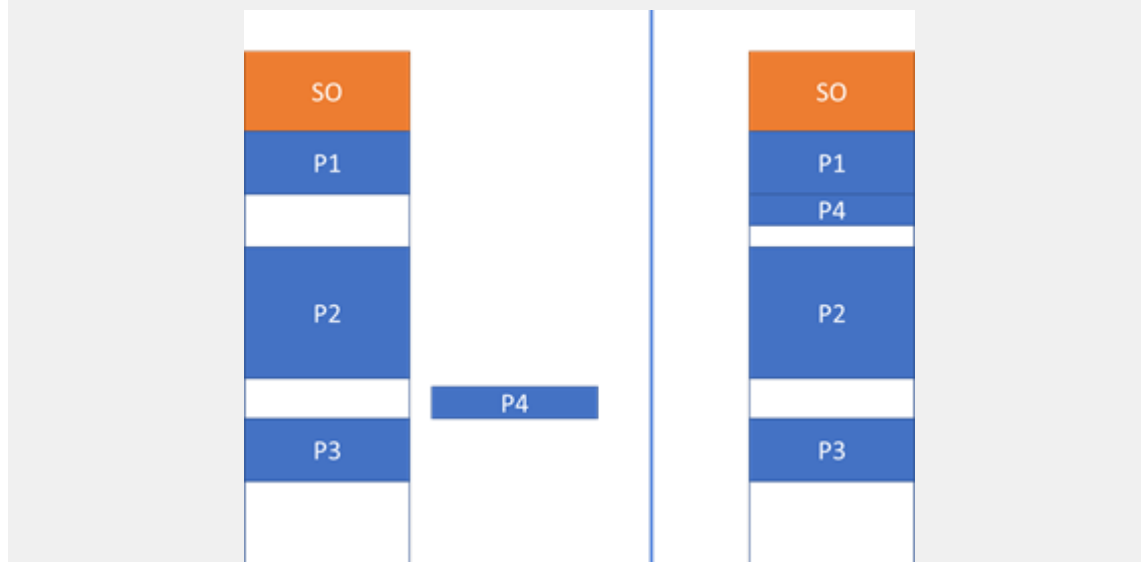
Desbloquear P1

23-En un sistema que use la aproximación discreta a LRU como criterio de sustitución de páginas, ¿cuál de las siguientes páginas sería la candidata a ser sustituida? -0

	Marco	R	...	
0	8	0	...	3
1	1204	1	...	0
2	12	0	...	1
3	32	0	...	2

24-

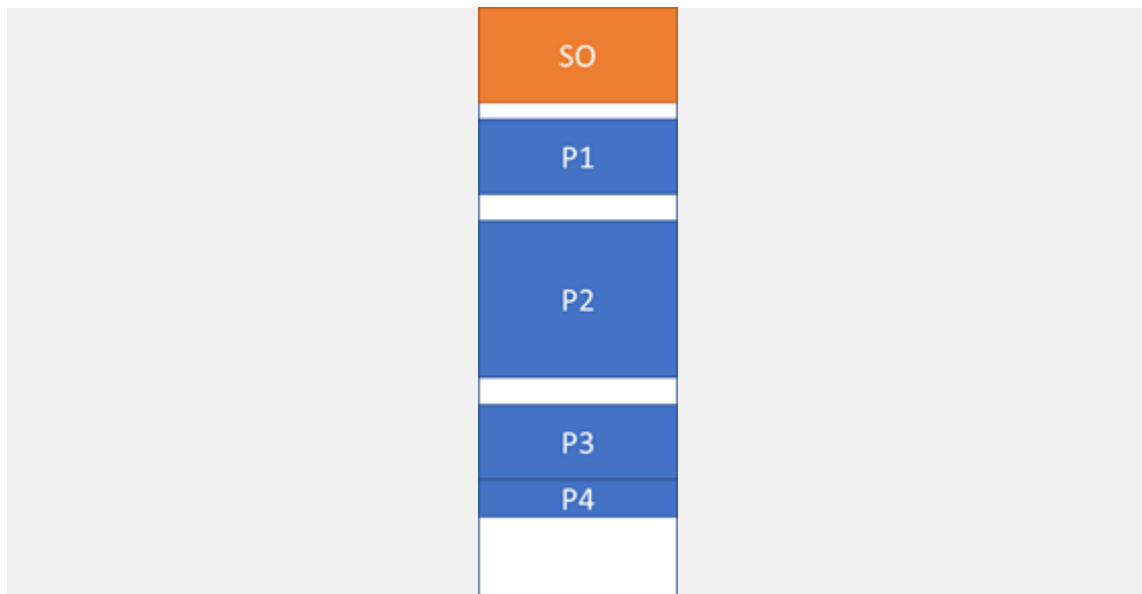
Dada la siguiente situación, ¿qué criterio se ha aplicado?



-PRIMER AJUSTE

25

Dada la siguiente representación de los procesos distribuidos en memoria, ¿con cuál de los siguientes mapas de bit se corresponde?



-000...011...100...011...100...011...100

-26 Si hablamos de la gestión de memoria a nivel de procesos, ¿qué ventaja tienen las listas de control frente a los mapas de bits?

-Dado un nuevo proceso de tamaño T, es más fácil buscarle un hueco libre en memoria

-27 ¿Cuál de las siguientes afirmaciones sobre los procesos no es correcta?

-Usan un espacio de memoria fijo preestablecido asignado por el sistema

-28 Un semáforo s tiene valor inicial 3 en su contador. Si un proceso ejecuta el siguiente bucle sin que ningún otro proceso utilice el semáforo s, ¿cuántas veces se mostraría el texto "¡estoy harto de estar en casa!"?

```
while(true)
{
    down(s);
    printf("¡estoy harto de estar en casa!\n");
    up(s);
}
```

-Nunca se bloquearía en el down(s), con lo que lo repetiría indefinidamente.

-29 La fragmentación del espacio libre de memoria en procesadores que soportan paginación...

-... no constituye problema alguno

30-En un sistema basado en microcontrolador la memoria se gestiona mediante lista de control. Considere los nodos de la lista que se muestran a continuación, sabiendo que los nodos que siguen representan **bloques ocupados**. Por simplicidad, la memoria se asigna en *ticks* de 16 bytes, por lo que el tamaño de cada bloque está expresado en *ticks* de dicho tamaño. En esta situación...

a. Un proceso solicita 110 bytes de memoria.

- b. A continuación, el proceso que tiene asignado el bloque de memoria de la dirección 240H lo libera.

Sabiendo que el criterio de asignación que se utiliza es **mejor ajuste**, ¿Cómo quedaría la lista tras ambas operaciones?

NOTA: Por simplicidad, las direcciones están en hexadecimal, ya que 16D=10H, 32D=20H, 48D=30H, etc.



31- El algoritmo de asignación de memoria por mejor ajuste tiene como propósito...

-... intentar minimizar la fragmentación del espacio libre de memoria

32-

En un sistema se utiliza sustitución por envejecimiento, asociándose a cada página una variable de **16 bits** que se actualiza **cada milisegundo**. En un momento dado, hay una página que tiene su variable asociada a cero. ¿Quiere eso decir que esa página no se ha usado nunca? En caso contrario, ¿cuánto tiempo hace como mínimo que no se usa?

-

La página sí se ha utilizado, y hace como mínimo 16 milisegundos que no se utiliza.

33- En un sistema en el que la sustitución de páginas se hace de forma local y sin ningún mecanismo de redistribución se va a ejecutar un proceso del que se sabe que tiene los siguientes números de páginas en los conjuntos de trabajo de sus fases de ejecución:

32, 64, 1024, 64, 16, 512, 64

¿Cuántos marcos deberíamos asignarle a dicho proceso para garantizar que no habrá hiperpaginación en ninguna fase de ejecución?

- 1024

34- Si un proceso realiza un petición de un recurso que está disponible, y aplicando el algoritmo del banquero llegamos a la conclusión de que atender dicha petición lleva el sistema a un estado no seguro, eso quiere decir que...

-. No se debe atender la petición porque si lo hacemos nos lleva a un estado que nos puede terminar conduciendo a un interbloqueo más adelante.

Revisar entrega de examen: Exa... X

Archivo C:\Users\PC\Desktop\examen%20%20segunda%20conv%20%20Revisar%20entrega%20de%20examen,%20Examen%20segunda%20Convocatoria%20Oficial...html

Comentarios para la respuesta: ¡Correcto!

Pregunta 4 0 de 0.33334 puntos

Supongamos un sistema con memoria paginada en el que las direcciones de memoria son de 16 bits, siendo los 4 bits de mayor peso el número lógico de página (NLP). La traducción a la dirección física se hace mediante una tabla de páginas en la unidad de gestión de memoria. En dicha tabla se tiene para cada página, además del número de marco, un bit P que indica si la página está presente (P=1) o ausente (P=0).

A la vista de la tabla de página, establezca la asociación entre cada página y su traducción (o fallo de página, si procede).

NLP	Marco	P	NLP	Marco	P	NLP	Marco	P	NLP	Marco	P
0	0	1	4	1	1	8	F	0	C	F	0
1	2	1	5	3	1	9	F	0	D	F	0
2	4	1	6	5	1	A	F	0	E	F	0
3	6	1	7	6	1	B	F	0	F	F	0

Pregunte Correspondencia correcta Correspondencia seleccionada
 0000 ☒ A. 0000 ☐ D. Fallo de página
 1000 ☒ B. 2000 ☒ B. 2000
 7FFF ☒ C. 6FFF ☒ C. 6FFF
 8FFF ☒ D. Fallo de página ☐ A. 0000

Todas las opciones de respuesta
 A. 0000
 B. 2000
 C. 6FFF
 D. Fallo de página

Comentarios para respuesta: No. Debe usar el dígito superior de la dirección para indexar la tabla de página, y de ahí obtener el número de marco, siempre y cuando el bit de presente sea P=1. Si el bit de presente es P=0, eso significa que la dirección produce un fallo de página.

Pregunta 5 -0.08333 de 0.33334 puntos

35-

De los algoritmos de asignación de memoria que hemos visto en clase, es de esperar que el más eficiente desde el punto de vista de la cantidad de cálculo necesario para servir una petición sea..

-... asignación por siguiente ajuste

-36

Ejecutamos un proceso que sabemos que hará referencia a P páginas diferentes durante su ejecución. Si dicho proceso lo ejecutamos con M marcos, ¿cuántos fallos de página se producirán **durante el arranque en frío** de dicho proceso?

-Se producirá un número F de fallos, donde F es el menor de M y P

37- ¿Cuándo se deben ejecutar los algoritmos de **detección** de interbloqueo?

-Cuando se solicita un recurso que está ocupado

-38 En un sistema se ejecutan cuatro procesos que comparten cuatro tipos de recursos. En un momento dado, el estado de asignación de recursos a procesos es el que viene dado por la matriz A (recursos asignados) y el vector D (recursos disponibles) que se muestran a continuación, estando en espera de satisfacerse las peticiones que se muestran en la matriz S. ¿Se encuentra en este momento el sistema en estado de interbloqueo?

		R 1	R 2	R 3	R 4			R 1	R 2	R 3	R 4	D =	R 1	R 2	R 3	R 4
A =	P 1	0	1	0	0	S =	P 1	0	1	0	0		2	0	0	0
	P 2	2	2	0	0		P 2	0	0	1	0					
	P 3	0	0	2	0		P 3	0	0	0	1					
	P 4	0	0	0	2		P 4	0	0	0	0					

-No, pues las esperas se resuelven en el siguiente orden: P4, P3, P2, P1