La segmentación:

| 1. Permite compartir memoria entre diferentes procesos 2. Requiere que las direcciones se vinculen en tiempo de carga 3. Permite que los diferentes segmentos de cada proceso se ubiquen y reubiquen en toda la memoria sin limitaciones ni operaciones extra, siempre que haya espacio disponible 4. Puede ahorrar espacio en memoria, utilizando una única tabla de segmentos para todos los procesos 5. Todas las opciones son CORRECTAS 6. Ninguna opción es CORRECTA |
| --- |
| Correcta: A |

La paginación simple con memoria virtual:

| 1. Permite que las diferentes páginas de cada proceso se ubiquen y reubiquen en toda la memoria sin limitaciones, siempre que haya espacio disponible 2. No permite compartir memoria entre diferentes procesos 3. Requiere que las direcciones se vinculen en tiempo de carga 4. Puede ahorrar espacio en memoria, utilizando una única tabla de páginas para todos los procesos 5. Todas las opciones son CORRECTAS 6. Ninguna opción es CORRECTA |
| --- |
| Correcta: A |

El fenómeno de Thrashing se produce cuando:

| 1. En un período de tiempo, un procesos requiere más páginas en memoria principal de las que se puede alojar en su conjunto residente.. 2. Los procesos no activan el principio de localidad. 3. El algoritmo de reemplazo es FIFO. 4. Sólo cuando no hay marcos disponibles en memoria y se utiliza asignación fija. 5. La ejecución de las instrucciones es más veloz que el tiempo requerido para recuperar una página de Memoria Virtual. 6. No hay TLB o la cantidad de aciertos no es suficiente. |
| --- |
| Correcta: |

El fenómeno de Thrashing puede evitarse:

| 1. Reduciendo el grado de multiprogramación cuando la asignación de marcos es variable. 2. Reduciendo el tamaño del conjunto residente. 3. Habilitando el principio de localidad. 4. Usando el algoritmo LRU. 5. Utilizando asignación fija de marcos. |
| --- |
| Correcta: A |

Peter detecta que siempre tiene poca RAM disponible cuando pone a correr dos procesos en particular (que corren en forma concurrente) por más que tenga memoria virtual. Analizando un poco el sistema y aprovechando sus conocimientos de sistemas operativos se da cuenta que los procesos son muy grandes por lo que siempre tiene una gran parte de RAM ocupada por las tablas de páginas. ¿Qué le recomendaría a Peter para tener más memoria disponible?

| 1. Que utilice un esquema de paginación jerárquica 2. Que utilice un esquema de tabla de páginas invertida con tabla de hash 3. Que elija un tipo de sustitución de páginas global 4. Que limite la cantidad de frames disponibles para dichos procesos y utilice sustitución local 5. Que agregue una TLB que permita indicar de qué proceso es la página 6. A y B 7. C y E 8. Todas las recomendaciones le podrían servir a Peter 9. Ninguna de las recomendaciones le sirve a Peter |
| --- |
| Correcta: F |

Peter posee un sistema con memoria virtual que solía usar paginación con tablas de página convencional y decide empezar a usar paginación jerárquica para optimizar la memoria. Sin embargo, empieza a percibir que los procesos comienzan a ralentizarse levemente. En el sistema suele haber un nivel de multiprogramación de 20 procesos que se ejecutan en forma concurrente. Ayude a Peter y realice recomendaciones para mejorar su sistema.

| 1. Volver a usar paginación normal ya que la paginación jerárquica siempre tiene mala performance 2. Utilizar un esquema de tabla de páginas invertida con tabla de hash ya que podría ser más rápido, pero no se puede asegurar. 3. Cambiar el esquema de asignación a Segmentación paginada con paginación jerárquica 4. Limitar la cantidad de frames disponibles para dichos procesos y utilizar sustitución local 5. Agregar una TLB que contenga PID, nro pág y nro de frame 6. A y B 7. B y E 8. C y E 9. Todas las recomendaciones le podrían servir a Peter 10. Ninguna de las recomendaciones le sirve a Peter |
| --- |
| Correcta: G |

¿Cuál de las siguientes afirmaciones es CORRECTA sobre la administración de archivos?

| 1. Todas las estrategias de asignación de bloques generan fragmentación interna pero sólo una externa 2. El Journaling sirve para evitar inconsistencias en los archivos por accesos concurrentes entre varios procesos 3. La estrategia de asignación de bloques indexada con varios niveles genera que apenas se crea un nuevo archivo se asigne toda la jerarquía de bloques de punteros. 4. La asignación de bloques enlazada es la que más favorece la velocidad del tipo de acceso secuencial sobre los archivos. 5. Todas las opciones son CORRECTAS 6. Ninguna de las opciones es CORRECTA |
| --- |
| Correcta: A |

¿Cuál de las siguientes afirmaciones es CORRECTA sobre la administración de archivos?

| 1. Todas las estrategias de administración de bloques generan fragmentación interna pero la asignación encadenada es la que genera en mayor cantidad 2. El Journaling sirve para evitar inconsistencias en los archivos por accesos concurrentes entre varios procesos 3. La estrategia de asignación de bloques indexada con varios niveles genera que tengamos que hacer accesos extra a disco para traernos los bloques de punteros. 4. La asignación de bloques indexada es la que más favorece la velocidad del tipo de acceso secuencial sobre los archivos. 5. Todas las opciones son CORRECTAS 6. Ninguna de las opciones es CORRECTA |
| --- |
| Correcta: C |

Indique cuál de las siguientes afirmaciones es correcta sobre el filesystem FAT

| 1. Si un cluster perteneciente a un archivo se daña y no puede ser leído, sólo se puede recuperar la información del comienzo del archivo hasta el cluster dañado (no inclusive) 2. Si un cluster perteneciente a un archivo se daña y no puede ser leído, no se puede recuperar el resto del archivo 3. Si un cluster perteneciente a un archivo se daña y no puede ser leído, se puede recuperar el archivo completo si existe una copia extra de la tabla FAT en otro lugar del volumen 4. Si un cluster perteneciente a un archivo se daña y no puede ser leído, igual se puede recuperar el resto del archivo, independientemente de que exista o no una copia extra de la tabla FAT en otro lugar del volumen 5. Ninguna de las afirmaciones es correcta |
| --- |
| Correcta: |

Indique cuál de las siguientes afirmaciones es correcta sobre el filesystem FAT

| 1. El tamaño máximo teórico de un archivo podría variar si la partición donde fue formateada es agrandada o achicada (volviendo a formatear con la misma configuración) 2. El tamaño máximo real de un archivo podría variar si la partición donde fue formateada es agrandada o achicada (volviendo a formatear con la misma configuración) 3. La lectura de un byte del primer cluster de un archivo no requiere acceso a disco alguno (asumiendo que los cluster no son cacheados en memoria ram). 4. La lectura de un byte del último cluster de un archivo implica inevitablemente la lectura de todos los clusters previos del mismo. 5. Ninguna de las afirmaciones es correcta |
| --- |
| Correcta: B |

Indique cuál de las siguientes afirmaciones es correcta sobre el filesystem EXT2

| 1. El tamaño máximo teórico de un archivo puede ser más grande que el tamaño máximo teórico del filesystem 2. El acceso a un byte específico de un archivo siempre implica un solo acceso a disco 3. El acceso a un byte específico de un archivo siempre implica varios accesos a disco 4. Un bloque del volumen que tenga datos podría en otro momento tener punteros 5. Ninguna de las afirmaciones es correcta |
| --- |
| Correcta: A y D |

Indique cuál de las siguientes afirmaciones es correcta sobre el filesystem EXT2

| 1. El tamaño máximo teórico de un archivo siempre es más grande que el tamaño máximo real del archivo 2. Agrandar el tamaño de puntero (manteniendo la estructura del inodo y el tamaño de bloque) es una forma de lograr que los archivos puedan ser más grandes 3. El acceso al último byte de un archivo cuyo inodo tiene todos sus punteros del inodo en uso, podría requerir un solo acceso a disco 4. Un bloque del volumen que tenga datos, será usado siempre para datos 5. Ninguna de las afirmaciones es correcta |
| --- |
| Correcta: E |

| El algoritmo del banquero   1. Tiene alto overhead, porque se corre cada vez que un proceso solicita un recurso 2. Tiene alto overhead, dado que calcula todas las combinaciones posibles, y si ninguna lleva a deadlock considera que hay un estado seguro 3. Tiene mejor performance que prevención, porque se corre en intervalos regulares 4. Todas son correctas 5. Ninguna es correcta |
| --- |
| Correcta: A |

| El algoritmo de detección de deadlock   1. Se corre cada intervalos regulares, buscando procesos que no puedan continuar ejecutando 2. No permite la existencia de deadlocks, porque los detecta y trata inmediatamente antes de que ocurran 3. Ignora a los procesos que tienen recursos asignados pero no están haciendo pedidos 4. Busca una secuencia en la que todos los procesos podrían finalizar y si la encuentra se considera estado seguro 5. Todas las opciones son CORRECTAS 6. Ninguna de las opciones es CORRECTA |
| --- |
| Correcta: A |

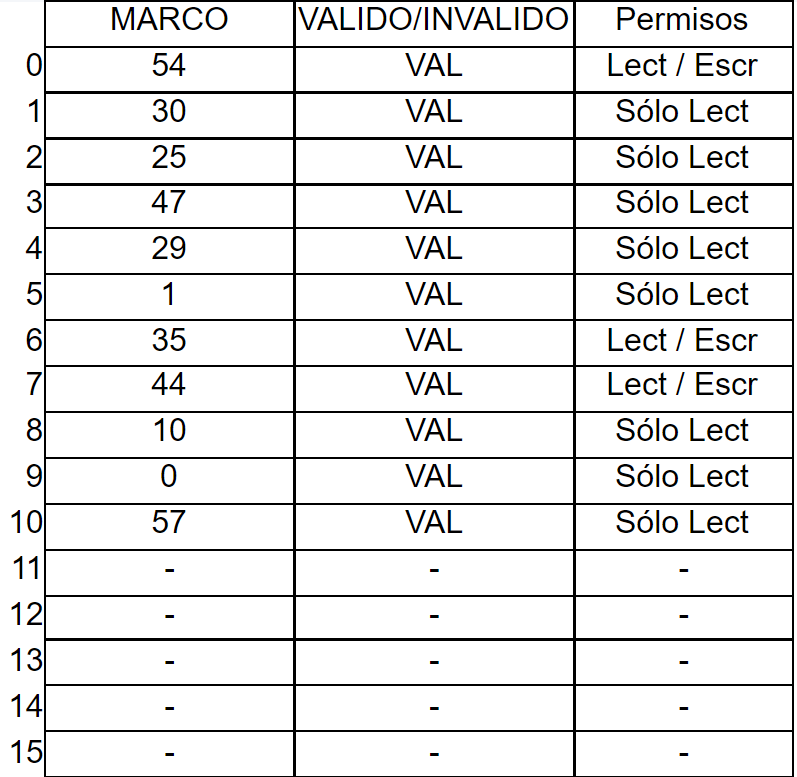
Indique cuál de las siguientes afirmaciones sobre deadlock es correcta

| 1. El algoritmo del banquero es difícil de implementar porque requiere conocer todas las peticiones posibles de cada proceso 2. El algoritmo de detección de deadlock es difícil de implementar porque requiere conocer todas las peticiones posibles de cada proceso 3. Los mecanismos de prevención de deadlock orientados a expropiación de recursos se pueden implementar para cualquier recurso 4. Todas son correctas 5. Ninguna de las afirmaciones es correcta |
| --- |
| Correcta: A |

Indique cuál de las siguientes afirmaciones sobre deadlock es correcta

| 1. El algoritmo de detección de deadlock es fácil de implementar porque requiere conocer todas las peticiones actuales de cada proceso y los recursos asignados 2. El algoritmo del banquero es factible de implementar porque requiere conocer todas las peticiones actuales de cada proceso y los recursos asignados 3. Los mecanismos de prevención de deadlock orientados a prevenir la espera circular se pueden implementar para variables compartidas 4. Todas son correctas 5. Ninguna de las afirmaciones es correcta |
| --- |
| Correcta: A |

En un sistema, con una memoria 512 KiB administrada con paginación simple sin memoria virtual, está ejecutando sólo un proceso con la siguiente tabla de páginas (la tabla está completa):



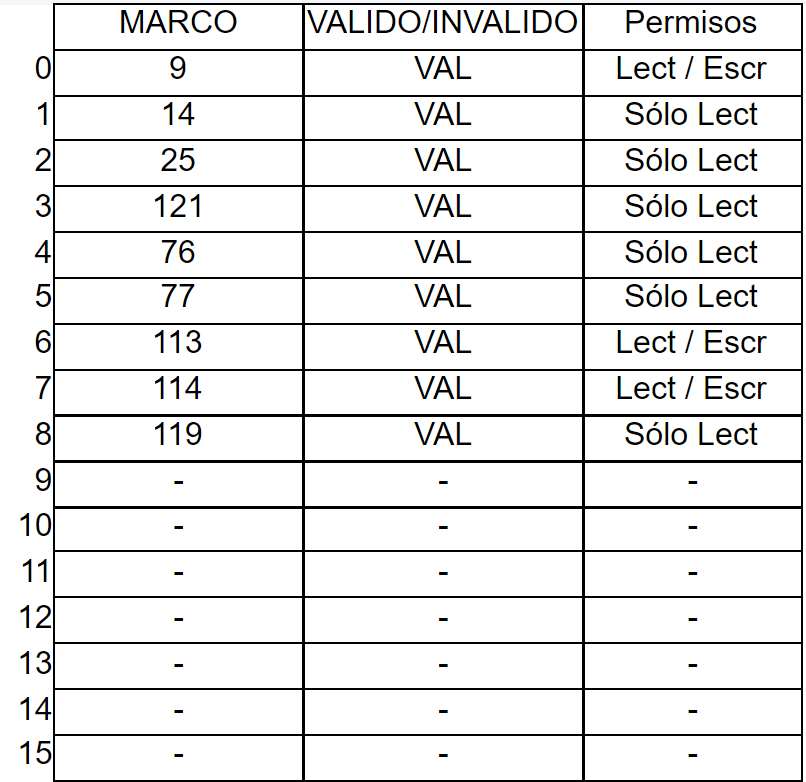
Sabiendo que la primera dirección válida proceso está en la dirección 442368 (decimal) de la memoria, responda:

1. Para la dirección lógica 23000 en decimal. Indique a qué página del proceso pertenece. Traduzca la dirección indicando el número de marco y desplazamiento (todo en decimal): **2-25-6616**
2. ¿Hasta cuántas instancias de este proceso puede haber en memoria?

512KiB / (11\*8KiB)  
**5**

1. Fragmentación interna máxima que podría haber por proceso:   
   **8191**
2. Teniendo en cuenta que las páginas de “Sólo Lectura”, pertenecen al código del proceso. ¿Cuánto procesos podría haber en memoria si se usan técnicas de compartición? **18**

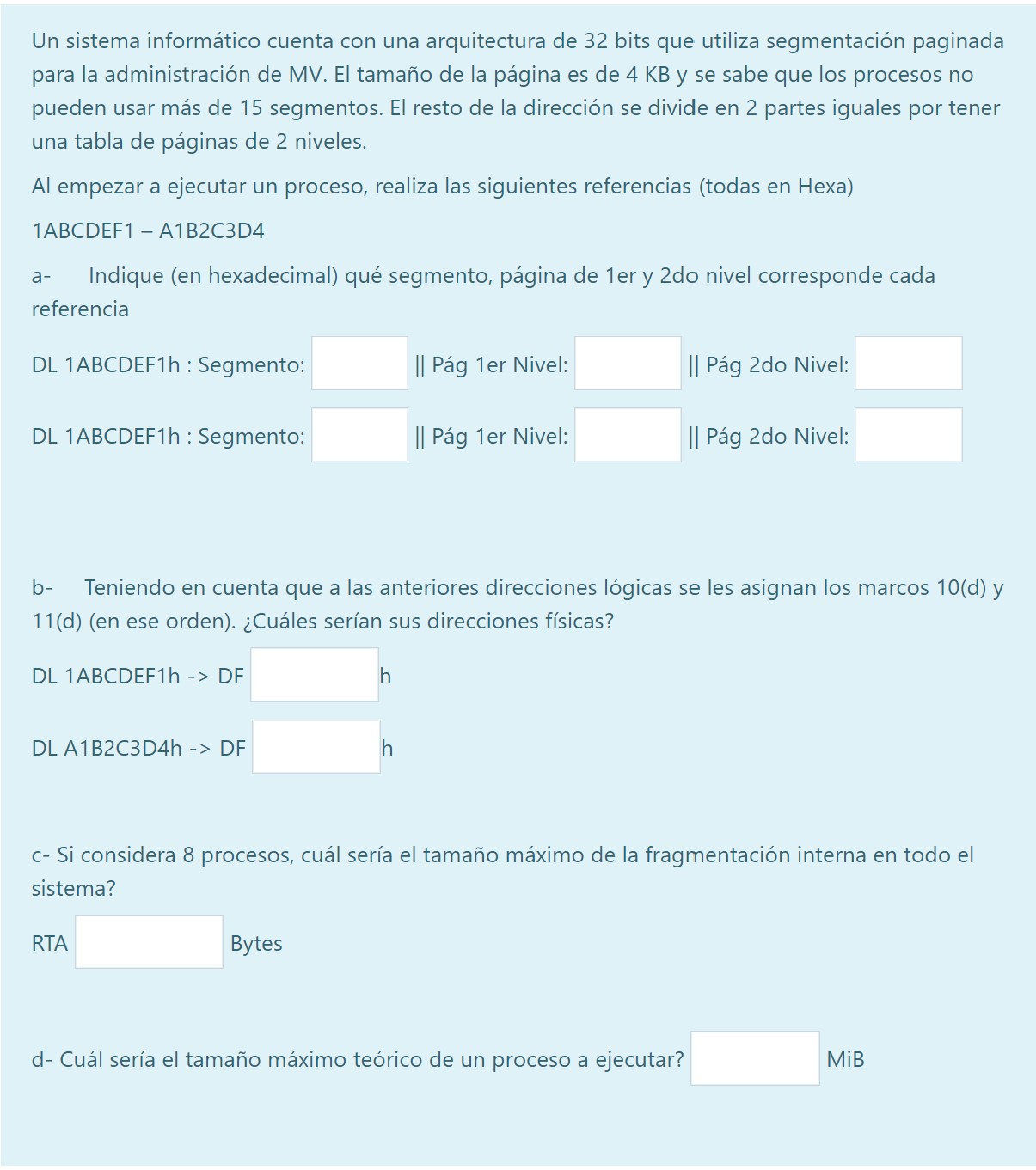
En un sistema, con una memoria de 512 KiB administrada con paginación simple sin memoria virtual, está ejecutando sólo un proceso con la siguiente tabla de páginas (la tabla está completa):

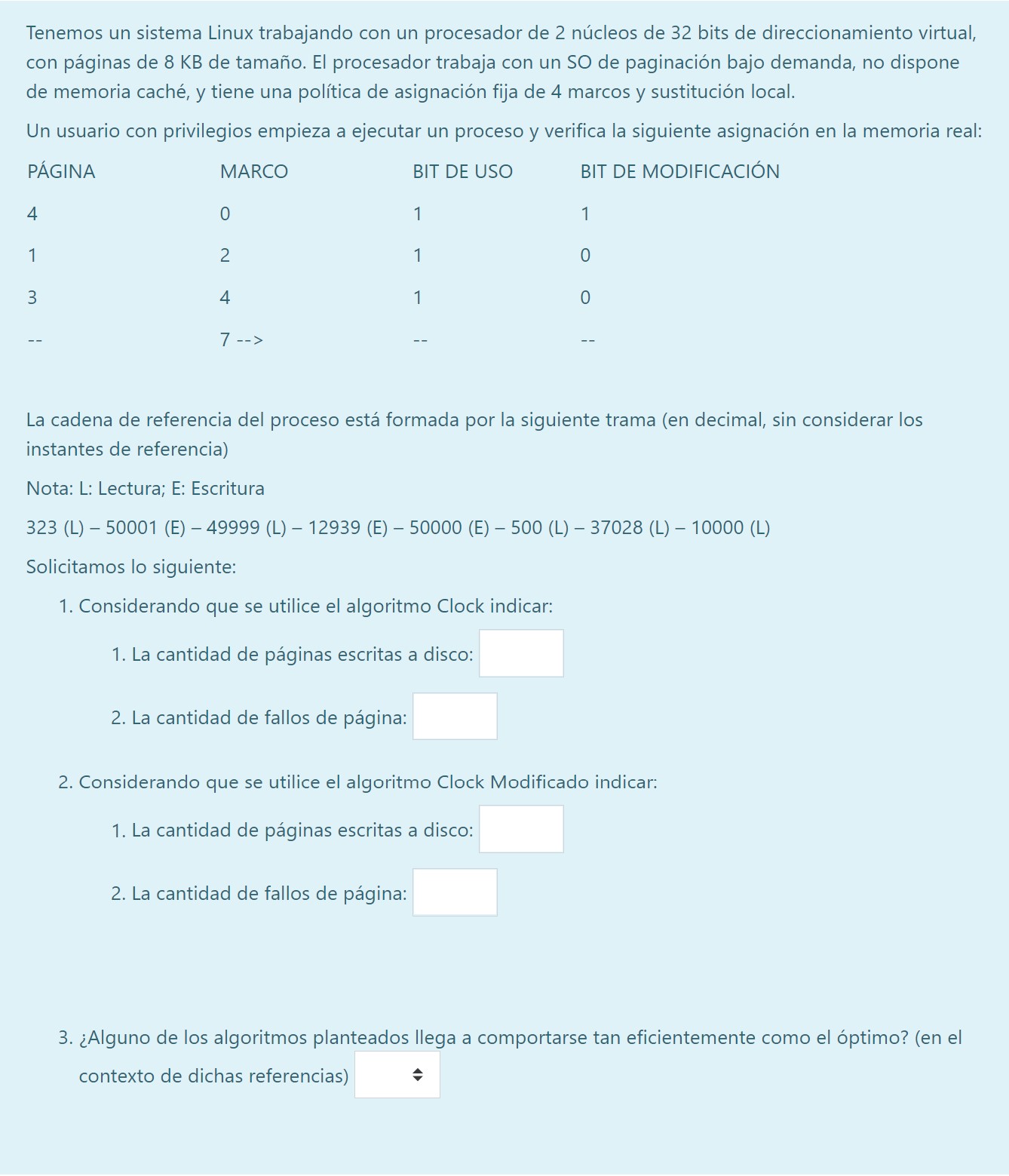


Sabiendo que la primera dirección válida del proceso está en la dirección 36864 (decimal) de la memoria, responda:

36864 = 9 \* tamMarco + 0  
 tamMarco = 4096

1. Para la dirección lógica 32000 en decimal. Indique a qué página del proceso pertenece. Traduzca la dirección indicando el número de marco y desplazamiento (todo en decimal): 7-114-
2. ¿Hasta cuántas instancias de este mismo proceso puede haber en memoria?  
   **14**
3. Fragmentación interna máxima que podría haber por proceso:  
   **4095**
4. Teniendo en cuenta que las páginas de “Sólo Lectura”, pertenecen al código del proceso. ¿Cuántos procesos podría haber en memoria si se usan técnicas de compartición? **40**

C= (4KiB -1)\*15\*8 = 491400

Referencias = 0 - 6 - 6 - 1 - 6 - 0 - 4 - 1

Un sistema utiliza UFS (unix file system) con direcciones de 8 bytes e inodos con 10 ptrs directos, 1 ind simple, 1 ind doble y 1 ind triple. En un momento se encuentra en memoria el contenido del directorio raíz que tiene el siguiente contenido (indicando nombre entrada y número de inodo) y que se sabe que ocupa 2 KiB.

| Dir raíz completo (bloque de datos 10) |
| --- |
| d facultad 9  d trabajo 11  d juegos 20 |

Contenido inodos

| Inodo 9 | ... | Inodo 12 | ... |
| --- | --- | --- | --- |
| ptrs directos:  bloque 50  - |  | ptrs directos  bloque 55 |  |

Contenido bloques

| ... | Bloque 50 | ... | Bloque 55 | ... |
| --- | --- | --- | --- | --- |
|  | d sistemas\_operativos 12  d fisica\_2 21  d gestion\_de\_datos 23  - |  | d videos 57  d parciales 60  - dudas.doc 100  - compilado\_guias.doc 101 |  |

Indique:

1. Tam máx archivo teórico en dicho FS valor aproximado (entero) en GiB: **32**
2. Cantidad de accesos a bloques para leer el primer y el último bloque de datos del archivo compilado\_guias.doc sabiendo que su tamaño es de 2,5 MiB. Tenga en cuenta los bloques necesarios que hay que traer de disco, sabiendo que los inodos y el bloque del directorio raíz se encuentran en memoria.
   1. Cantidad de accesos a bloques de directorio: **2**
   2. Cantidad de accesos a bloques para acceder al 1er bloque del archivo: **1**
   3. Cantidad de accesos a bloques para acceder al último bloque del archivo: **3**
3. Responda lo mismo que en b teniendo en cuenta que el archivo se encuentre en FAT 32 (mismo tamaño de bloque) e ignorando los accesos requeridos para los directorios y la tabla FAT.
   1. 1er bloque -> Accesos a bloques =
   2. Último bloque > Accesos a bloques =
4. Si quiere crear un link en el directorio raíz que apunte a “dudas.doc” dentro de “sistemas\_operativos”. ¿Qué convendría utillizar para agilizar su posterior acceso?

Un sistema utiliza UFS (unix file system) con direcciones de 8 bytes e inodos con 10 ptrs directos, 1 ind simple, 1 ind doble y 1 ind triple. En un momento se encuentra en memoria el contenido del directorio raíz que tiene el siguiente contenido (indicando nombre entrada y número de inodo) y que se sabe que ocupa 1 KiB.

| Dir raíz completo (bloque de datos 10) |
| --- |
| d facultad 9  d trabajo 11  d juegos 20 |

Contenido inodos

| Inodo 9 | ... | Inodo 12 | ... |
| --- | --- | --- | --- |
| ptrs directos:  bloque 50  - |  | ptrs directos  bloque 55 |  |

Contenido bloques

| ... | Bloque 50 | ... | Bloque 55 | ... |
| --- | --- | --- | --- | --- |
|  | d sistemas\_operativos 12  d fisica\_2 21  d gestion\_de\_datos 23  - |  | d videos 57  d parciales 60  d dudas 100  - complilado\_guias.doc 101 |  |

Indique:

1. Tam máx archivo teórico en dicho FS valor aproximado (entero) en GiB:
2. Cantidad de accesos a bloques para leer todo el archivo compilado\_guias.doc si el mismo tiene un tamaño de 5 MiB sabiendo que los inodos y el bloque del directorio raíz se encuentran en memoria.
   1. Accesos a bloques de directorios = **2**
   2. Accesos a bloques del archivo = **5161**
3. Responda lo mismo que en b teniendo en cuenta que el archivo se encuentre en FAT 32 (mismo tamaño de bloque) e ignorando los accesos requeridos para los directorios.
   1. Accesos a bloques = **5120**
4. Si quiere crear un link en el directorio raíz que apunte a “dudas” dentro de “sistemas\_operativos”. ¿Qué convendría utilizar para agilizar su posterior acceso?.

Dadas las matrices:

| Solicitudes Pendientes   |  | R1 | R2 | R3 | R4 | | --- | --- | --- | --- | --- | | P1 | 0 | 1 | 0 | 2 | | P2 | 0 | 0 | 1 | 0 | | P3 | 0 | 0 | 1 | 0 | | P4 | 1 | 0 | 0 | 0 | | Recursos Asignados   |  | R1 | R2 | R3 | R4 | | --- | --- | --- | --- | --- | | P1 | 1 | 0 | 2 | 0 | | P2 | 0 | 1 | 0 | 0 | | P3 | 0 | 0 | 0 | 1 | | P4 | 0 | 0 | 0 | 1 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Recursos Totales   |  | R1 | R2 | R3 | R4 | | --- | --- | --- | --- | --- | |  | 1 | 1 | 2 | 2 | |  |

Determine si hay deadlock, en caso que exista, elija una solución posible:  
P1 libera 2 instancias de R3

P1 libera 1 instancias de R1

P2 libera 1 instancias de R2

**P1 libera 1 instancias de R1 y una de R3**

Finalizar dos procesos al azar

Ninguna alternativa es correcta

No hay Deadlock

Teniendo en cuenta la respuesta anterior, indique la secuencia de finalización:

P4-P2-P3-P1

P1-P4-P2-P3

P4-P1-P3-P2

P3-P4-P1-P2

Ninguna alternativa es correcta

Dadas las matrices:

| Solicitudes Pendientes   |  | R1 | R2 | R3 | R4 | | --- | --- | --- | --- | --- | | P1 | 1 | 0 | 0 | 0 | | P2 | 0 | 2 | 0 | 1 | | P3 | 0 | 0 | 1 | 0 | | P4 | 1 | 0 | 0 | 0 | | Recursos Asignados   |  | R1 | R2 | R3 | R4 | | --- | --- | --- | --- | --- | | P1 | 0 | 1 | 0 | 0 | | P2 | 2 | 0 | 1 | 0 | | P3 | 0 | 1 | 0 | 0 | | P4 | 0 | 0 | 0 | 1 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Recursos Totales   |  | R1 | R2 | R3 | R4 | | --- | --- | --- | --- | --- | |  | 2 | 2 | 1 | 1 | |  |

Determine si hay deadlock, caso que exista, elija una solución posible:  
P2 libera 2 instancias de R1  
P2 libera 1 instancias de R3  
P4 libera 1 instancias de R4  
P2 libera 1 instancias de R1 y una de R3  
Finalizar dos procesos al azar  
Ninguna alternativa es correcta  
No hay Deadlock

Teniendo en cuenta la respuesta anterior, indique la secuencia de finalización:

P3-P4-P1-P2  
P3-P4-P2-P1  
P1-P2-P4-P3  
P3-P1-P2-P4  
Ninguna alternativa es correcta