

APELLIDOS, NOMBRE: _____

Calificación: 80% de la suma de puntos (“*”) asociados a cada pregunta.

Todas las respuestas han de estar justificadas.

No se puede utilizar internet ni ningún medio de comunicación.

1. (4*) Nos situamos en un sistema de **paginación sin memoria virtual**. Explique qué utilidad podría tener el campo “**modificación**” en la Tabla de Páginas.

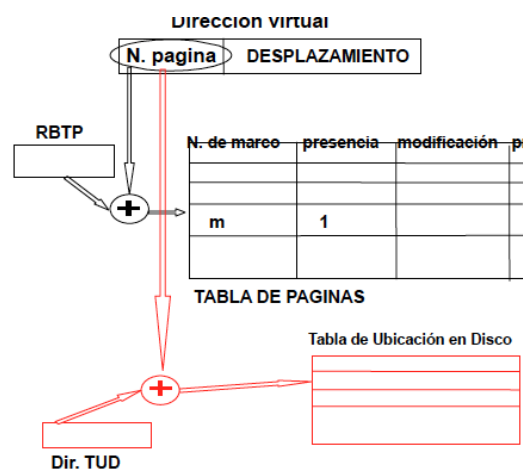
Si hay que retirar un proceso entero de memoria a almacenamiento secundario, evitaremos escribir aquellas paginas que no se modificaron desde que llegaron a memoria evitando así operaciones de escritura.

Puede ser interesante conocer qué nº de páginas de un proceso en memoria habría que reescribir si hubiera que llevarlo a almacenamiento secundario. De este modo si el SO tiene que elegir entre varios procesos para decidir a cuál quitarle memoria, en caso de igualdad se elegiría quien tiene un nº de págs modificadas menor.

2. (4*) Explique cuáles podrían ser los contenidos posibles de la fila **i** de la Tabla de Ubicación en Disco del proceso **j**

Algunas posibilidades:

- nº de bloque de disco donde está la página **i** del proceso **j** (en este caso no se utilizan archivos para guardar los procesos en disco, sino directamente número de bloque).
- si se utiliza un archivo global para guardar los espacios de memoria de todos los procesos, podría tener el desplazamiento desde el comienzo de este archivo donde está la página **i** del proceso **j**.



3. (6*) Nos situamos en la traducción de direcciones siendo el proceso **i** el proceso actual. Se utilizan dos tablas: la Tabla de Páginas (TP) y la Tabla de Ubicación en Disco (TUD); se utilizan los datos de sus respectivas direcciones de comienzo: para la Tabla de Páginas su dirección de comienzo se almacena en el RBTP (“Registro Base de la Tabla de Páginas”) y para la Tabla de Ubicación en Disco se menciona el dato de “Dirección de comienzo de la TUD). La pregunta es: ¿porqué la dirección de comienzo de la TP se almacena en un registro de la CPU mientras que la dirección de comienzo de la TUD es un dato en memoria?

La dirección de comienzo de la TP es utilizada por la MMU, por lo que no debe ubicarse en un lugar distante como la unidad de memoria principal. Por otro lado, se utiliza muy frecuentemente, en cada alusión a una dirección de memoria, por lo que es muy rentable que haya un registro para este propósito.

Sin embargo la dirección de comienzo de la TUD no es utilizada por la MMU sino únicamente por el SO. por lo que la primera razón dicha anteriormente no se da. Por otra parte su frecuencia de uso es muchísimo menor, pues se usa solo cuando hay falta de página. Con estas características no se justifica que haya una registro para este dato, aunque si lo hubiera su acceso sería más rápido claro, pero también mas caro por lo que el nº de registros suele ser bajo y estar muy justificado las razones por las que se usa.

4. Nos situamos en un esquema de paginación con memoria virtual en que el número de marcos asociados a cada proceso es al menos 2.

a) (2*) En el algoritmo de sustitución FIFO, ¿podría ocurrir que se elija para la sustitución la página objeto de la referencia anterior?

b) (2*) En el algoritmo de sustitución FIFO, ¿podría ocurrir que se elija para la sustitución la página objeto de la referencia siguiente?

c) (2*) En el algoritmo de sustitución LRU, ¿podría ocurrir que se elija para la sustitución la página objeto de la referencia anterior?

d) (2*) En el algoritmo de sustitución LRU, ¿podría ocurrir que se elija para la sustitución la página objeto de la referencia siguiente?

5. (4*) Comparando paginación con segmentación, se dice que la paginación se presta mal para implementar la protección, ¿porqué?

Con segmentación se divide el espacio de direcciones lógicas de un programa en unidades (segmentos) que tienen un significado particular (código, datos, etc...). Por tanto es mas natural asociar una determinanda protección a todo un segmento que a todo un conjunto de páginas. La protección asociada a un segmento se guarda solo una vez, asociada a todo el segmento. Sin embargo, con paginación, ese dato tendría que repetirse para cada una de las páginas, en el campo correspondiente de la tabla de páginas.

6. (4*) Inconvenientes del algoritmo LRU

En general, se dice que es más costoso, que requiere más recursos, aunque podría estudiarse que esto se economizara. Un inconveniente importante es que requiere de un comportamiento determinado del hardware: que en cada referencia a MP se actualice un determinado dato (el contador, o actualizar la pila de páginas en orden de acceso.....). Por un lado es costoso el que esa actuación sea necesaria en toda referencia a memoria, por otro lado el software que se ejecute en la máquina estará precondicionado por esa actuación determinada del hardware.

7. Desde que se escribe un programa fuente (en determinado lenguaje distinto al lenguaje máquina), hasta el momento en que se está realizando un acceso a memoria, hay varios momentos en que podría realiarse la traducción de direcciones (de lógicas a físicas).....

a) (2*) Si la traducción de direcciones se realiza en la carga del ejecutable en memoria (y no antes), ¿en distintos momentos de esta ejecución el proceso puede ubicarse en distintas zonas de memoria?

b) (2*) Si la traducción de direcciones se realiza en la carga del ejecutable en memoria (y no antes), ¿en distinas ejecuciones el proceso puede ubicarse en distintas zonas de memoria?

c) (2*) Si la traducción de direcciones se realiza en cada acceso a memoria ¿en distintos momentos de esta ejecución el proceso puede ubicarse en distintas zonas de memoria?

8. Nos situamos en un esquema de paginación a 2 niveles, concretamente en el ejemplo que se proporciona en el folio siguiente.

TABLAS DE PAG. A 2º NIVEL:

MEMORIA REAL:

n° marco:

| | |
|-------|-------------|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | No presente |
| 4 | No presente |
| | |
| | |
| | |
| | |
| | |
| 31 | No presente |

| | |
|---|------------------------|
| 0 | Codigo 0 |
| 1 | Codigo 1 |
| 2 | Datos 0 |
| 3 | Pila |
| 4 | Pag0 de TP a 2º nivel |
| 5 | Pag31 de TP a 2º nivel |
| 6 | TP a 1er nivel |

TABLA DE PAG.
A 1er NIV

| | |
|-----|-------------|
| 0 | 4 |
| 1 | No presente |
| 2 | No presente |
| 3 | No presente |
| ... | ... |
| 30 | No presente |
| 31 | 5 |

| | |
|-----|-------------|
| 0 | No presente |
| 1 | No presente |
| 2 | No presente |
| 3 | No presente |
| ... | ... |
| 30 | No presente |
| 31 | No presente |

PAG 1

| | |
|-----|-------------|
| 0 | No presente |
| 1 | No presente |
| 2 | No presente |
| 3 | No presente |
| ... | ... |
| 30 | No presente |
| 31 | 3 |

PAG 31

a) (2*) En la materia general sobre procesos, se dice que en el PCB (Bloque de Control de Proceso) se almacena la información sobre memoria que el SO necesita mantener para cada proceso. De entre los datos que se manejan en el ejemplo, ¿qué se almacenaría en el PCB en este caso, y qué valor tiene para el proceso del ejemplo?

Dirección de comienzo de la TP a primer nivel: marco número **6**

b) (2*) Explique qué modificaciones habría que hacer en la figura de la página 2 del ejemplo para representar el hecho de que la Pila del proceso pasa a tener tamaño 0. Representelo en la figura y justifique a continuación su respuesta.

Hay que decir que este es un supuesto para hacer un ejercicio, pues la pila nunca se queda a 0. Cuando se ejecuta main tiene un marco donde están al menos el "nombre del programa" y argumentos.

Por otra parte, no es necesario "compactar" la memoria! pues la paginación permite precisamente que trozos estén dispersos, no hemos de llevarlos todos justos al principio.

MEMORIA REAL:

n° marco:

| | |
|-------|-------------|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | No presente |
| 4 | No presente |
| | |
| | |
| | |
| | |
| | |
| 31 | No presente |

PAG 0

| | |
|---|-----------------------|
| 0 | Codigo 0 |
| 1 | Codigo 1 |
| 2 | Datos 0 |
| 3 | libre |
| 4 | Pag0 de TP a 2º nivel |
| 5 | libre |
| 6 | TP a 1er nivel |

TABLA DE PAG.
A 1er NIV

| | |
|-----|-------------|
| 0 | 4 |
| 1 | No presente |
| 2 | No presente |
| 3 | No presente |
| ... | ... |
| 30 | No presente |
| 31 | No presente |

| | |
|-------|-------------|
| 0 | No presente |
| 1 | No presente |
| 2 | No presente |
| 3 | No presente |
| | |
| 30 | No presente |
| 31 | No presente |

PAG 1

| | |
|-------|-------------|
| 0 | No presente |
| 1 | No presente |
| 2 | No presente |
| 3 | No presente |
| | |
| 30 | No presente |
| 31 | No presente |

PAG 31

c) (2*) Explique qué modificaciones habría que hacer en la figura de la página 2 del ejemplo para representar el hecho de que la zona de datos aumenta en 20 bytes. Decida usted qué marco asigna para almacenar estos nuevos 20 bytes. Representelo en la figura y justifique a continuación su respuesta.

TABLAS DE PAG. A 2º NIVEL:

MEMORIA REAL:

n° marco:

| | |
|-------|-------------|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 7 |
| 4 | No presente |
| | |
| | |
| | |
| | |
| 31 | No presente |

PAG 0

| | |
|---|------------------------|
| 0 | Codigo 0 |
| 1 | Codigo 1 |
| 2 | Datos 0 |
| 3 | Pila |
| 4 | Pag0 de TP a 2º nivel |
| 5 | Pag31 de TP a 2º nivel |
| 6 | TP a 1er nivel |
| 7 | Datos 1 |
| | Pag1 de TP a 2º nivel |

TABLA DE PAG.
A 1er NIV

| | |
|-----|-------------|
| 0 | 4 |
| 1 | 8 |
| 2 | No presente |
| 3 | No presente |
| ... | ... |
| 30 | No presente |
| 31 | 5 |

| | |
|-------|-------------|
| 0 | 7 |
| 1 | No presente |
| 2 | No presente |
| 3 | No presente |
| | |
| 30 | No presente |
| 31 | No presente |
| PAG 1 | |

| | |
|-------|-------------|
| 0 | No presente |
| 1 | No presente |
| 2 | No presente |
| 3 | No presente |
| | |
| 30 | No presente |
| 31 | 3 |

PAG 31

[illegible]

9. (4*) Nos situamos en el algoritmo de sustitución variable basado en el Working Set. Encuentre el error en la figura siguiente, y describa la corrección a realizar.

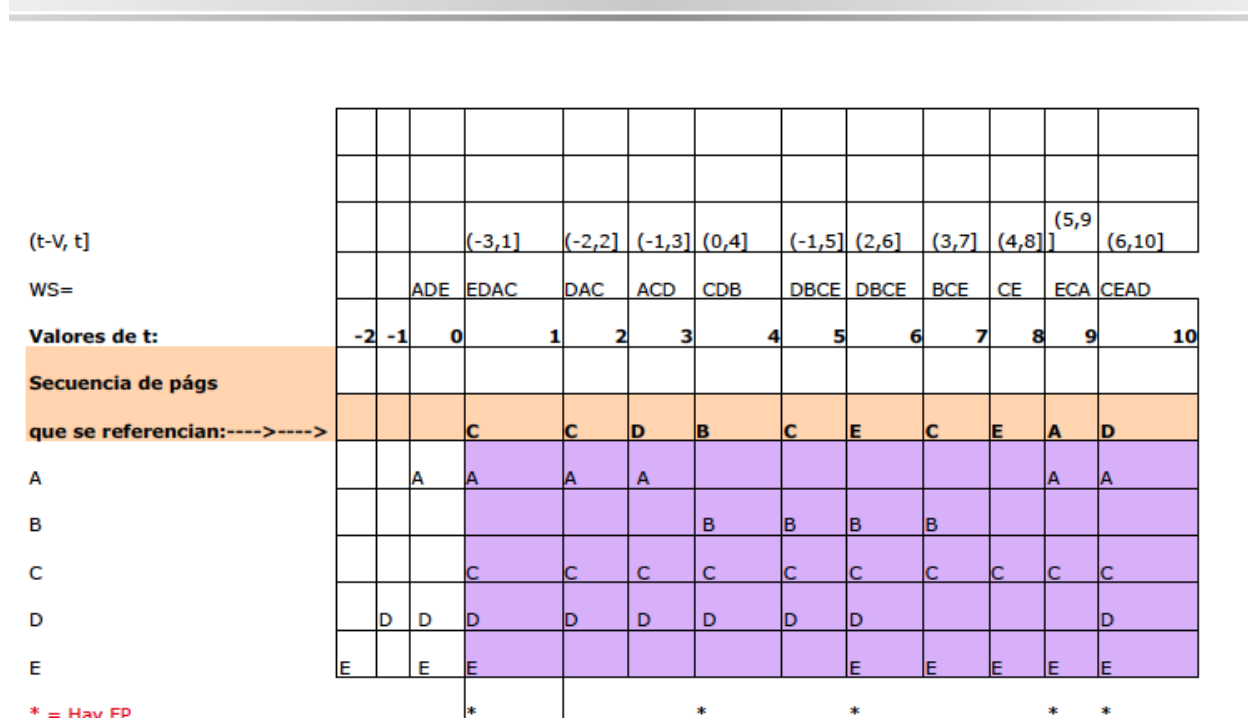
Ejemplo de algoritmo basado en el WS con V = 4

En t=-2 se referencia E

En t=-1 se referencia D

En t_0 se referencia A, y WS={A,D,E}

Comenzamos a ejecutar el algoritmo en t=1



10. (4*) Nos situamos en un sistema de memoria virtual con paginación. Explique cuál debe ser el número mínimo de marcos que se deben asignar a un proceso. Explique qué ocurriría si se asignara a un proceso un número de marcos inferior a dicho mínimo.

Es el máximo número de faltas de página que genere una instrucción reinicialable en la arquitectura en cuestión.