



La MMU de un computador utiliza un sistema de traslación de direcciones con las siguientes características:

Dirección física: 14 bits

Dirección virtual: 16 bits

Tamaño de página: 1 KB

Además se sabe que la memoria física ocupa la primera mitad del espacio de direcciones físico.

En la figura que se proporciona a la derecha se muestra la tabla de páginas de un proceso que se encuentra en ejecución en el sistema. Todas las entradas de la tabla de páginas no mostradas corresponden a páginas no utilizadas por el proceso.

El proceso utiliza dos páginas de código (C1 y C2), dos de datos (D1 y D2) y dos de pila (P1 y P2). La tabla que se muestra a continuación indica qué páginas virtuales utiliza el proceso para direccionar sus páginas de código, datos y pila.

Página del proceso	Página virtual asignada
C1	0C
C2	0D
D1	10
D2	11
P1	20
P2	21

A partir de toda esta información contesta a las preguntas A), B) y C) que se proporcionan a continuación.

- A) Determina cuál sería la dirección física generada por la MMU al recibir ésta la dirección virtual más significativa utilizable por el proceso. Contesta en hexadecimal.

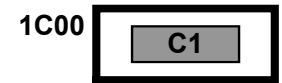
1FFF **0,75**

- B) A continuación, se proporciona un esquema del espacio de almacenamiento físico del computador, formado por la memoria física (organizada en marcos de página) y el disco. Debes rellenar este esquema indicando las posiciones que ocupan en él las páginas C1, C2, D1, D2, P1 y P2 del proceso. A la hora de rellenar la figura ten en cuenta los siguientes ejemplos:

Tabla de Páginas

Nº Pag. Virtual	Pre-sencia	Nº Pag. Fis. Offset Dis.
00	No	---
•	•	•
0C	Si	1
0D	No	Offset X
•	•	•
10	Si	4
11	No	Offset Y
•	•	•
20	Si	5
21	Si	7
•	•	•
3F	No	---

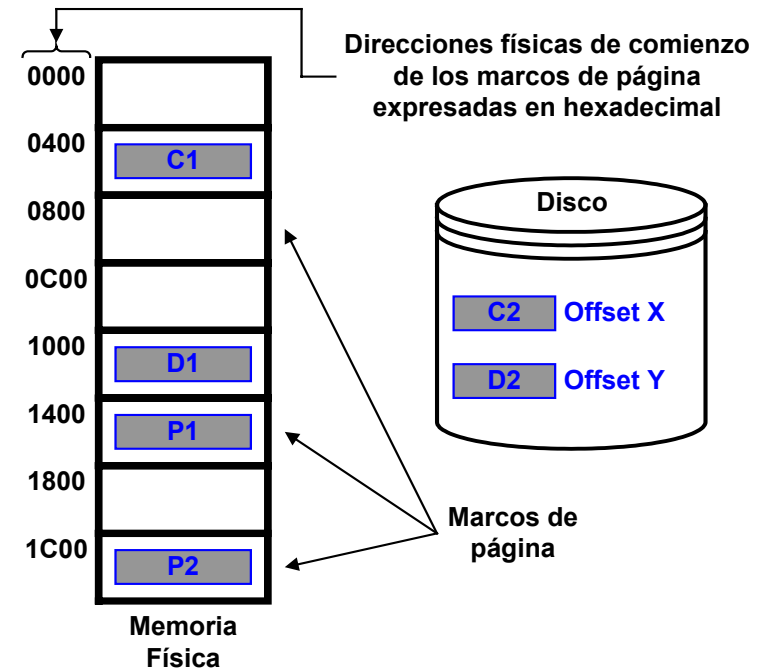
- 1) Si por ejemplo la página C1 estuviera en el marco de página ubicado en la dirección 1C00, dibujarías dentro de dicho marco lo que se muestra a la derecha →



- 2) Si por ejemplo la página C1 estuviera en el disco en el offset X, entonces dibujarías dentro del disco lo que se muestra a la derecha →



A Rellenar **0,75**



- C) Si la página de datos D1 está formada por un *array* de 512 elementos de tipo *char*, seguido de otro *array* de 128 elementos de tipo entero (entendiéndose que cada entero ocupa 4 bytes), determina la dirección física que generará la MMU cuando el programa intente acceder al elemento 2 del *array* de enteros (téngase en cuenta que los elementos de un *array* se numeran a partir del 0). Contesta en hexadecimal.

1208 **1**

A

- Indica y describe brevemente los componentes básicos de un disco duro

0,75

Platos: Están formados por una aleación rígida de aluminio y recubiertos por una capa de material magnético sobre la que se graba la información.

Motor de giro: Su objetivo es hacer girar los platos a velocidad constante.

Cabezas de lectura/escritura: Su objetivo es escribir información sobre la superficie de los platos y leer de ellos. Hay una cabeza por cada superficie.

Brazo: Su objetivo es servir de soporte para la cabeza.

Actuador: Es un servomotor encargado de mover los brazos para posicionar las cabezas en las posiciones de los platos requeridas.

- Indica cuáles son las tres operaciones básicas que llevan a cabo las herramientas de desarrollo (compilador + linker) durante el proceso de generación de un programa ejecutable:

0,5

Generan el código máquina, que está formado por instrucciones y datos.

Almacenan el código generado en el disco en un fichero.

Asignan direcciones, virtuales, a las instrucciones y datos generadas.

- Indica cuál o cuáles de las siguientes afirmaciones son ciertas. Contesta NINGUNA si crees que ninguna lo es.

- A) La API de un sistema operativo es el conjunto de rutinas software que implementan los servicios de dicho sistema operativo.
- B) En las llamadas al sistema el intercambio de información entre el programa que hace la llamada y el sistema operativo sólo puede realizarse mediante los registros de la CPU.
- C) En la plataforma Windows la interfaz gráfica de comandos se implementa mediante un proceso de sistema.
- D) En un sistema operativo multiprogramado, la planificación de trabajos es el mecanismo utilizado para decidir qué programas se cargan en la memoria para su ejecución.

0,5

C y D

- Contesta a las siguientes preguntas breves:

0,75

¿Qué es un SDK? Indica los elementos habituales que contiene

conjunto de herramientas (tales como librerías, ficheros de cabecera, etc.), que permiten desarrollar componentes software que usen los servicios de otro componente software que actúa como proveedor.

Indica la razón por la que un proceso que se encuentra en el estado "en ejecución" pasa al estado "listo"

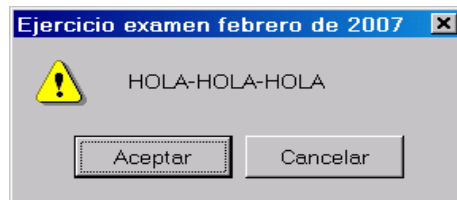
Porque ha expirado su quantum.

¿Qué ocurre cuando la MMU genera una dirección virtual no asignada (ni a memoria física ni a disco) y cómo las trata el sistema operativo?

Se genera una excepción de protección general, que indica una violación de acceso a memoria. Esto provoca una transferencia de control al SO, el cual elimina el programa causante de la excepción.



El objetivo de un programa es mostrar mediante una ventana de mensaje una cadena de caracteres que el usuario introduce a través de la consola. En una determinada ejecución de este programa en la que el usuario ha introducido la cadena "HOLA-HOLA-HOLA", el programa muestra la ventana de mensaje siguiente:



Para generar esta ventana, el programa utiliza la función `MessageBox()`. El programa termina imprimiendo en la consola la cadena "Se ha pulsado Aceptar", cuando se pulsa el botón `Aceptar`. En el caso de que se pulse el botón `Cancelar`, en la consola no se imprime nada.

(NOTA: debe tenerse en cuenta que el botón `Aceptar` se corresponde con el identificador OK en las constantes definidas en los ficheros de cabecera)

A continuación se proporciona el listado del programa perfilado mediante comentarios. Debes rellenar los huecos del programa teniendo en cuenta los comentarios.

```
#include <windows.h>
#include <stdio.h>

main()
{
    char texto[80];
    char titulo[]="Ejercicio examen febrero de 2007";
    int valor_ret; // Para recoger el valor devuelto por MessageMox()

    // Introducir el mensaje por consola
    printf("Introduzca el texto del mensaje: ");

    scanf_s("%s", texto, 80);

    // Imprimir el mensaje mediante MessageBox

    valor_ret = MessageBox( NULL, texto, titulo,
                          MB_OKCANCEL | MB_ICONEXCLAMATION );

    // Si se ha pulsado Aceptar, indicarlo en consola con un mensaje

    if (valor_ret == IDOK)
        printf("Se ha pulsado Aceptar\n");
}
```

0,5

1

0,5

— Contesta a las siguientes preguntas breves sobre la práctica de procesos:

0,5

Nombre del proceso que controla las sesiones de usuario y que responde a la combinación de pulsaciones de teclado Ctrl-Alt-Supr mostrando la ventana de autenticación de usuarios:

WINLOGON.EXE

Nombre de los campos o miembros del tipo de estructura STARTUPINFO que se utilizan para determinar el tamaño de la consola que se asocia a un proceso puesto en ejecución por `CreateProcess()`:

dwXSize, dwYSize

Uno de los campos o miembros de la estructura STARTUPINFO se utiliza para determinar el color del texto y del fondo de la consola asociada al proceso que se crea, pero para que este campo sea tenido en cuenta por `CreateProcess()`, es necesario inicializar el campo `dwFlags` de la estructura STARTUPINFO con un determinado flag o constante. ¿Cuál es dicho flag?

STARTF_USEFILLATTRIBUTE

A

El objetivo de un programa es mostrar por pantalla los N primeros múltiplos de tres, empezando por el número 0 (el 0 es múltiplo de todos los números). El programa recibe a través de la consola el valor N, entonces reserva y compromete una región de memoria virtual para almacenar en ella los N números. La región comprometida debe tener el tamaño suficiente para almacenar los N números, pero no debe ser mayor de lo estrictamente necesario. Después el programa calcula mediante un bucle los múltiplos de tres, almacenándolos consecutivamente en la región comprometida. Al comienzo de dicha región se almacena el 0, a continuación el 3, después el 6 y así sucesivamente. Una vez que la región ha sido rellena con los N números, entonces el programa los imprime en la consola empezando por el número 0. A continuación se muestra la salida que realiza el programa cuando N = 5.

```
Numero de elementos: 5
```

```
Elemento [0] = 0
Elemento [1] = 3
Elemento [2] = 6
Elemento [3] = 9
Elemento [4] = 12
```

```
Exito en la liberacion
```

Finalmente el programa libera totalmente la región de memoria virtual utilizada.

El listado del programa perfilado mediante comentarios se muestra a continuación. Debes rellenar los huecos existentes en el programa, poniendo atención a los comentarios del mismo. El programa debe realizar una salida por consola idéntica a la indicada en el recuadro anterior.

```
#include <windows.h>
#include <stdio.h>

main()
{
    int *p;    // Para apuntar a la region reservada y comprometida
    int N;    // Numero de elementos a procesar
    int i;    // Auxiliar para bucles

    // Pedir por pantalla el número de elementos a procesar
    printf("Numero de elementos: ");
    scanf_s("%d", &N);
```

```
// Reservar y comprometer region
// El sistema operativo elige la direccion de reserva

p=VirtualAlloc(NULL, 4*N,
               MEM_RESERVE | MEM_COMMIT,
               PAGE_READWRITE);

// Si VirtualAlloc() falla, enviar mensaje a la consola
// y abortar la ejecucion

if ( p == NULL )
{
    printf("Fallo en VirtualAlloc, se aborta la ejecucion\n");
    ExitProcess(0);
}

// Almacenar los multiplos de tres en la memoria

for(i=0; i<N; i++)
    *(p+i)=i*3;

// Imprimir los lementos almacenados

for(i=0; i<N; i++)
    printf("\nElemento [%d] = %d", i, *(p+i));

// Liberar la region comprometida
if ( VirtualFree( p, 0, MEM_RELEASE ) )
    printf("\n\nExito en la liberacion\n");
else
    printf("\n\nFallo en la liberacion\n");
}
```

- En una determinada ejecución de este programa el usuario pide que se generen 10.000 números múltiplos de 3 y *VirtualAlloc()* devuelve la dirección 00370000 (hexadecimal). Determina el rango de direcciones de la página en la que se ubica el número de índice 7000 (¡jojo! No el número 7000, que no es múltiplo de 3, sino el número de índice 7000 en la generación de múltiplos.) (Ejemplo de respuesta: 00370000 – 00378888.)

```
00376000 – 00376FFF
```