

VirtualAlloc()

❑ Objetivo

Reservar o comprometer una región de páginas en el espacio de direcciones del proceso que la llama

❑ Prototipo

```
LPVOID VirtualAlloc(  
    LPVOID lpAddress  
    SIZE_T dwSize  
    DWORD flAllocationType  
    DWORD flProtect );
```

❑ Descripción de los parámetros

lpAddress: Dirección de la región a reservar o comprometer. (Se redondea hacia abajo teniendo en cuenta la granularidad de reserva.)

dwSize: Tamaño de la región a reservar expresada en bytes. (Se redondea hacia arriba hasta un múltiplo del tamaño de página.)

flAllocationType: Tipo de operación a realizar

MEM_RESERVE: Reservar

MEM_COMMIT: Comprometer una región reservada previamente.

MEM_RESERVE | MEM_COMMIT: Reservar y comprometer simultáneamente.

flProtect: Tipo de protección asignada a la región reservada o comprometida (PAGE_NOACCESS, PAGE_READONLY ó PAGE_READWRITE.)

❑ Valor retornado

Dirección de la región reservada o comprometida, o bien NULL si al función falla.



Ejemplo de uso de VirtualAlloc()

□ Enunciado

Hacer una llamada a la función VirtualAlloc() para reservar 8 páginas en el espacio de direcciones de un proceso, a partir de la dirección 40000000, y de modo que las páginas puedan ser leídas y escritas.

Si la función falla, enviar un mensaje a pantalla indicando el fallo y si acierta, enviar un mensaje indicando la dirección a partir de la cual se ha realizado la reserva.

□ Solución

```
main()
{
    void *p;

    p = VirtualAlloc( (void *)0x40000000,
                    8*4096,
                    MEM_RESERVE,
                    PAGE_READWRITE );

    if (p == NULL)
        printf("Fallo en la RESERVA");
    else
        printf("Dir. de RESERVA = %p", p);
}
```



VirtualFree()

❑ Objetivo

Liberar completamente o de compromiso de almacenamiento una región de páginas en el espacio de direcciones del proceso que la llama

❑ Prototipo

```
BOOL VirtualFree(  
    LPVOID lpAddress  
    SIZE_T dwSize  
    DWORD dwFreeType );
```

❑ Descripción de los parámetros

lpAddress: Dirección base de la región de páginas a liberar.

dwSize: Tamaño en bytes de la región a liberar.

dwFreeType: Tipo de liberación a realizar

MEM_DECOMMIT: Elimina el compromiso de almacenamiento de la región. Sin embargo, la región continúa reservada.

MEM_RELEASE: Libera completamente la región, es decir, se elimina el compromiso y la reserva de la región.

NOTA: Si dwFreeType = MEM_RELEASE, entonces dwSize debe ser cero. En caso contrario, la función falla.

❑ Valor retornado

Diferente de cero si la función tiene éxito, o bien cero en el caso contrario.



Ejemplo de uso de VirtualFree()

□ Enunciado

Reservar y comprometer 4 páginas a partir de la dirección 0x01000000 y después liberarlas completamente. Enviar un mensaje a pantalla indicando si la liberación ha tenido éxito o no.

□ Solución

```
main()
{
    void *p;

    p = VirtualAlloc( (void *)0x01000000,
                    4*4096,
                    MEM_RESERVE,
                    PAGE_READWRITE );

    if ( VirtualFree( p, 0, MEM_RELEASE ) )
        printf("Exito en la liberacion");
    else
        printf("Fallo en la liberacion");
}
```

