

© Copyright 2010 tv <tvaira@free.fr>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License,

Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover.

You can obtain a copy of the GNU General Public License : write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

SOMMAIRE

API Win32.....3

Fichier mappé.....3

API WIN32

L'API Win32 de Microsoft propose différents moyens pour faire communiquer plusieurs processus ou threads.

Les mécanismes fournis sont décrits dans la MSDN dans l'article « Interprocess Communications » :

- <http://msdn.microsoft.com/en-us/library/aa365574%28v=VS.85%29.aspx>

FICHER MAPPÉ

La technique officielle préconisée par Microsoft pour partager de la mémoire est d'utiliser les différentes fonctions pour accéder à un fichier mappé (FileMapping).

Un des processus ou threads doit créer en espace de mémoire partagée avec un fichier mappé en utilisant **CreateFileMapping()**.

Les autres processus ou threads peuvent accéder à la mémoire partagée en l'ouvrant avec la fonction **OpenFileMapping()**.

Pour avoir un pointeur utilisable pour accéder à une zone de la mémoire, il faut utiliser **MapViewOfFile()**. Après utilisation, il faut appeler **UnmapViewOfFile()** pour détacher le handle.

Pour libérer la mémoire, il suffit d'appeler **CloseHandle()** sur le handle obtenu.

Remarque : l'accès simultanée à la mémoire partagée doit être sécurisé par l'utilisation d'un mutex par exemple.

Producteur.c :

```

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

#define NOM_MEM_PARTAGEE "NomMemoirePartagee"
#define TAILLE_MEM 50

int main(void)
{
    HANDLE hMem; // handle pour la gestion de la mémoire
    char* ptrDonnees; // Données de la mémoire partagée
    int i;

    // Création d'une zone de mémoire partagée pour les échanges
    hMem = CreateFileMapping(INVALID_HANDLE_VALUE, NULL, PAGE_READWRITE,
    0, TAILLE_MEM, NOM_MEM_PARTAGEE);

    // Création d'une vue dans cette mémoire partagée des TAILLE_MEM
    premiers caractères
    ptrDonnees = (char*)MapViewOfFile(hMem, FILE_MAP_ALL_ACCESS, 0, 0,
    TAILLE_MEM);

    printf("Producteur : ecrit les lettres de l'alphabet dans la memoire
    partagee\n");
    for(i=0;i<26;i++)
        *(ptrDonnees+i) = 'A' + i;

    printf("Attente ... Appuyez sur une touche pour quitter\n");
    getchar();

    // Libération de la mémoire partagée
    UnmapViewOfFile(ptrDonnees);

    CloseHandle(hMem);

    return 0;
}

```

consommateur.c :

```

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

#define NOM_MEM_PARTAGEE "NomMemoirePartagee"
#define TAILLE_MEM 50

int main(void)
{
    HANDLE hMem; // handle pour la gestion de la mémoire
    char* ptrDonnees; // Données de la mémoire partagée
    int i;

```

```

printf("Attente ... Appuyez sur une touche pour continuer\n");
getchar();

// Création d'une zone de mémoire partagée pour les échanges
hMem = OpenFileMapping(FILE_MAP_ALL_ACCESS, FALSE, NOM_MEM_PARTAGEE);

// Création d'une vue dans cette mémoire partagée des 20 premiers
caractères
ptrDonnees = (char*)MapViewOfFile(hMem, FILE_MAP_ALL_ACCESS, 0, 0, 26
);

printf("Consommateur : lit la memoire partagee\n");
for(i=0;i<26;i++)
    printf("%c ", *(ptrDonnees+i));

// Libération de la mémoire partagée
UnmapViewOfFile(ptrDonnees);

CloseHandle(hMem);

return 0;
}

```

Ce qui donne l'exécution suivante :

```

Administrateur : C:\Windows\system32\cmd.exe
C:\Dev-Cpp\Examples\mapping>
C:\Dev-Cpp\Examples\mapping>producteur
Producteur : ecrit les lettres de l'alphabet dans la memoire partagee
Attente ... Appuyez sur une touche pour quitter
C:\Dev-Cpp\Examples\mapping>_

Administrateur : C:\Windows\system32\cmd.exe
C:\Dev-Cpp\Examples\mapping>
C:\Dev-Cpp\Examples\mapping>consommateur
Attente ... Appuyez sur une touche pour continuer

Consommateur : lit la memoire partagee
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
C:\Dev-Cpp\Examples\mapping>

```