

Les Systèmes d'Exploitation

Les mémoires virtuelles

Gérard Padiou

Département Informatique et Mathématiques appliquées
ENSEEIH

Octobre 2008



plan

- 1 Mémoires virtuelles et fichiers
- 2 Système Windows/NT
- 3 Système Linux



Intégration des fichiers dans l'espace virtuel

Idée

- Les images binaires de code exécutables sont couplées en mémoire virtuelle.
- **Pourquoi** ne pas coupler aussi les fichiers de données dans l'espace virtuel ?
- **Avantage** : permet d'accéder au fichier par des simples instructions de lecture/écriture en mémoire centrale.

Un précurseur : Système MULTICS

- Mise en œuvre dans le système MULTICS (conçu en 1965) :
- Tout fichier accédé est d'abord couplé dans un segment de mémoire virtuelle ;
- Autre idée novatrice : édition de liens dynamique ;



Intégration des fichiers dans l'espace virtuel

Systemes actuels

- Utilisée par le noyau pour coupler :
 - l'image binaire exécutable en mémoire virtuelle
 - l'image binaire d'une bibliothèque (exemple : .dll sous Windows)
- Disponible pour les programmes applicatifs par appel explicite de primitives de couplage.



Mémoire virtuelle sous Windows-NT

Structure

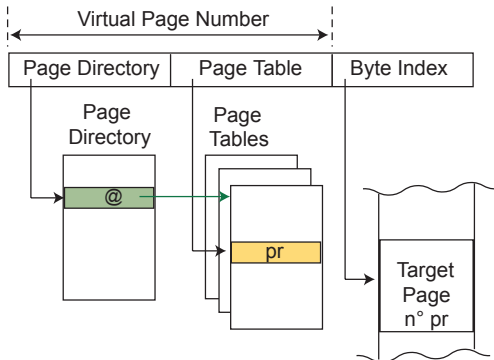
- Mémoire à deux niveaux : segmentation et pagination
- Adresses virtuelles sur 32 bits \Rightarrow 4G octets adressables
- Découpage logique :
 - Intelx86 : 1024 segments de 1024 pages de 4K
 - Digital Alpha : 256 segments de 2048 pages de 8K

Règles de couplage

- Un zone interdite : 0x0 à 0xFFFF (64Ko de début)
- La zone programme applicatif : 0x1000 à 0x7FFEFFFF (2G - 128K)
- La zone Noyau : 0x80000000 à 0xFFFFFFFF (2G)

Mémoire virtuelle sous Windows-NT

Dans la terminologie Windows-NT



Remarque : Page Directory à adresse fixe.



Mémoire virtuelle sous Windows-NT

Programmation du couplage

Couplage en deux étapes

- La réservation/libération d'un espace d'adressage (région) ;
- Le couplage/découplage proprement dit d'un contenu.

Primitives

- `VirtualAlloc` : réservation ou/et couplage ;
- `VirtualFree` : libération ou/et couplage ;
- `VirtualLock` : verrouillage en mémoire centrale ;
- `VirtualUnlock` : déverrouillage.



Mémoire virtuelle sous Windows-NT

Réservation/libération d'une région et/ou (dé)couplage de pages

```
LPVOID VirtualAlloc (  
    LPVOID lpAddress, // adresse de base  
    DWORD dwSize, // taille en octets  
    DWORD flAllocationType, // type d'opération  
    DWORD flProtect, // attributs de protection  
);  
LPVOID VirtualFree (  
    LPVOID lpAddress, // adresse de base  
    DWORD dwSize, // taille en octets  
    DWORD dwFreeType, // type d'opération  
);
```

avec :

```
flAllocationType = {MEM_RESERVE, MEM_COMMIT, MEM_RESET}  
dwFreeType={MEM_RELEASE, MEM_DECOMMIT}
```



Le couplage de fichiers

Rappels : Idée utilisée pour charger :

- au lancement, l'images binaire exécutable (.exe) ;
- dynamiquement les bibliothèques (.dll).

Exemple

Lors du lancement d'un processus :

- 1 Couplage du binaire exécutable à partir de l'adresse 0x400000 ;
- 2 Couplage des .dll's connus.



Le couplage de fichiers

Les étapes

- 1 Créer ou ouvrir le fichier : `CreateFile`, `OpenFile` ;
- 2 Définir le couplage du fichier : `CreateFileMapping` ;
- 3 Associer une région au fichier : `MapViewOfFile`.



Le couplage de fichiers


Les primitives

Le couplage de contenu

```
HANDLE CreateFileMapping (  
    HANDLE hFile, // le fichier  
    LPSECURITY_ATTRIBUTE lpFileMappingAttributes,  
    DWORD flProtect, // attributs de protection  
    DWORD dwMaximumSizeHigh, // forts poids  
    DWORD dwMaximumSizeLow, // faibles poids  
    LPCTSTR lpName  
);
```

avec :

```
flProtect = {PAGE_READONLY, PAGE_READWRITE, PAGE_WRITECOPY}
```

 PAGE_WRITECOPY : voir FILE_MAP_COPY



Le couplage de fichiers

Les primitives

Réserver/Libérer une région en mémoire virtuelle

```
LPVOID MapViewOfFile (  
    HANDLE hFileMappingObject, // l'objet couplage  
    DWORD dwDesiredAccess,  
    DWORD dwFileOffsetHigh, // forts poids  
    DWORD dwFileOffsetLow, // faibles poids  
    DWORD dwNumberOfBytesToMap  
);  
BOOL UnmapViewOfFile(  
    LPCVOID lpBaseAddress  
);
```

avec :

```
dwDesiredAccess =  
    {FILE_MAP_WRITE, FILE_MAP_READ, FILE_MAP_COPY}
```



Système Linux (segmentation + pagination)

La bible : Understanding the Linux Virtual Memory Manager, *Mel Gorman*^a

^awww.informit.com/content/images/0131453483/downloads/gorman_book.pdf

- Trois niveaux de pagination (au plus) ;
- Pagination à la demande ;
- Primitives de couplage de fichiers avec différents modes :
 - région au contenu partageable ;
 - région au contenu privé ;
 - pas de préservation d'espace de swap ;
 - couplage d'un contenu à la demande (pad de fichier) ;



Système Linux

Les primitives

- Réservation d'une région avec couplage de contenu

```
void *mmap(void *addr, size_t len, int prot,  
           int flags, int fildes, off_t off);
```

- Libération d'une région avec découplage

```
int munmap(void *addr, size_t len);
```

- Modification de la protection d'une région

```
int mprotect(void *addr, size_t len, int prot);
```



Conclusion

- Mécanisme indispensable de protection ;
- Gestion complexe mais maîtrisée ;
- Utilisation des portables aux super-calculateurs ;
- Pagination à la demande utile mais avec ses limites :
 - ☞ écroulement
- Devenir et efficacité avec les multi-cores ?

