



CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE



Rappel sur le fonctionnement de la mémoire

Denis PUGNÈRE – IN2P3/IPNL

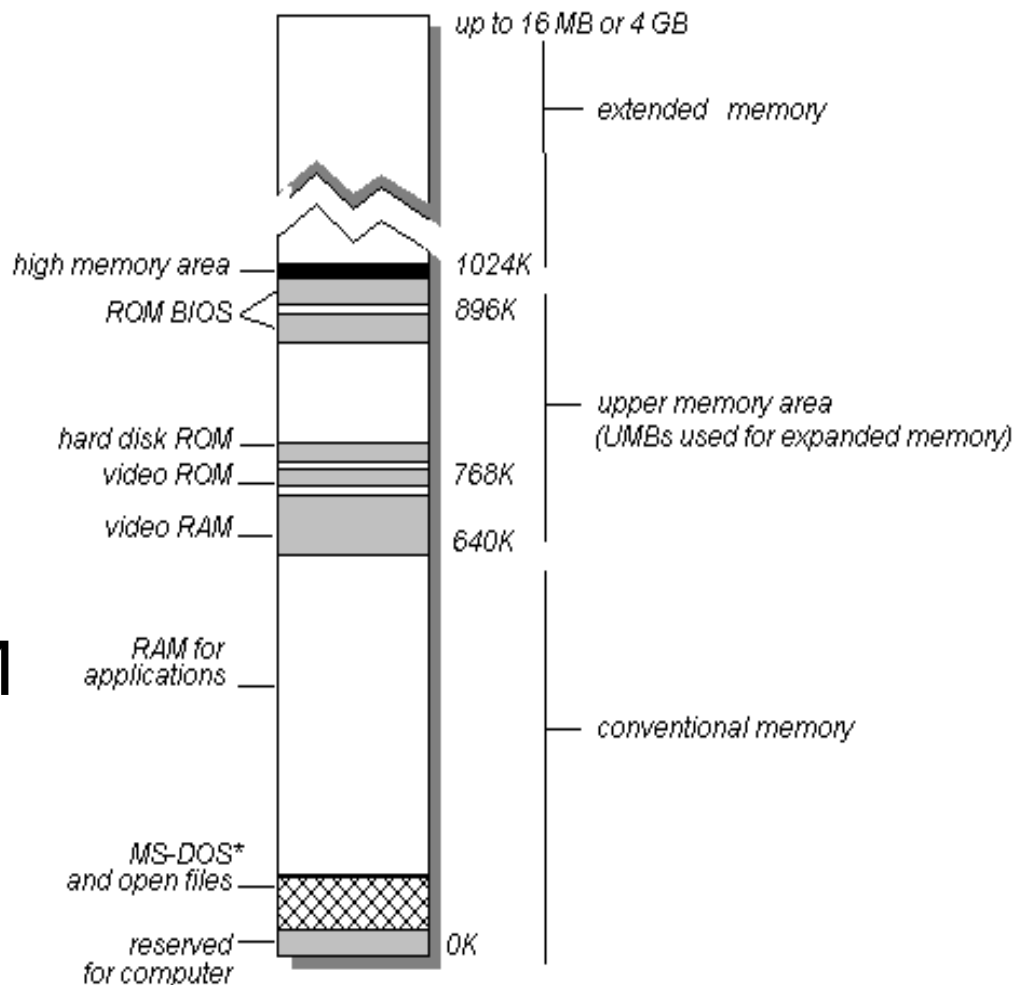
d.pugnere@ipnl.in2p3.fr

A3IMP - La Grande Motte - 24-26/09/2007



Mémoire système sur plate-forme x86

- 4 barrettes RAM de 1Go \neq 4Go
- Certaines zones sont réservées par le matériel (PCI, AGP...)
- Certaines zones sont en lecture seule (ROM BIOS...)



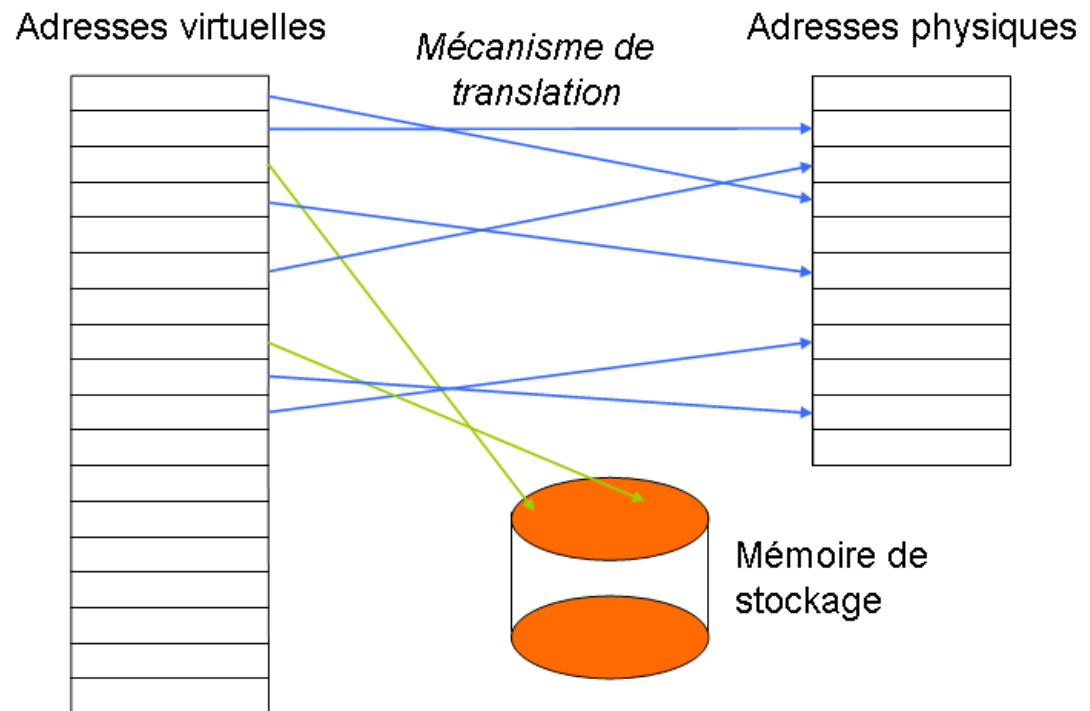
* If MS-DOS is not loaded in high memory area



Différents types de mémoire

- Mémoire physique, mémoire virtuelle
- Pagination : Mécanisme qui étend la capacité de mémoire en utilisant le disque comme mémoire volatile

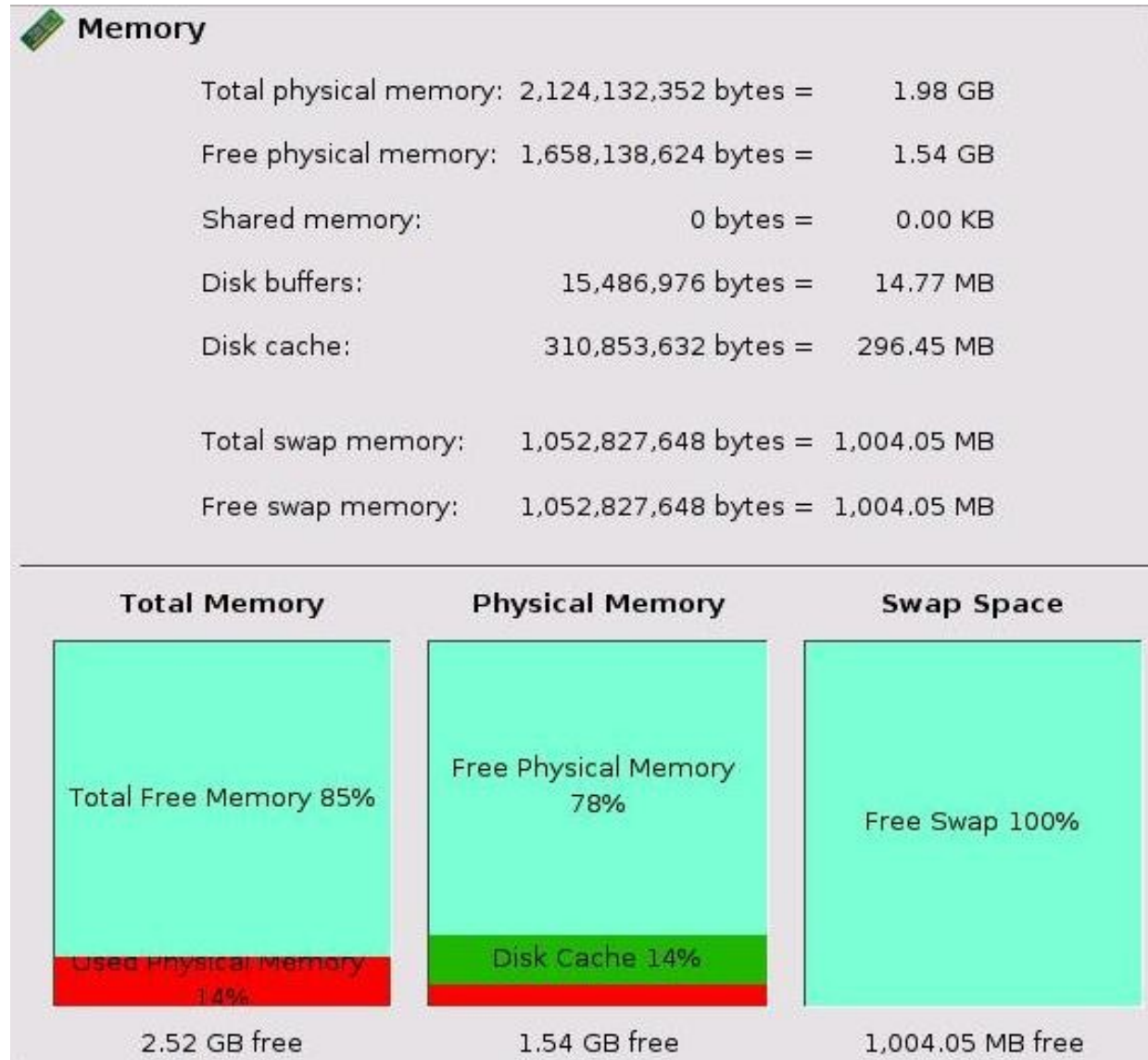
Mémoire virtuelle =
RAM + espace SWAP





Buffers RAM gérés par le noyau

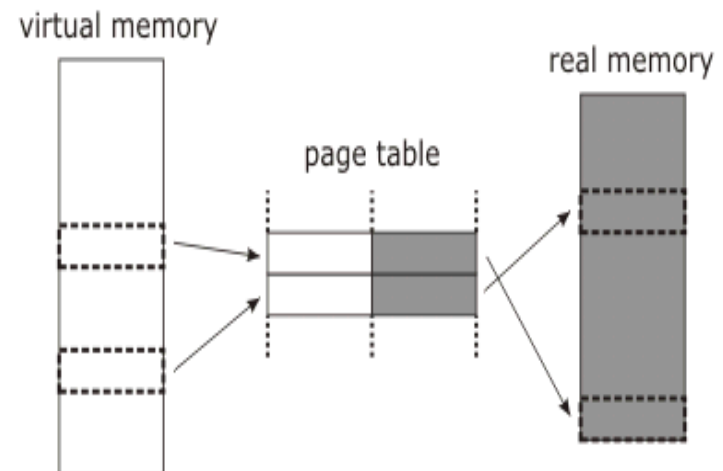
- Cache
- Free
- Dirty
- Mapped





Mémoire virtuelle et pagination

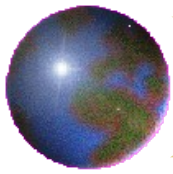
- Formée de zones de même taille, appelées pages (généralement 4Ko)
 - Traduction d'adresses physiques en adresses virtuelle par la MMU
 - Utilisation d'une table de translation :
adresses physiques \Leftrightarrow adresses virtuelles





Mécanisme de pagination

- En cas d'occupation à 100% de la RAM : Différents algorithmes pour
 - Le choix de la page en RAM pour la basculer en SWAP et libérer de la RAM pour un processus en cours d'exécution :
 - LRU : Least recently used, soit « la moins récemment utilisée »,
 - FIFO : First in, first out ou « Premier rentré, premier sorti »)
 - Charger en RAM une page précédemment SWAPée



Traces sur le SWAP

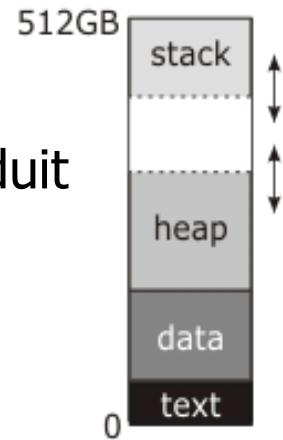
- **Conséquences**

- Pas de trace en SWAP si le système ne SWAPe pas
- Le SWAP peut contenir des informations très anciennes,
- Les informations inscrites dans le SWAP sont rémanentes (même après plusieurs arrêts-démarrages)
- Certains systèmes chiffrent le SWAP
- Le mode « veille prolongée » peut utiliser le SWAP pour sauvegarder le contenu de la RAM avant la veille



Allocation mémoire d'un processus

- Statique (segment text, data ou bss)
 - Espace réservé à la compilation dans le fichier binaire produit
 - Accessible au chargement du programme, juste avant son exécution
- Dynamique sur la pile (segment stack)
 - « Variables lexicales » allouées lors de l'entrée dans une fonction, et désallouées automatiquement lors de la sortie de la fonction
- Dynamique sur le tas (segment heap)
 - Allouée à des moments arbitraires dans l'exécution du programme
 - Sous la responsabilité du programmeur
 - Exemples : C (malloc, free), C++ (new, delete)





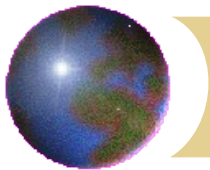
Différents buffers d'un exécutable

Exécutable statique :

```
$ file dd-static
  dd: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
  for GNU/Linux 2.2.5, statically linked, stripped

$ ll dd-static
-rwxr-xr-x  1 pugnere  infor      453300 Sep 18 15:12 dd

$ size dd-static
   text    data    bss     dec     hex filename
 447461   3400   6152  457013  6f935 dd
```



Exécutable dynamique :

```
$ file /bin/dd
/bin/dd: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
for GNU/Linux 2.2.5, dynamically linked (uses shared libs), stripped

$ ll /bin/dd
-rwxr-xr-x    1 root    root          30804 Jun 14 12:09 /bin/dd

$ size /bin/dd
   text    data     bss      dec       hex filename
 28421     760     648   29829   7485 /bin/dd

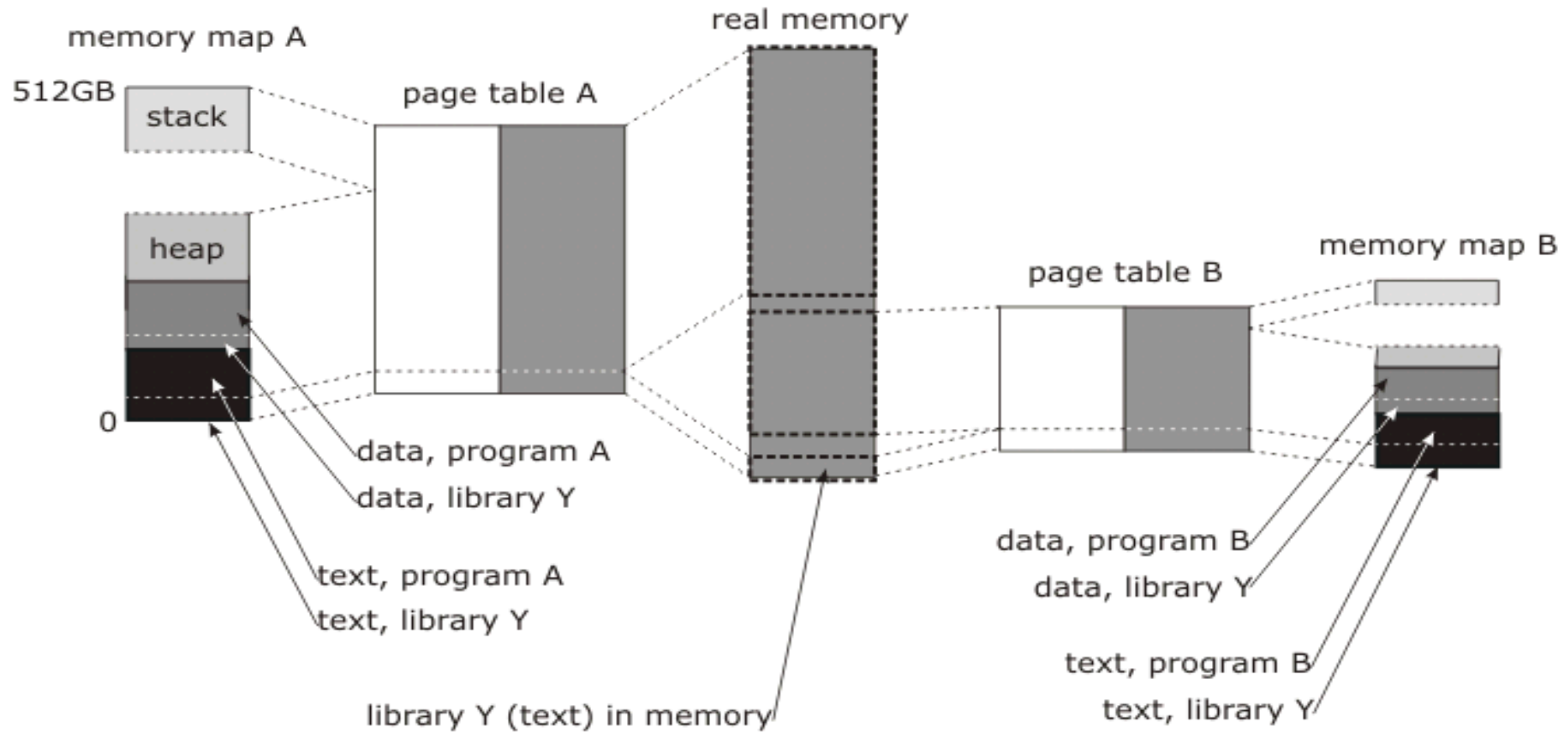
$ ldd /bin/dd
libc.so.6 => /lib/tls/libc.so.6 (0x001d7000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00620000)

$ size /lib/tls/libc.so.6
   text    data     bss      dec       hex filename
1253956  11076   10988 1276020 137874 /lib/tls/libc.so.6

$ size /lib/ld-linux.so.2
   text    data     bss      dec       hex filename
 85381     1348     432   87161   15479 /lib/ld-linux.so.2
```



Processus différents, bibliothèques identiques



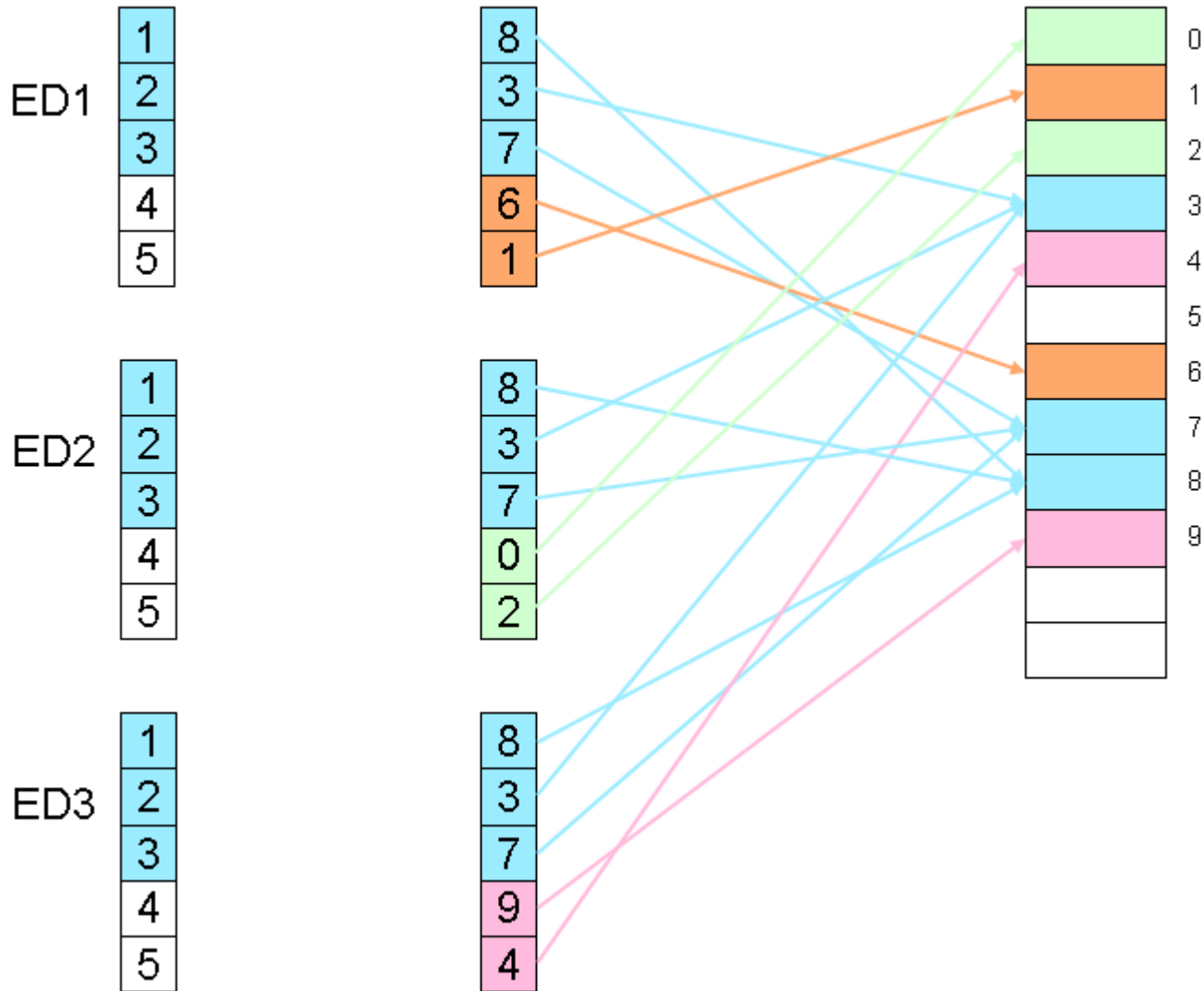


Processus identiques

Processus

Table des pages

Mémoire physique





Traces en mémoire RAM

- **Conséquences**

- Un même exécutable peut être exécuté plusieurs fois en même temps, son code est présent une seule fois,
- Les bibliothèques dynamiques sont chargées une seule fois,
- Une partie de la mémoire non ré-utilisée contient des traces d'anciennes allocations (code ou données),
- Les binaires statiques chargent plusieurs fois les mêmes codes.