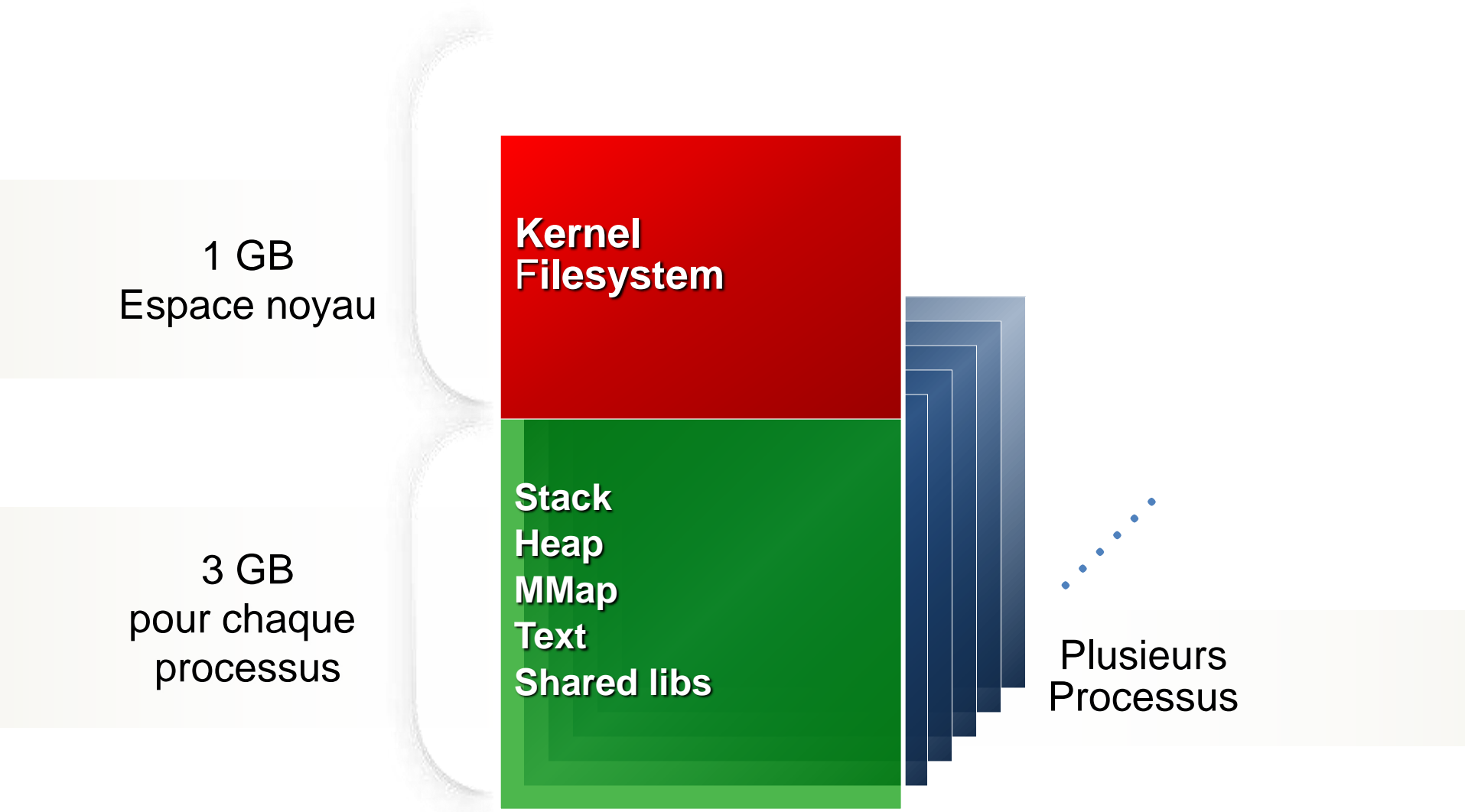


Développement d'applications natives

Mémoire virtuelle

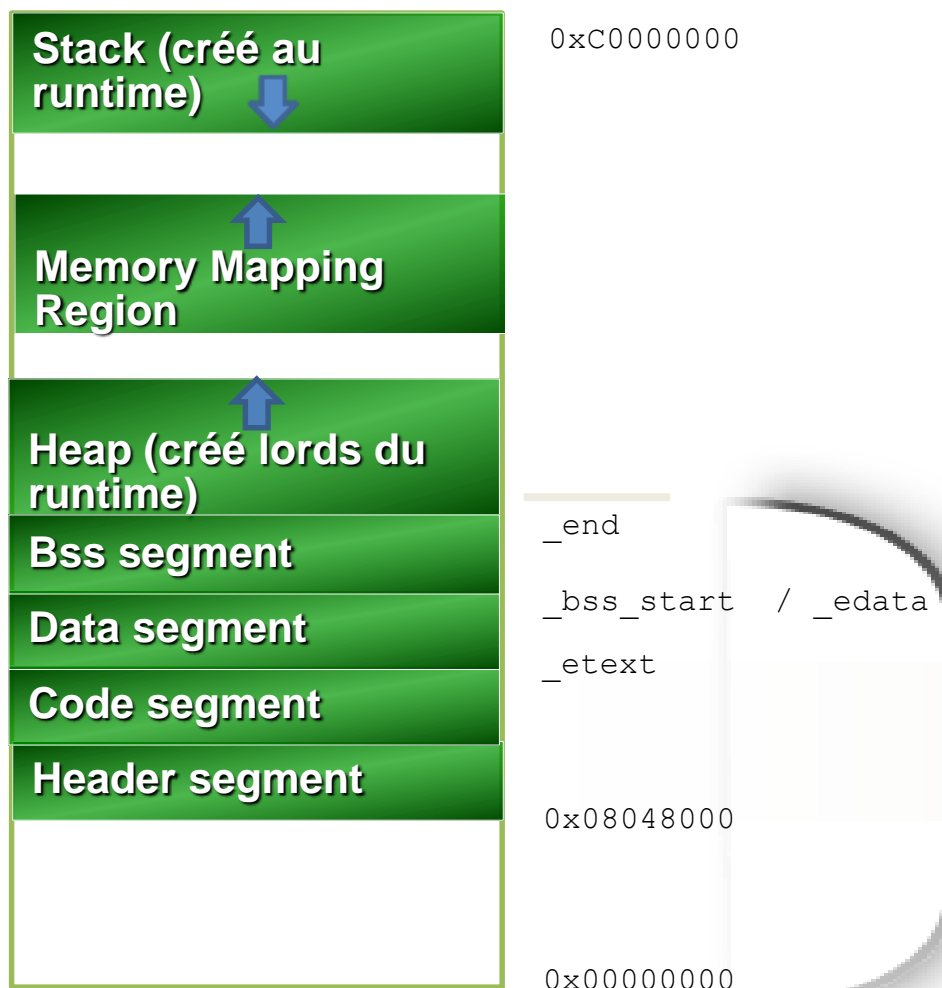
Mémoire virtuelle sous Linux 2.6



Espace d'adressage utilisateur

Espace
d'adressage
utilisateur
3 Gigabytes

Chaque
processus
possède son
propre mapping



Chargé depuis
le fichier
exécutable

Statut de la mémoire

/proc/meminfo

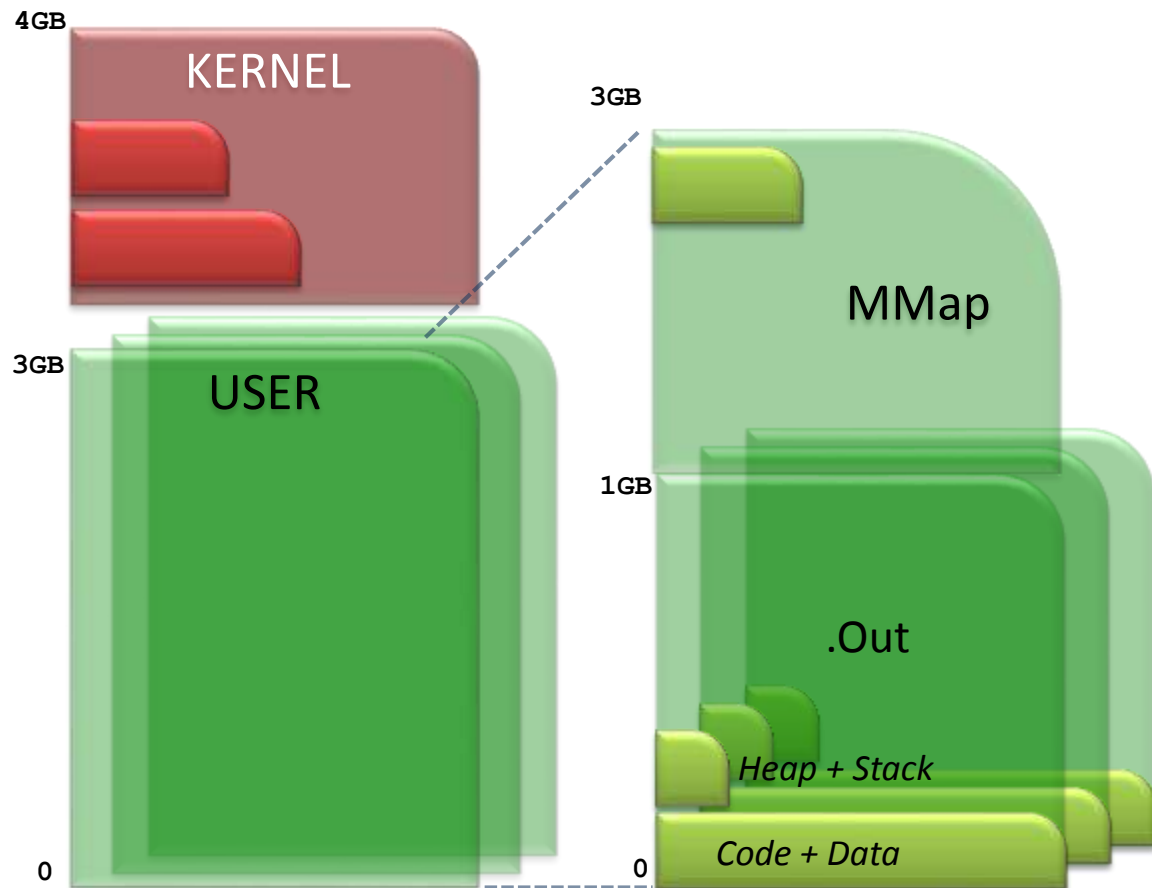
Total Physical memory

Available Physique memory

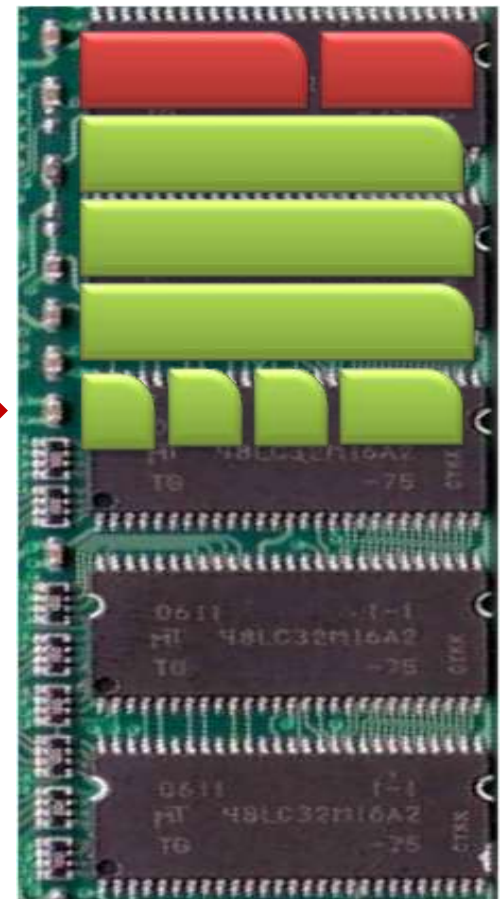
```
theoris@ubuntu: ~  
[root@X-Linux]:~ # cat /proc/meminfo  
MemTotal:      516068 kB  
MemFree:       492780 kB  
Buffers:        128 kB  
Cached:        2776 kB  
SwapCached:      0 kB  
Active:        1220 kB  
Inactive:      2080 kB  
Active(anon):   412 kB  
Inactive(anon): 0 kB  
Active(file):   808 kB  
Inactive(file): 2080 kB  
Unevictable:    0 kB  
Mlocked:        0 kB  
SwapTotal:      0 kB  
SwapFree:       0 kB  
Dirty:          0 kB  
Writeback:      0 kB  
AnonPages:      432 kB  
Mapped:         556 kB  
Slab:          1856 kB  
SReclaimable:   140 kB  
SUnreclaim:    1716 kB  
PageTables:     112 kB  
NFS_Unstable:   0 kB  
Bounce:         0 kB  
WritebackTmp:   0 kB  
CommitLimit:   258032 kB  
Committed_AS:  2476 kB  
VmallocTotal:  516012 kB  
VmallocUsed:    33064 kB  
VmallocChunk:  482728 kB  
DirectMap4k:   524288 kB  
DirectMap4M:   0 kB  
[root@X-Linux]:~ #
```

Occupation mémoire

Espace virtuel



Espace physique
(RAM)



Mémoire virtuelle : accès bas niveau

En-tête: `sys/mman.h`

```
void *mmap(void *addr, size_t length, int prot,  
           int flags, int fd, off_t offset);
```

- **addr**: adresse de départ. Si 0, le système choisit la zone libre en commençant par l'adresse la plus haute.
- **length**: longueur de l'espace demandé
- **prot**: Droit d'accès (PROT_NONE, PROT_EXEC, PROT_READ et PROT_WRITE).
- **flags**: Attributs MAP_SHARED (Partage de la projection) ou MAP_PRIVATE (projection privée). MAP_NORESERVE (ne pas réserver d'espace de swap pour cette projection), MAP_LOCKED (verrouille la page projetée en mémoire), MAP_GROWSDOWN (la projection doit s'étendre vers le bas de la mémoire). MAP_32BIT (faire la projection dans les premiers 2 Go). MAP_ANONYMOUS permet de faire de l'allocation sur le tas.
- **fd**: Descripteur de fichier.
- **offset**: Multiple de la taille d'une page. Normalement à 0

```
int msync(void *addr, size_t length, int flags);
```

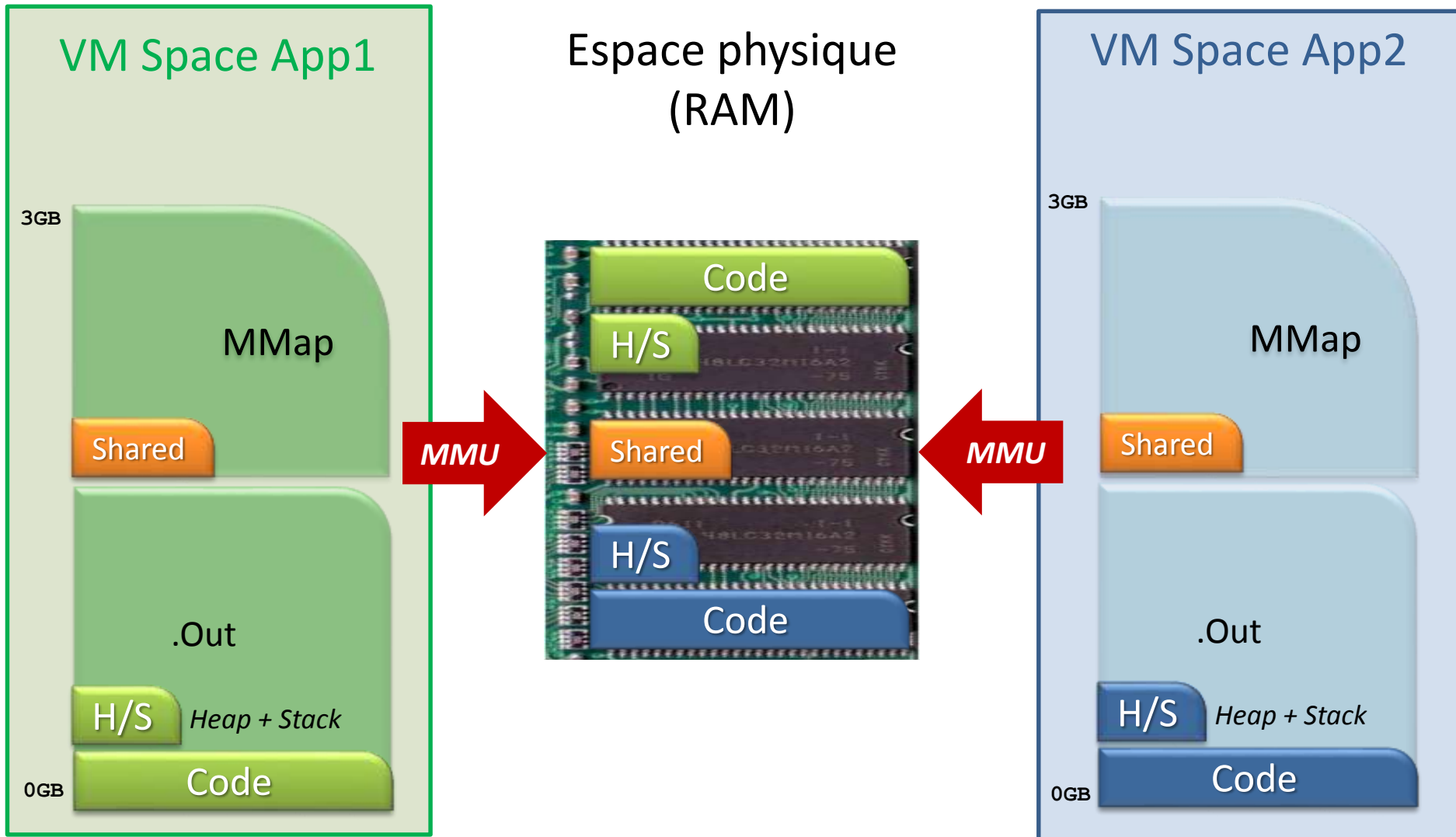
synchronisation du fichier ou projection en mémoire .

- **Flags**: MS_ASYNC (Mise à jour planifiée) ou MS_SYNC (mise à jour immédiate). MS_INVALIDATE (mise à jour des autres projections).

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

-> suppression en mémoire.

Concept de mémoire partagée



Allocation en mémoire partagée

- Création ou accès à l'espace d'adressage du processus

En-têtes: `sys/mman.h`, `sys/stat.h` et `fcntl.h` Utilisation de « -lrt » lors de l'édition de liens.

```
int shm_open(const char *name, int oflag, mode_t mode);  
int shm_unlink(const char *nom);
```

- Réserve de l'espace mémoire à l'espace d'adressage partagé

En-têtes: `sys/types.h` et `unistd.h`

```
int ftruncate(int fd, off_t length);
```

- Etablir et supprimer une projection en mémoire des fichiers ou des périphériques

Utilisation des fonctions `mmap` et `munmap` avec le flag:

`MAP_SHARED`