# Systèmes d’exploitation avancés

## Liens

Les liens durs : sont représentés par des pairs nom/inode pointant au même endroit qu’une autre paire. Ce type de lien n’est pas possible sur un répertoire ou entre deux périphériques différents.  
Les liens symboliques : sont représentés par un nouveau fichier de type « lien » qui lui pointera vers un autre fichier.

## Structure d’un disque

Un disque commence par un superbloc contenant la place utilisée par les données et les inodes.  
Au début du disque, juste après le superbloc, se trouvera les inodes.  
Après tout ceci arrivera finalement les données réelles des fichiers.

Dans les inodes se trouvent des pointeurs vers les datablocks. Pour aider à gagner de la place pour les petits fichiers, on utilise des blocks fichier pour pointer vers d’autres blocks fichiers (qui peuvent eux-mêmes contenir des pointeurs).

## Processus

Un processus permet de sécuriser les programmes entre eux, de les isoler. Le seul programme qui n’est pas un processus est le noyau système !

Un programme est dit **statique** car c’est un script qui a été compilé et qui ne changera pas dans le temps.  
Un processus est dit **dynamique** car il possède des attributs qui vont changer dans le temps (priorité, environnement d’exécution, dossier de travail, fichiers ouverts, statistiques, etc…)

Toutes les données d’un processus (informations) sont stockées dans un tableau des processus, se trouvant à l’extérieur de l’espace utilisateur (ne permettant pas à un processus malicieux de modifier ces données).

Ce système va permettre de réaliser du multi-tâche en enchaînant les processus un par un, pour réaliser des petites tâches.

Quand un processeur affiche un « s » , il change lors de l’exécution l’utilisateur qui le lance par le propriétaire du programme. (Si je lance « ps » avec l’utilisateur *Thomas*, lors de l’exécution, le programme sera automatiquement lancé comme s’il avait été lancé par *root*, permettant d’afficher les programmes des autres utilisateurs)

Topologie d’un processus :

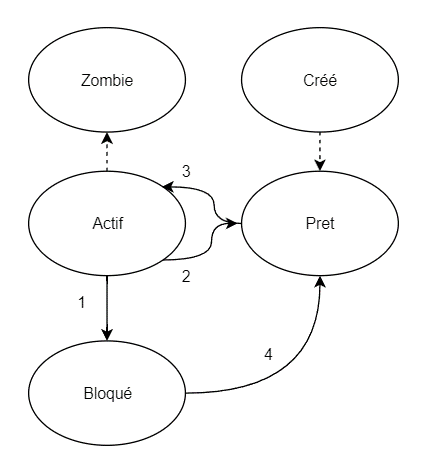
|  |
| --- |
| Initialisation |
| Code |
| Pile (stack) du programme (spécifique à chaque fonction) |
| Données statiques et globales |
| **Tas** (dynamique) |

Il faut faire attention, les pointeurs ne fonctionne **que** dans l’espace d’un processus, mais il n’est pas possible de les passer d’un processus à un autre.

Il est possible d’augmenter ou de réduire la taille du tas avec les fonctions **brk** et **sbrk**.

Un système de libération de la mémoire sur le disque dur se mets en place une fois la RAM pleine. Seulement appliqué sur les programmes qui ne sont pas actifs depuis un certain temps.

Cycle des processus



A chaque appel système, le processus passe dans l’état « Bloqué », dans l’attente de ressources par exemple. Cela permet d’éviter l’exécution du processus alors qu’il est dans l’attente de ressources.

Pour dupliquer un programme, on peut utiliser la fonction « fork ». Pour savoir si on est le fils ou le père, on peut regarder le retour de la fonction fork (-1 : erreur, 0 : fils, X : PID de l’enfant).