

Les processus

- c'est un programme exécution, (un espace mémoire, des ressources)
- C'est un ensemble d'instructions.
- Le processus se considère seul au monde !
- Accès à un espace mémoire constant.

à l'inverse l'OS doit tout savoir, il vérifie la sécurité et agit selon les tentatives. L'OS choisit qui travaille, donne de la mémoire et l'accès aux ressources.

Mono-tâche :

- un système qui peut faire un seul processus à la fois
- pas de problème de gestion de mémoire
- pas de conflit de ressources.

Multi-tâche :

- Mise en commun, partage des ressources.
- Gestion de la sécurité.

Processus VS Thread

Le flot de contrôle du processus:

- déroulement du programme.
- progression de la liste d'instructions.

Processus : un seul flot de contrôle.

Thread (plus petit flot de contrôle dans un processus):

- plusieurs flots de contrôle dans le même processus.
- Conflits possibles.

Sous Linux

Chaque processus est identifié par :

- Un ID **PID**
 - accès avec `echo $$`.
 - 0 c'est le PID du système.
- L'identité du propriétaire.
- Le numéro du processus père **PPID**.

Plusieurs étapes d'un processus :

- En exécution (élu).
- En attente d'exécution (prêt).
- En attente de ressource (bloqué).

`ps` permet de voir les processus en cours.

`ps aux` permet de voir les utilisateurs associés.

`ps axms` permet de voir les informations sur les threads.

Les signaux

Les processus peuvent communiquer entre eux via des signaux appelés signaux, ils sont :

- asynchrone
- nombre limité
- réactions prédéfinies

Arrêter un processus le signal **SIGNIT**.

`trap -1` pour connaître les différents signaux.

`wait PID` permet dans un programme à un processus d'attendre que son fils soit terminé.