

Introduction au noyau MUTEKP

Figure

Un programme en exécution peut être décrit par trois segments mémoire:

- Code : contient les instructions constituant le code du programme.
- Données : contient les variables globales qui sont des données initialisées ou non initialisées que le code du programme manipule.
- Pile : zone mémoire utilisée par le programme, lors de l'appel à une fonction, pour y stocker des données temporaire tel que paramètre de la fonction appelée et ses variables locales.

Donc pour pouvoir charger un programme pour l'exécuter, il faut créer au moins ces trois segments.

L'idée d'un thread est de pouvoir partager les deux segments code et données et de créer un segment de pile propre à chaque thread.

Ainsi un thread est un flux d'exécution d'un programme où chaque thread possède :

- Un contexte d'exécution (un compteur ordinal, un pointeur de pile et d'autres registres de travail du processeur).
- Une pile.

Quelques avantages :

- Création et gestion plus rapide (par rapport aux processus).
- Partage des ressources.
- Communication entre les threads est plus simple via la mémoire (les variables globales)
- Déploiement plus efficace de l'application sur des architectures multi-processeurs.