

TP ORDONNANCEUR LINUX

Objectif : Mettre en évidence le travail de l'ordonnanceur Linux dans ces différents modes de fonctionnement.

Matériel : PC Linux ou RPI Raspbian

Pré-requis : Cours sur les processus et les ordonnanceurs.

Au préalable :

Utilisez le programme d'exemple (sched_exemple.c) pour vérifier que votre Linux autorise les utilisateurs à passer en mode SCHED-RR ou SCHED_FIFO. Testez ce programme en tant que *su* (utilisateur root) également.

En cas d'erreur, procédez à une mise à jour du noyau (kernel) Linux avec Yast, et modifiez le fichier `/etc/security/limits.conf` en y ajoutant la ligne :

```
*      -      rtprio      100
```

Travail :

1. Déterminez le nombre de cœurs CPU dont dispose votre Linux (voir avec la commande **top**, **F** pour la liste des champs, **P** pour afficher les processeurs).

Créer un programme qui lance plusieurs processus identiques. Ex : pour 4 cœurs, il faudra lancer au moins 8 processus. Le processus à lancer correspond au programme « travail.c » fourni. Le lancement se fera par un « `execl` » dans un « `fork` ».

Rappel : Pour lancer un processus en arrière plan en C :

```
if ( fork() == 0 ) {
    // Processus fils
    // ...
    execl("./travail", "travail", NULL);
}
```

2. Modifiez votre programme pour qu'il donne un niveau de priorité différent à chacun des processus. Le programme doit fixer les réglages des processus avant l'« `execl` ». Vous testerez différentes configurations parmi les suivantes :

Processus fils 1	Autres processus
SCHED_OTHER	SCHED_OTHER avec nice(20)
SCHED_FIFO, priorité maxi	SCHED_OTHER
SCHED_RR, priorité 20	SCHED_OTHER
SCHED_RR, priorité 20	SCHED_RR, priorité 80
SCHED_FIFO, priorité 50	SCHED_RR, priorité 99
... autres combinaisons de votre choix.	

Pour chaque essai, observez sur une fenêtre Terminal le comportement des processus fils avec la commande **top**, notamment la colonne PRI et la colonne TIME+. Vous pourrez compléter ces informations avec une commande de type « `ps al` », pour avoir le numéro exact de la priorité du processus.

Votre programme principal devra afficher également le quantum de temps alloué pour les processus.

Vous utiliserez les primitives C suivantes (en vous aidant du manuel en ligne ou web) :

```
sched_getscheduler(); // Obtenir le type d'ordonnancement d'un proc
sched_getparam();     // remplissage d'une structure sched_param
sched_get_priority_max(); // Obtenir priorité statique max
sched_setscheduler(); // Réglage du type d'ordonnancement et de la priorité
sched_rr_get_interval(); // Obtenir le quantum pour un processus
nice(20);              // ajuster la priorité d'un processus SCHED_OTHER
```