

Les threads LINUX (processus de poids légers)

dernière mise à jour le 12 février 2014

Description des threads

Les threads de LINUX sont une implantation normalisée des MIT-threads. Ces threads sont gérés à la fois par le système et par une librairie au niveau utilisateur. Pour créer des threads, LINUX applique la stratégie du *One-For-One* (une entrée dans la table des processus pour chaque thread).

`pthread_create(thread, attribut, routine, argument)` Création d'un thread. Le nouveau flot d'exécution démarre en se branchant à la routine spécifiée. Cette routine reçoit l'argument prévu.

`pthread_exit(résultat)` Suicide d'un thread.

`pthread_join(thread, résultat)` Attendre la terminaison d'un autre thread.

`pthread_kill(thread, nu_du_signal)` Envoyer un signal (UNIX) à un thread. C'est un moyen dur pour tuer un thread. Il existe des méthodes plus conviviales.

`sched_yield()` Abandonner la CPU pour la donner à un autre thread (ou un autre processus). Attention : il n'existe pas de préemption de la CPU à l'intérieur des threads d'un même processus. En clair, si un thread garde la CPU, les autres threads ne vont pas s'exécuter. Cette routine permet de programmer un partage équitable de la CPU entre threads coopératifs.

Bien entendu, une aide plus complète peut être obtenue sur chaque fonction en utilisant le manuel UNIX :

```
man nom_de_la_fonction
```

Un exemple d'utilisation des threads

Cet exemple est la traduction d'un exercice de TP avec un thread qui lit des caractères au clavier et les passe à un autre thread qui se charge de les afficher. Il faut noter que le thread principal (le père) se charge de la création de ses fils et de l'attente de leur mort. Cette disparition est programmée à l'arrivée du caractère "F".

```

#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

volatile char theChar = '\0';
volatile char afficher = 0;

void* lire (void* name)
{
    do {
        while (afficher == 1) ; /* attendre mon tour */
        theChar = getchar();
        afficher = 1; /* donner le tour */
    } while (theChar != 'F');
    return NULL;
}

void* affichage (void* name)
{
    int cpt = 0;
    do {
        while (afficher == 0) cpt++; /* attendre */
        printf("cpt = %d, car = %c\n", cpt, theChar);
        afficher = 0; /* donner le tour */
    }
    while (theChar != 'F');
    return NULL;
}

int main (void)
{
    pthread_t filsA, filsB;

    if (pthread_create(&filsA, NULL, affichage, "AA")) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }
    if (pthread_create(&filsB, NULL, lire, "BB")) {
        perror("pthread_create");
        exit(EXIT_FAILURE);
    }

    if (pthread_join(filsA, NULL))
        perror("pthread_join");

    if (pthread_join(filsB, NULL))
        perror("pthread_join");

    printf("Fin du pere\n") ;
    return (EXIT_SUCCESS);
}

```

Pour compiler et effectuer l'édition de liens vous devez utiliser la ligne suivante :

```
cc votre_programme.c -lpthread -o votre_programme
```

c'est la fin !