

Module Système Exploitation II

Nom : LAHCEN

Prénom : KAZAZ

N apogée : 1904081

Rapport Langage MultiThread

I. Introduction

Le modèle de processus décrit au chapitre précédent est un programme qui s'exécute selon un chemin unique avec un seul compteur ordinal. On dit qu'il a un flot de contrôle unique ou un seul thread. De nombreux systèmes d'exploitation modernes offrent la possibilité d'associer à un même processus plusieurs chemins d'exécution ou multithread (figure 4.1). Ils permettent ainsi l'exécution simultanée des parties d'un même processus. Chaque partie correspond à un chemin d'exécution du processus. Le processus est vu comme étant un ensemble de ressources (code exécutable, segments de données, de fichiers, de périphériques, etc.) que ces parties appelées flots de contrôle ou processus légers (threads en anglais) partagent. Chaque flot de contrôle (thread) a cependant, en plus des ressources communes, sa propre zone de données ou de variables locales, sa propre pile d'exécution, ses propres registres et son propre compteur ordinal.

II. Définition

LES PROCESSUS et LES THREADS sont des séquences d'exécution indépendantes. La principale différence est la suivante : Les threads (du même processus) s'exécutent dans un espace mémoire partagé, tandis que les processus s'exécutent dans des espaces mémoire différents.

LE MULTITHREADING (ou multifil) est la capacité d'un programme ou d'un processus du système d'exploitation à gérer son utilisation par plusieurs utilisateurs à la fois, voire à gérer plusieurs demandes d'un même utilisateur, sans qu'il soit nécessaire de créer plusieurs exemplaires du programme exécuté sur l'ordinateur. Chaque demande d'utilisation d'un programme ou service du système (l'utilisateur pouvant dans ce cas être un autre programme) est suivie sous la forme d'un fil d'exécution (« thread ») doté d'une identité distincte. Lorsque l'exécution de la demande initiale correspondant à ce fil est interrompue par d'autres demandes, l'état des tâches qui composent le fil est suivi jusqu'à ce que ces tâches soient terminées.

III. Avantages

Comparativement aux processus à un flot de contrôle unique, un thread ou processus léger avec plusieurs flots de contrôle présente plusieurs avantages, notamment :

- **Réactivité**: Le processus léger continue à s'exécuter même si certaines de ses parties sont bloquées.
- **Économie d'espace mémoire et de temps**: Par exemple sous Solaris, la création d'un processus est 30 fois plus lente que celle d'un processus thread.

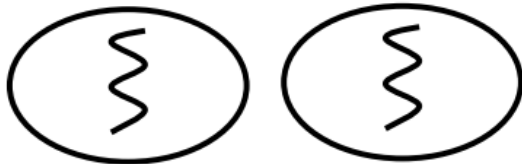
IV. Threads vs. processus



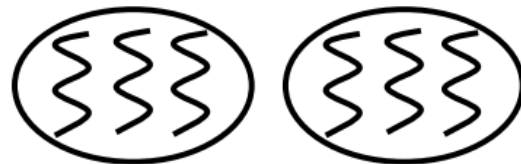
un processus
un thread



un processus
multiples threads



multiples processus
un thread par processus



multiples processus
multiples threads par processus

V. Exemple

Le programme `thread-pid.c` montre l'implantation de threads dans GNU/Linux, ainsi que la récupération du pid du thread.

`thread-pid.c`

```
#include <unistd.h>    // pour sleep
#include <pthread.h>    // pthread_create, pthread_join, pthread_exit
#include <stdio.h>

void *fonction(void *arg)
{
    printf("pid du thread fils = %d\n", (int) getpid());

    while(1); // forever
    return NULL;
}

int main()
{
    pthread_t thread;

    printf("pid de main = %d\n", (int) getpid());
    pthread_create(&thread, NULL, &fonction, NULL);
    while(1); // forever

    return 0;
}
```

Activate W
Go to Settings