

Exercice n°3 (Etude des mutex et variables de condition)

Vous devez lancer un certain nombre de threads dont les paramètres transmis sont

```
typedef struct
{
    char          szDonnee[20];
    int           NbSecondes;
} DONNEE;

DONNEE Elm[] = { "MERCENIER" , 5 ,
                 "VILVENS" , 9 ,
                 "WAGNER" , 3 ,
                 "DE FOOZ" , 7 ,
                 "" , 0 };
```

et attendre la fin de ceux-ci.

Pour cela, chaque thread secondaire

- Affichera une chaîne "Thread 10529.3 lancé" (PID = 10529 et NumSequence = 3).
- Affichera le nom passé en paramètre dans la structure.
- Attendra un nombre de secondes (nanosleep), fourni par la structure.
- Affichera une chaîne "Thread 10529.3 se termine".
- Se termine.

Le thread principal

- Lance les threads par une boucle qui contient
memcpy(&Param,Elm[i],sizeof(DONNEE)) ;
pthread_create(...,&Param) ;
Param étant une **variable globale partagée par tous les threads (!!!)**.
- Attend la fin des threads en implémentant un **compteur de threads (combinaison mutex – variable de condition)**. Le thread principal attend que ce compteur passe à 0 pour se terminer.

En plus....

Il est également demandé de lancer un threadAffTime qui affichera l'heure dans le coin supérieur droit de l'écran (la fonction time() donne le nombre de secondes écoulées depuis le 1/01/70, et la fonction ctime() la fournit sous forme d'une chaîne de 24 caractères : « Sat Jan 31 11:40:06 EST 2004 » par exemple)

De plus, on demande que l'affichage se fasse dans un HandlerAffTime (afin de jouer avec des signaux). Il faut donc armer et masquer correctement SIGALRM.

```
void *fctThreadAffTime(void *p)
{
    while (1)
    {
        alarm(1);
        pause();
    }
    // pthread_exit(NULL);
    // return NULL;
}
```