

Informatique embarqué

TP n°3 - MION Giovanni

Temps estimé: 10 heures

Temps effectif: 11 heures

I)Compilation du code

Compilation avec liaison des dynamique librairies:

make

Suppression des fichiers temporaires:

make clean

Suppression de l'ensemble des fichiers générés par make:

make mrproper

NB: le makefile ne fonctionne qu'avec le compilateur GCC.

II)Choix de l'implémentation

L'implémentation des sémaphores utilisé est celle des sémaphores système V.

Cela permet d'utilisé les mêmes sémaphores pour les threads et les processus sans se soucier de la gestion de la mémoire partagé (fait à partir d'un fichier).

III)Problèmes rencontrés

Voici une liste des problèmes rencontrés:

- Gestion des sémaphores

IV)Question du TP

3.1 Pour ce faire il nous faut mesurer un bloc de feuille mesurant plus d'un centimètre et calculer l'épaisseur moyenne d'une feuille en divisant la mesure par le nombre de feuille dans le paquet. Nous ferons de même lors du TP

3.2 time – appel system/bibliothèque – unité : seconde – précision : 1 seconde

gettimeofday – appel system/bibliothèque – unité:microseconde – précision 1 microseconde

clock – bibliothèque – unité : inconnue (CLOCKS_PER_SEC pour convertir) – précision : dans l'ordre de 10 millisecondes

clock_gettime – bibliothèque – unité : nanoseconde – précisions : 1 nanoseconde

3.3 Le point initial de mesure est avant la création du paquet de processus /threads . Le point final de mesure est après réception par le père/processus de la mort de ses fils/threads. Les opérations mesurées sont donc la création, la destruction et l'envoi du message de mort.

Résultats (en moyenne) pour la création/destruction:

processus : 26695.484580 nanosecondes CPU et 87189.555200 nanosecondes RT

threads : 9833.692260 nanosecondes CPU et 11003.878400 nanosecondes RT

Les résultats obtenus montrent qu'il est plus rapide de créer/détruire un thread qu'un processus. Cela est le comportement attendu sachant que les threads ont été créés dans le but d'alléger la notion de processus.

3.4 Les valeurs ne sont pas constantes car elles dépendent de l'environnement d'exécution qui lui évolue avec le temps. Pour publier des résultats il faut avoir un nombre d'échantillons important pour mener une analyse statistique.

3.5 Les événements suivants peuvent perturber la mesure :

- ordonnancement avec plusieurs autres processus/threads
- interruption système
- changement de processeur (si multi-cœur)

4 Résultats (en moyenne) pour un changement de contexte:

processus : 717.504660 nanosecondes CPU et 1472.757760 nanosecondes RT

threads : 1373.334660 nanosecondes CPU et 1379.921920 nanosecondes RT

On peut voir que le changement de contexte entre thread est plus rapide que celui de processus pour le wall time. Cela peut s'expliquer car il n'est pas nécessaire de chercher un processus éligible pour faire le changement (il est fait de manière interne au processus).

En revanche, si l'on regarde le temps CPU alors c'est le phénomène inverse qui se passe.

On pourrait expliquer cela par le fait qu'un changement de contexte de processus, chaque

processus est attaché a un contexte au niveau du système alors que pour les threads les contexte sont contenue dans les donnée du processus (il faut donc aller les chercher puis les charger)

Pour obtenir l'ordonnancement désiré j'utilise deux sémaphores qui sont initialisé a 0.

Chaque processus/thread vend un des sémaphores et réclame l'autre dans un ordre opposé.

Cela a pour effet de crée une alternance entre les deux entités informatique.

Les événements suivants peuvent perturber la mesure :

- ordonnancement avec plusieurs autre processus
- interruption système
- changement de processeur (si multi-coeur)

5 Voici les résultats donnés par l'outil lmbench

création de processus : 155000.0 nanosecondes RT

changement de contexte : 4130.0 nanosecondes RT

On peut voir que les résultats de lmbench sont plus grand que ceux que calculés précédemment. Cela peut venir du fait que lmbench donne les pires cas alors que les valeurs calculés précédemment sont des valeurs moyennes.

V)Commentaire d'un autre étudiant