Xenomai 3 – Prise en main

Damien Masson / Laurent George – prenom.nom@esiee.fr – bureaux 4206/5255

22 mai 2017

Vous allez utiliser le système d'exploitation Xenomai 3 installé en mode co-noyau.

1 Qu'est-ce que Xenomai

Vous allez commencer par lire la documentation sur la page `https://xenomai.org/start-here/` (lire les deux premières sections : "what is Xenomai about" ? et "How does Xenomai deliver real-time ?".

Vous êtes ensuite libre de chercher de la documentation pour répondre à ces questions :

— pourquoi Linux ne peut pas être utiliser pour programmer des applications avec des contraintes temps-réel fortes ?
— qu'apporte le patch "Preempt-RT" pour résoudre ce problème ?
— quelles sont ses limites ?
— quelle est l'autre approche proposé par Xenomai 3 (et par d'autre RTOS comme RTAI par exemple) ?

La documentation pour la suite se trouve à l'adresse `https://xenomai.org/documentation/xenomai-3/html/xeno3prm/group__alchemy.html`**.**

2 Changement de contexte et passage en mode secondaire

Téléchargez l'archive `http://igm.univ-mlv.fr/~masson/v2/Teachings/IN4R21/tp2/tp2.tgz`. Compilez en modifiant le fichier `Makefile` les fichiers `exemple-hello-01.c` et `exemple-hello-01-printf.c`

On souhaite analyser le nombre de changement de contexte observés lors de l'execution des deux programmes hello.

1. A quelle priorité sont exécutés ces programmes (cf fichier `/proc/xenomai/sched/threads`)
2. Comment évoluent les changements de contexte et le passage en mode secondaire pour les deux programmes ? (cf fichier `/proc/xenomai/sched/stat`, examinez les variables CSW et MSW).

3 Précision de l'activation périodique d'une tâche

1. Compilez le programme `exemple-periodique-01.c` avec le fichier `Makefile`. Ce programme lance une tâche de période 1 seconde qui affiche le temps courant. Lancez le programme et analyser si la durée entre deux periodes est bien constante (il ne doit pas y avoir de dépassement significatif)
2. Compilez le programme `exemple-periodique-02.c` avec le fichier `Makefile`. Dans ce programme, on execute une tache périodique de période 0,5 secondes mais avec une durée supérieure à 0,8 secondes. Interprétez les résultats.
3. Compilez le programme `exemple-periodique-03.c` avec le fichier `Makefile`. Dans ce programme, on execute une tache périodique de période 0,5 secondes mais avec une durée supérieure à 1 secondes. Interprétez les résultats. Le comportement vous semble-t-il conforme à la documentation ?
4. Compilez le programme `exemple-periodique-04.c` avec le fichier `Makefile` et le fichier `calc-stat.c` avec la commande :

   ```
   gcc -o calc-stat calc-stat.c -lm
   ```

   Le programme calc-stat permet de calculer le minimum, la moyenne et l'écart type d'une grandeur se trouvant dans un fichier.

   Lancez le programme :

   ```
   exemple-periodique-04 1000
   ```

   1000 correspond à une tâche de période 1ms. Vérifiez qu'il affiche bien la durée écoulée entre les activations successives de la tâche (en nano secondes).

   Sauvegardez les résultats obtenus dans un fichier (fic-res) pendant une minute :

   ```
   exemple-periodique-04 1000 > fic-res
   ```

Analysez les résultats pour un système non chargé :

`calc-stat <fic-res`

5. On souhaite maintenant evaluer l'influence de la charge sur les performances du programme `exemple-periodique-04` Pour cela, compilez le programme `hackbench` (procédure de compilation détaillée dans le fichier)

   Comparez les résultats obtenus par la commande `ping 127.0.0.1` (temps de réponse : time) avant et après l'execution de la commande :

   `/usr/xenomai/bin/dohell -b ./hackbench -s 127.0.0.1 -m /mnt/ 10`

   Refaites le protocole de la question 4 avec dohell qui tourne en parallèle. Concluez.