

TP3 MASTER2LSE – Partie 4 – NOYAU TEMPS REEL

Lors de ce TP, nous allons installer et tester différentes versions du noyau Linux. Nous allons voir tout d'abord, qu'il y a différentes versions de « préemption ». Ensuite, nous allons *patcher* le noyau pour tester l'extension temps réel PREEMPT-RT.

EXPLORATION DES DIFFERENTES OPTIONS DU NOYAU TEMPS REEL

Lister les possibilités du noyau classique Linux en terme de préemption. Allez sur le menu de configuration du noyau : `$ make linux-menuconfig`



Explorez le menu « Kernel Features » puis « Preemption Model », à quoi correspond chacune des options ?

Nous allons tester les 3 versions du noyau afin de savoir laquelle présente les meilleures « performances » temps réel. Nous n'allons pas tester l'ensemble des métriques bien sûr, nous allons nous attacher au test du temps de gigue (vu en TP2).

- 1) Testez le programme 3-b (et c) du TP2 sur les 3 noyaux. Utilisez des scripts pour accélérer les tests (les scripts sont demandés en annexe du rapport).
- 2) Faites la même chose pour le programme en 3-d du TP2

Que remarquez-vous ?



Concernant le rapport, il faut présenter les mesures obtenues sous forme de courbes (moyennes et écarts types pour différentes périodes). Il est demandé de fournir une analyse des résultats.

Nous allons à présent modifier le processus perturbateur. Au lieu de perturber simplement le CPU, nous allons générer des interruptions (vu du noyau de la carte). Pour ce faire, développez un programme ou un script sur la VM permettant de générer des *burst* de ping aléatoires vers la carte.

- 3) Ré exécuter le programme 3-d pour les 3 noyaux.



Que remarquez-vous ?

LINUX-RT

Nous allons dans cette partie utiliser un noyau Linux temps réel.

Nous utiliserons le patch Linux-rt, projet de Thomas Gleixner et Ingo Molnar (qui a aussi développé l'ordonnanceur CFS vu en cours). Linux-rt (aussi appelé Preempt-rt) se présente sous la forme d'un patch à appliquer au noyau Linux.

La page où l'on peut télécharger le patch est la suivante <https://www.kernel.org/pub/linux/kernel/projects/rt/>

Nous allons utiliser le patch pour la version 4.13.13 du noyau Linux, il faut donc télécharger le patch correspondant à cette version et modifier la version du noyau dans Buildroot. (Avant, nous avons utilisé la version 4.15.16)

- `menuconfig` → `Kernel` → `Kernel version` : `4.13.13`
- `menuconfig` → `Toolchain` → `Kernel Headers Linux 4.13.x kernel headers`
- `$ wget https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/4.13/patch-4.13.13-rt5.patch.gz`

Une fois le fichier téléchargé, on va le décompresser.

Copiez le patch dans le répertoire du source Linux : `./buildroot/output/build/linux-4.13.13`

A présent, nous allons tester le patch, entrez d'abord dans la racine du noyau : `./buildroot/output/build/linux-4.13.13/` et exécutez :

```
$ patch -p1 < ./patch-4.13.13-rt5.patch --dry-run
```

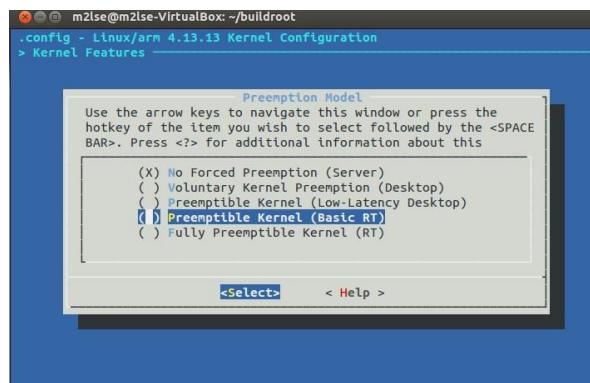


Que fait l'option `-dry-run` ? Testez aussi cette commande avec le patch dans le répertoire `buildroot/output/build/linux-4.15.16`

Si vous n'avez aucune erreur, refaites l'exécution **sans l'option `-dry-run`**:

```
$ patch -p1 < ./patch-4.13.13-rt5.patch
```

Maintenant, on peut revenir sur le menu de configuration du noyau Linux. Si tout s'est bien passé, vous devriez voir apparaître 2 options supplémentaires dans la configuration de la préemptibilité :



Faites un programme permettant de tester, tout comme pour la question 3, l'interruption des tâches par la réponse à des « ping ». On fera un programme qui s'exécute sur la carte et qui permet, dans une boucle active, de mesurer les occurrences d'interruptions du programme au-delà d'un certain seuil :

Boucle

```
// mesure de temps
// Différence avec mesure de la précédente itération
// Si différence > un certain seuil afficher le temps (delta)
```

Fin

- 4) Faites tourner ce programme (avec un processus temps réel) en parallèle avec celui permettant de générer des ping sur la carte. Faites cela pour les 5 noyaux, que remarquez-vous, analysez les résultats.