

Temps réel PREEMPT_RT

Christophe Blaess

christophe.blaess@logilin.fr

<https://www.blaess.fr/christophe/>
twitter: @chrisblaess



Ingénierie et formations sur Linux et les logiciels libres
<https://www.logilin.fr>

Patch PREEMPT_RT.....	3
Introduction.....	3
Principe d'installation.....	4
Threaded Interrupts.....	6
Travaux pratiques : interruptions threadées.....	7
Mesure des performances.....	8
Outil cyclicttest.....	8
Travaux pratiques : mesure de latences d'interruptions.....	9

Ce support de formation est distribué sous licence **Creative Commons 4.0**



(Attribution - Partage dans les mêmes conditions).

Vous êtes libres de copier et partager ce document, en mentionnant son origine. Si vous l'intégrez dans un contenu plus vaste, ce dernier devra être distribué avec les mêmes droits.

Ce cours a été rédigé en utilisant des logiciels libres sur système d'exploitation Linux :

- *LibreOffice Writer* pour le support et la mise en page
- *LibreOffice Draw* pour les dessins vectoriels
- *Gimp* pour les images bitmap

Temps réel Linux et Xenomai

v. 5.4

<https://www.blaess.fr/christophe/>

<https://www.logilin.fr>

Patch PREEMPT_RT

Introduction

Le projet *PREEMPT_RT* (parfois nommé Linux-RT) mené par Ingo Molnár et Thomas Gleixner est un patch améliorant les performances temps-réel de Linux.

Ne pas confondre ce projet avec RT-Linux – produit commercial de WindRiver.

Une intégration lente du patch PREEMPT_RT dans le noyau standard se poursuit depuis plusieurs années.

Le site principal du projet PREEMPT_RT est <http://rt.wiki.kernel.org/>

Les patches se trouvent sur www.kernel.org, par exemple :

<https://www.kernel.org/pub/linux/kernel/projects/rt/5.15/patch-5.15.79-rt54.patch.xz>

Attention, il existe une archive `patches-####.tar.bz2` qui contient tous les petits *patches* que Thomas Gleixner a intégré dans le fichier `patch-###.patch.bz2`. C'est seulement ce dernier fichier qui nous intéresse.

Principe d'installation

Télécharger la version du patch qui nous convient

```
$ wget https://www.kernel.org/pub/linux/kernel/projects/rt/5.15/patch-5.15.79-rt54.patch.xz
```

```
$ unxz patch-5.15.79-rt54.patch.xz
```

Télécharger les sources du noyau Linux correspondant au patch.

```
$ wget https://www.kernel.org/pub/linux/kernel/v5.x/linux-5.15.79.tar.xz
```

```
$ tar xf linux-5.15.79.tar.xz
```

Appliquer le patch en vérifiant qu'il n'y ait pas d'erreur.

```
$ cd linux-5.15.79
```

```
$ patch -p1 < ../patch-5.15.79-rt54.patch --dry-run
```

```
$ echo $?
```

```
0
```

```
$ patch -p1 < ../patch-5.15.79-rt54.patch
```

Configurer le kernel en partant de la configuration du système actuel

```
$ cp /boot/config-$(uname -r) ./config
```

```
$ make menuconfig
```

```
General setup --->
```

```
Preemption Model (Fully Preemptible Kernel (Real-Time)) --->
```

```
Cryptographic API --->
```

```
Certificates for signature checking --->
```

```
( ) Additional X.509 keys for default system keyring
```

```
( ) X.509 certificates to be preloaded into the system blacklist..
```

Compiler le noyau

```
$ make -j 8
```

Installer le noyau (plusieurs possibilités suivant les distributions)

```
$ sudo make modules_install
```

```
$ sudo make install
```

Au reboot, la présence de *PREEMPT_RT* est visible grâce au flag RT dans le résultat de la command ``uname -a``.

Threaded Interrupts

Avec le patch *PREEMPT_RT*, les interruptions sont gérées sous formes de threads du noyau.

La première partie du traitement est réalisée dans le handler d'interruption du kernel. La seconde dans un thread du kernel qui dispose par défaut d'une priorité temps réel FIFO 50.

On peut modifier la priorité d'un thread kernel avec `chrt`.

```
# ps
PID    USER      TIME    COMMAND
   1   root         0:01    init
   2   root         0:00   [kthreadd]
  [...]
  19   root         0:00   [irq/65-ARM Mail]
  [...]
  36   root         0:00   [irq/16-bcm2708_]
  [...]
  43   root         0:00   [irq/32-dwc_otg]
  44   root         0:00   [irq/32-dwc_otg_]
  45   root         0:00   [irq/32-dwc_otg_]
  47   root         0:00   [irq/24-DMA IRQ]
  48   root         0:00   [irq/25-DMA IRQ]
  49   root         0:00   [irq/84-mmc0]
  [...]
  53   root         0:00   [irq/83-uart-pl0]
  [...]
#
```

Travaux pratiques : interruptions threadées

Reprenez le programme `exemple-02` du chapitre précédent et exécutez-le.

Un « `ping` » depuis une autre station peut-il interrompre l'exécution de la boucle ?

Voyez-vous avec « `ps aux` » les threads correspondants au driver Ethernet ?

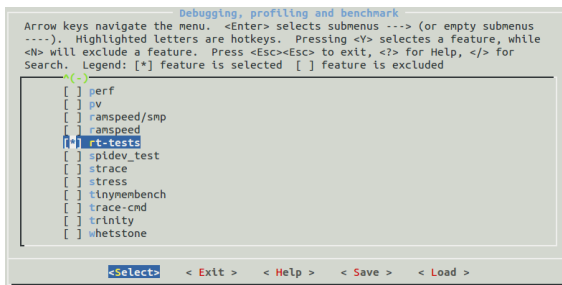
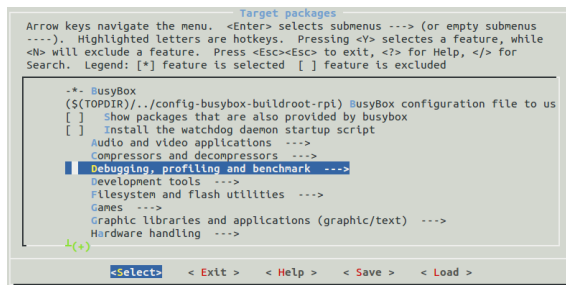
Si oui, essayez de modifier leurs priorités pour que le `ping` fonctionne à nouveau durant l'exécution de la boucle.

Mesure des performances

Outil cyclictest

Cyclictest, écrit par Thomas Gleixner et maintenu dans le package *RT-Tests* de Clark Williams est un outil fameux pour mesurer les fluctuations des timers temps-réel.

Il est disponible avec les packages de mesures et évaluations de Buildroot.



Sur une distribution type Debian il est disponible avec la commande

```
$ sudo apt install -y rt-tests
```


Travaux pratiques : mesure de latences d'interruptions

Installez et exécutez **cyclictest** sur un noyau Vanilla et sur un noyau PREEMPT_RT.

Remarquez les différences de tenue des *timers* entre les deux noyaux.

Essayez de monter la charge système de manière intense.

Arrangez-vous également pour charger le système de manière importante en interruptions.

```
$ sudo cyclictest -n -i 200 -p 99 -D 2m
```

```
# /dev/cpu_dma_latency set to 0us
```

```
policy: fifo: loadavg: 0.22 0.39 0.18 1/211 785
```

```
T: 0 (774) P:99 I:200 C:599960 Min: 12 Act: 24 Avg: 20 Max: 98
```