

Sistemas Operacionais Embarcados

RT-PREEMPT

Multitasking cooperativo e preemptivo

- O kernel Linux não oferece performance hard real-time, pois ele não garante a temporização tão precisa na execução de processos e threads. Isso acontece porque o kernel Linux executa multitasking cooperativo
- O patch RT PREEMPT (Realtime Preemption) do kernel Linux é uma modificação do mesmo que garante a execução de multitasking preemptivo.

Multitasking cooperativo e preemptivo

- No multitasking COOPERATIVO, os processos devem voluntariamente ceder sua vez para outros processos.
- No multitasking PREEMPTIVO, os processos podem ser interrompidos por outros processos de maior prioridade.
- Fica claro que, no multitasking cooperativo, o kernel pode atrapalhar a execução de processos no userspace, afetando a temporização destes.

Patch RT-PREEMPT

- A instalação do patch RT-PREEMPT requer uma série de cuidados:
 - Faça um backup do cartão SD do Raspberry Pi no seu computador. Se houver algum imprevisto na instalação ou na execução do RT PREEMPT, você poderá recuperar o estado do cartão pelo backup.
 - Algumas formas de fazer o backup de um cartão SD:
 - <https://raspberrypi.stackexchange.com/a/313>
 - <http://lifehacker.com/how-to-clone-your-raspberry-pi-sd-card-for-super-easy-r-1261113524>
 - <https://www.raspberrypi.org/documentation/linux/filesystem/backup.md>
 - Outra opção interessante é ter dois cartões SD, um com o patch e outro sem.

Patch RT-PREEMPT

- Para instalar o patch RT PREEMPT, execute os comandos a seguir em um computador com Linux instalado e com pelo menos 4GB de memória disponível. O cifrão indica o terminal do Linux e o caminho atual no terminal.

```
~$ mkdir ~/rpi-kernel
~$ cd ~/rpi-kernel
~/rpi-kernel$ mkdir rt-kernel
~/rpi-kernel$ git clone
https://github.com/raspberrypi/linux.git -b rpi-4.14.y-rt
~/rpi-kernel$ git clone
https://github.com/raspberrypi/tools.git
```

(Esta parte demora um pouco, pois os repositórios baixados tem de 2,5 a 3GB de dados.)

Patch RT-PREEMPT

```
~/rpi-kernel$ cd linux
~/rpi-kernel/linux$ git checkout rpi-4.14.y-rt
~/rpi-kernel$ export ARCH=arm
~/rpi-kernel$ export
CROSS_COMPILE=~/rpi-kernel/tools/arm-bcm2708/gcc-linaro-arm-l
inux-gnueabihf-raspbian/bin/arm-linux-gnueabihf-
~/rpi-kernel$ export INSTALL_MOD_PATH=~/rpi-kernel/rt-kernel
~/rpi-kernel$ export INSTALL_DTBS_PATH=~/rpi-kernel/rt-kernel
```

Se o seu computador for de 64 bits, execute

```
$ export
CROSS_COMPILE=~/rpi-kernel/tools/arm-bcm2708/gcc-linaro-arm-l
inux-gnueabihf-raspbian-x64/bin/arm-linux-gnueabihf-
```

Patch RT-PREEMPT

Se a versão do seu Raspberry Pi for 1/1.2 B(+), A(+) ou Zero (W), execute

```
~/rpi-kernel$ export KERNEL=kernel  
~/rpi-kernel$ cd ~/rpi-kernel/linux/  
~/rpi-kernel/linux/$ make bcmrpi_defconfig
```

Se a versão do seu Raspberry Pi for 2, 3 B(+), execute

```
~/rpi-kernel$ export KERNEL=kernel7  
~/rpi-kernel$ cd ~/rpi-kernel/linux/  
~/rpi-kernel/linux/$ make bcm2709_defconfig
```

Patch RT-PREEMPT

Confira quantos núcleos a CPU do seu Raspberry Pi tem, substitua este valor na opção -jX nos comandos abaixo, e os execute. (Por exemplo, para 4 núcleos, substitua -jX por -j4.)

```
~/rpi-kernel/linux$ make -jX zImage  
~/rpi-kernel/linux$ make -jX modules  
~/rpi-kernel/linux$ make -jX dtbs  
~/rpi-kernel/linux$ make -jX modules_install  
~/rpi-kernel/linux$ make -jX dtbs_install
```

(Essa parte demora.)

A última linha do comando `make -jX modules_install` indica a versão do kernel que você acabou de compilar (por exemplo, DEPMOD 4.14.52-rt34-v7+). Salve esta informação em um arquivo-texto.

Patch RT-PREEMPT

Agora, execute

```
~/rpi-kernel/linux$ mkdir $INSTALL_MOD_PATH/boot
~/rpi-kernel/linux$ ./scripts/mkknimg ./arch/arm/boot/zImage
$INSTALL_MOD_PATH/boot/$KERNEL.img
~/rpi-kernel/linux$ cd $INSTALL_MOD_PATH
~/rpi-kernel/rt-kernel$ tar czf ../rt-kernel.tgz *
~/rpi-kernel/rt-kernel$ cd ..
```

Estes últimos comandos criam o arquivo `~/rpi-kernel/rt-kernel/rt-kernel.tgz`, que é uma imagem comprimida do sistema operacional com o kernel modificado. Transfira este arquivo para o seu Raspberry Pi usando o comando scp:

```
~/rpi-kernel$ scp rt-kernel.tgz pi@<ipaddress>:/tmp
```

onde <ipaddress> é o endereço IP do seu Raspberry Pi.

Patch RT-PREEMPT

Agora, faça acesso remoto ao seu Raspberry Pi:

```
~/rpi-kernel$ ssh pi@<ipaddress>
```

Recupere a versão do kernel que você obteve após o comando `make -jX modules_install`, substituindo <VERSÃO_DO_KERNEL> nos comandos abaixo:

```
~$ cd /tmp
/tmp$ tar xzf rt-kernel.tgz
/tmp$ cd boot
/tmp/boot$ sudo cp -rd * /boot/
/tmp/boot$ cd ../lib
/tmp/lib$ sudo cp -dr * /lib/
/tmp/lib$ cd ../overlays
/tmp/overlays$ sudo cp -d * /boot/overlays
/tmp/overlays$ cd ..
/tmp$ sudo cp -d bcm* /boot/
sudo echo "kernel=vmlinux-<VERSÃO_DO_KERNEL>" >>
/boot/config.txt
```

Patch RT-PREEMPT

Execute `sudo reboot`, entre novamente no Raspberry Pi via ssh e execute `uname -r` para conferir a nova versão do kernel.

Benchmarking

Para comparar o desempenho do Raspberry Pi com e sem o patch RT PREEMPT, instale a suíte de testes `rt-tests` no Raspberry Pi com e sem o patch, da seguinte forma:

```
~$ sudo apt-get install build-essential libnuma-dev
~$ git clone
git://git.kernel.org/pub/scm/utils/rt-tests/rt-tests.git
~$ cd rt-tests
~/rt-tests$ git checkout stable/v1.0
~/rt-tests$ make all
~/rt-tests$ sudo make install
```

Benchmarking

Esta suíte instala uma série de programas que podem ser usados para testar o desempenho do sistema. Por exemplo, o comando `cyclictest` mede diversas vezes a diferença entre o momento pretendido de reativação de uma thread e o momento de real reativação da mesma, oferecendo estatísticas das latências do sistema. Execute:

```
$ sudo cyclictest -l100000 -m -S -p90 -i200 -h400 -q | tail -8
```

em um Raspberry Pi com o patch RT PREEMPT e em um Raspberry Pi sem o patch para conferir o efeito do patch sobre o desempenho do sistema.

Referências

Multitasking cooperativo e preemptivo:

- https://en.wikipedia.org/wiki/Cooperative_multitasking
- [https://en.wikipedia.org/wiki/Preemption_\(computing\)](https://en.wikipedia.org/wiki/Preemption_(computing))

Algumas formas de fazer o backup de um cartão SD:

- <https://raspberrypi.stackexchange.com/a/313>
- <http://lifehacker.com/how-to-clone-your-raspberry-pi-sd-card-for-super-easy-r-1261113524>
- <https://www.raspberrypi.org/documentation/linux/filesystem/backup.md>

Referências

Os passos de instalação foram baseados em:

- <https://lemariva.com/blog/2018/07/raspberry-pi-preempt-rt-patching-tutorial-for-kernel-4-14-y>

Benchmarking:

- <https://wiki.linuxfoundation.org/realtime/documentation/howto/tools/rt-tests>
- <https://wiki.linuxfoundation.org/realtime/documentation/howto/tools/cyclicttest/start>

Os códigos nesta pasta são baseados nos seguintes documentos:

- <https://www.youtube.com/watch?v=uIXkvz1-weQ>
- http://www.isy.liu.se/edu/kurs/TSEA81/lecture_linux_realtime.html