

Programmation Temps Réels

Travaux à réaliser

La liste des travaux à réaliser apparaîtra progressivement ici.

Vous devez réaliser vos travaux et compléter votre carnet de laboratoire.

Indiquez dans votre carnet :

- votre nom au haut du carnet
- la date du jour
- les commandes introduites et leur résultat (sous forme de copie/collé plutôt que de copie d'écran)

Vous ajouterez vos fichiers sources sous forme de fichier .zip au devoir en cours (pensez à indiquer le nom du fichier zip dans votre carnet)

Vous rendrez le devoir en fin d'UE. **La réalisation continue au fil des laboratoires de votre carnet de laboratoire fera office d'examen de première session.**

J'ajouterai régulièrement des commentaires directement dans votre carnet de laboratoire.

A. Ajout d'une courtoisie aux threads (Voir le listing exemple-nice-threads.c, chapitre 3 du livre "Solutions temps réel sous Linux")

1. Modifier votre solution Reader/Writer pour supprimer la priorité des writers (listing 4.20 et 4.21 du livre "little book of semaphore"). Tester la solution avec beaucoup de readers. Que se passe-t-il ?
2. Ajouter une courtoisie *inférieure* aux thread writers. Comparer les résultats

Exemple : de commandes à placer dans le rapport :

Sans courtoisie 20 threads dont 15 écrivains :

```
bin/readerswriters 20 15 20 > rw-20-15-20.txt
cat rw-20-15-20.txt | grep r | wc -l
724
```

Avec courtoisies 1 et 10 :

```
bin/readerswriters 20 15 20 > rw-nice-20-15-20.txt
cat rw-nice-20-15-20.txt | grep r | wc -l
865
```

Conclusion : la courtoisie permet d'augmenter la place des lecteurs. Est-ce significatif ?

Réalisons 10 mesures pour chaque programme et calculons les moyennes et écart-type des nombres de threads lecteur.

B. Ajout d'une priorité aux threads (Voir le listing exemple-threads-temps-reel.c, chapitre 5 du livre "Solutions temps réel sous Linux")

1. Modifier votre solution Reader/Writer pour supprimer la priorité des writers (listing 4.20 et 4.21 du livre "little book of semaphore"). Tester la solution avec beaucoup de readers. Que se passe-t-il ?

2. Ajouter une priorité supérieure aux thread writers. Comparer les résultats

Pour les tests des exercices A et B, choisissez des timings (nombre de threads, temps de repos) qui mettent en évidence la différence entre les points 1 et 2. Ne montrer qu'une portion des résultats (c'est long) et surtout vos conclusions. Enregistrer les résultats dans des fichiers (via les redirections du shell) et ajouter les fichiers résultats avec les codes sous forme de .zip

~~G. Ajout d'une priorité aux threads (Voir le listing exemple threads-temps-reel.c, chapitre 5 du livre "Solutions temps réel sous Linux) en ajoutant un mécanisme d'héritage de priorité aux mutexes roomEmpty en paramètre du lightswitch (voir listing exemple pip.c du chapitre 7 du livre "Solutions temps réel sous Linux)~~

- ~~1. Modifier votre solution Reader/Writer pour supprimer la priorité des writers (listing 4.20 et 4.21 du livre "little book of semaphore"). Tester la solution avec beaucoup de readers. Que se passe-t-il ?~~
- ~~2. Ajouter une priorité supérieure aux thread writers. Comparer les résultats~~
- ~~3. Comparer avec le code original qui donne une priorité absolue programmatique aux écrivains~~

Ne semble pas fonctionner

D. Comparaison Linux-Vanilla / patch PREEMPT_RT / XENOMAI 3 sur Raspberry PI 1, 2, 3 & Raspberry PI 4

Voir tout d'abord la documentation Raspberry-Pi-et-temps-reel.pdf

1. Documentation

Pi1 : <http://www.simplerobot.net/2018/06/build-realtime-xenomai-3-kernel-for.html>

Pi2 : http://www.simplerobot.net/2018/06/build-realtime-xenomai-3-kernel-for_3.html

Pi3 : http://www.simplerobot.net/2018/06/build-realtime-xenomai-3-kernel-for_3.html

Pi4 : <http://www.simplerobot.net/2019/12/xenomai-3-for-raspberry-pi-4.html> et <https://lemariva.com/blog/2019/09/raspberry-pi-4b-preempt-rt-kernel-419y-performance-test>

ou

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

2. Installation du cross-compileur sur le Host Ubuntu 18.04 !

```
sudo apt install gcc-arm-linux-gnueabihf
```

3. Suite pour le Pi 1 sur le Host Ubuntu 18.04 !

Kernel Vanilla :

On va récupérer le noyau 4.1 avec la commande suivante :

```
git clone -b rpi-4.1.y --depth 1
```

```
git://github.com/raspberrypi/linux.git
```

```
rpi-kernel-vanilla-4.1
```

Suite comme sur le document mais avec le kernel 4.1

```
make -j2 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-  
bcmrpi_defconfig  
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j2 bzImage  
modules dtbs  
make ARCH=arm INSTALL_MOD_PATH=dist -j2 modules_install  
  
cd arch/arm/boot  
tar -cjvf linux-dts-4.1.21.tar.bz2 dts  
  
cd dist/lib/modules/  
tar -cf 4.1.21.tar 4.1.21+/  

```

On copie zImage, 4.1.21.tar et linux-dts-4.1.21.tar.bz2 sur le Pi1

Sur le Pi1:

On copie zImage en /boot/kernel.img (ne pas oublier de garder une copie de l'original)

On *detar* 4.1.21.tar sur /lib/modules

On copie les fichiers/dossiers de dts sur /boot

On redémarre

```
pi@raspberrypi:~ $ uname -a  
Linux raspberrypi 4.1.21+ #1 Thu May 20 15:09:42 UTC  
2021 armv6l GNU/Linux
```

4. patch PREEMPT_RT sur le Host Ubuntu 18.04 !

```
git clone -b rpi-4.1.y --depth 1  
git://github.com/raspberrypi/linux.git  
rpi-kernel-preempt-4.1  
cd rpi-kernel-preempt-4.1/  
wget  
https://www.kernel.org/pub/linux/kernel/projects/rt/4.1/olde  
r/patch-4.1.20-rt23.patch.xz  
xz -d patch-4.1.20-rt23.patch.xz  
patch -p1 < patch-4.1.20-rt23.patch  
(ignorer les erreurs : Reversed (or previously applied) patch detected! Assume -R?  
[n] y)  
make -j2 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-  
bcmrpi_defconfig  
make -j2 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-  
menuconfig
```

Dans le menu « Kernel features », pour l'option « Preemption model » sélectionner la configuration « Fully Preemptible Kernel (RT) ».

```
make ARCH=arm CROSS_COMPILE=arm-linux- -j2 bzImage modules  
dtbs  
make ARCH=arm INSTALL_MOD_PATH=dist -j2 modules_install
```

```
cd arch/arm/boot
tar -cjvf linux-dts-4.1.21-preempt.tar.bz2 dts
```

```
cd dist/lib/modules/
tar -cf 4.1.21-rt23.tar 4.1.21-rt23+/*
```

On copie zImage, 4.1.21-rt23.tar et
linux-dts-4.1.21-preempt.tar.bz2 sur le Pi1

Sur le Pi1:

On copie zImage en /boot/kernel.img (ne pas oublier de garder une copie
de l'original)

On **detar** 4.1.21-rt23.tar sur /lib/modules

On copie les fichiers/dossiers de dts sur /boot

On redémarre

```
pi@raspberrypi:~ $ uname -a
Linux raspberrypi 4.1.21-rt23+ #1 PREEMPT RT Thu May 27
14:49:54 UTC 2021 armv6l GNU/Linux
```

5. patch Xenomai sur le Host Ubuntu 18.04 !

Preparation on host PC

```
sudo apt-get install gcc-arm-linux-gnueabi
```

(cf <http://www.simplerobot.net/2018/06/build-realtime-xenomai-3-kernel-for.html>)

voir aussi

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

Extraction de xenomai 3.

```
wget
https://xenomai.org/downloads/xenomai/stable/xenomai-3.0.5.tar.bz2
tar -xjvf xenomai-3.0.5.tar.bz2
ln -s xenomai-3.0.5 xenomai
```

MODIFY 'xenomai/scripts/prepare-kernel.sh' file Replace 'ln -sf'
by 'cp' so that it will copy all necessary xenomai files to linux source

```
git clone -b rpi-4.1.y --depth 1
git://github.com/raspberrypi/linux.git
rpi-kernel-xenomai-4.1
ln -s rpi-kernel-xenomai-4.1 linux
```

```
mkdir xeno3-patches
```

```
cd xeno3-patches
```

Download all files in this directory and save them to xeno3-patches directory

<https://github.com/thanhtam-h/rpi01-4.1.21-xeno3/tree/master/scripts>

```
wget
```

```
https://raw.githubusercontent.com/thanhtam-h/rpi01-4.1.21-xeno3/master/scripts/1-rpi-4.1.y-add-pi0W.patch
```

```
wget
```

```
https://raw.githubusercontent.com/thanhtam-h/rpi01-4.1.21-xeno3/master/scripts/2-ipipe-core-4.1.18-arm-10-for-4.1.21.patch
```

```
wget
```

```
https://raw.githubusercontent.com/thanhtam-h/rpi01-4.1.21-xeno3/master/scripts/3-ipipe-core-4.1.21-raspberry-post.patch
```

```
wget
```

```
https://raw.githubusercontent.com/thanhtam-h/rpi01-4.1.21-xeno3/master/scripts/pinctrl-bcm2835.c
```

Patching

```
cd linux
```

```
patch -p1 <../xeno3-patches/1-rpi-4.1.y-add-pi0W.patch
../xenomai/scripts/prepare-kernel.sh --linux=./ --arch=arm
--ipipe=../xeno3-patches/2-ipipe-core-4.1.18-arm-10-for-4.1.21.patch
patch -p1
<../xeno3-patches/3-ipipe-core-4.1.21-raspberry-post.patch
cp ../xeno3-patches/pinctrl-bcm2835.c drivers/pinctrl/bcm/
```

Building kernel

```
make -j2 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
bcmrpi_defconfig
make -j2 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-
menuconfig
```

Select options as following:

```
General setup ---> | | (-xeno3) Local version - append to kernel release | | Stack
Protector buffer overflow detection (None) --->
```

```
Kernel Features --->
```

```
| | Preemption Model (Preemptible Kernel
(Low-Latency Desktop)) --->
| | Timer frequency (1000 Hz) --->
| | [ ] Allow for memory compaction
| | [ ] Contiguous Memory Allocator
```

```
CPU Power Management --->
```

```
| | CPU Frequency scaling --->
| | [ ] CPU Frequency scaling
```

```
Kernel hacking --->
```

```
| | [ ] KGDB: kernel debugger ---
```

Attention, il y a une erreur de compilation

```
../drivers/char/broadcom/vc_sm/vmcs_sm.c:200:26: error: 'sm_cache_map_vector'
defined but not used [-Werror=unused-const-variable=]
static const char *const sm_cache_map_vector[] = {
```

=> il faut modifier le fichier

```
../drivers/char/broadcom/vc_sm/vmcs_sm.c:
```

retirer ou commenter la déclaration de `sm_cache_map_vector`

```
make O=build ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j4
bzImage modules dtbs
make O=build ARCH=arm INSTALL_MOD_PATH=dist -j4
modules_install
make O=build ARCH=arm KBUILD_DEBARCH=armhf
CROSS_COMPILE=arm-linux-gnueabihf- -j4 deb-pkg
```

=>

```
dpkg-deb: building package 'linux-firmware-image-4.1.21-xeno3+' in
'../linux-firmware-image-4.1.21-xeno3+_4.1.21-xeno3+-1_armhf.deb'.
dpkg-deb: building package 'linux-headers-4.1.21-xeno3+' in
'../linux-headers-4.1.21-xeno3+_4.1.21-xeno3+-1_armhf.deb'.
dpkg-deb: building package 'linux-libc-dev' in
'../linux-libc-dev_4.1.21-xeno3+-1_armhf.deb'.
dpkg-deb: building package 'linux-image-4.1.21-xeno3+' in
'../linux-image-4.1.21-xeno3+_4.1.21-xeno3+-1_armhf.deb'.
make[1]: Leaving directory '/home/pol/tempsReel/pil/linux-rpi-4.1.y-xeno3/build'
```

Compress dts files

```
cd build/arch/arm/boot
tar -cjvf linux-dts-4.1.21-xeno3+.tar.bz2 dts
cd ..
cd ..
cd ..
cd ..
cp build/arch/arm/boot/linux-dts-4.1.21-xeno3+.tar.bz2 ./
```

COPY linux-headers, linux-image and linux-dts to rpi

In rpi, deploy:

```
sudo dpkg -i linux-image*
sudo dpkg -i linux-headers*
sudo tar -xjvf linux-dts-4.1.21-xeno3+.tar.bz2
cd dts
sudo cp -rf * /boot/
sudo mv /boot/vmlinuz-4.1.21-xeno3+ /boot/kernel.img
sudo reboot
```

```
pi@raspberrypi:~ $ uname -a
Linux raspberrypi 4.1.21-xeno3+ #1 PREEMPT Sat May 29
21:36:44 UTC 2021 armv6l GNU/Linux
```

Build xenomai user-space libraries and tools (sur le Host PC Ubuntu 18.04)

To remove errors install any required packages : `sudo apt install pkg-config build-essential libfuse-dev libcurl4-openssl-dev libxml2-dev mime-support automake libtool`

```
cd xenomai
./scripts/bootstrap
./configure CFLAGS="-no-pie" --host=arm-linux-gnueabi
--disable-smp --with-core=cobalt
```

To remove this error I add CFLAGS="-no-pie" to .configure script :

```
make[2]: Entering directory '/home/pol/tempsReel/pi1/xenomai-3.0.5/testsuite/latency'
CC      latency-latency.o
CCLD    latency
/usr/lib/gcc-cross/arm-linux-gnueabi/7/../../../../arm-linux-gnueabi/bin/ld: -r
and -pie may not be used together
```

To remove this error:

```
In file included from calibration_ni_m.h:27:0,
                  from calibration_ni_m.c:26:
analogy_calibrate.h: In function '__debug':
analogy_calibrate.h:95:2: error: implicit declaration of function 'clock_gettime'
[-Werror=implicit-function-declaration]
```

Add `#include "time.h"` to `utils/analogy/analogy_calibrate.h`

```
make
sudo make install
```

After installation, the built xenomai for raspberry will be located at `/usr/xenomai` directory on host PC, compress it and transfer to rpi:

```
tar -cjvf rpi01-xeno3-deploy.tar.bz2 /usr/xenomai
```

Transfer this file (rpi01-xeno3-deploy.tar.bz2) to rpi and extract it:

```
sudo tar -xjvf rpi01-xeno3-deploy.tar.bz2 -C /
```

On Pi1 Make a configuration file and link to xenomai directory

```
sudo nano /etc/ld.so.conf.d/xenomai.conf
```

Add to this file:

```
#xenomai lib path
/usr/local/lib
/usr/xenomai/lib
```

Save it and run `ldconfig` command:

```
sudo ldconfig
```

```
sudo /usr/xenomai/bin/latency
```

```
-> Illegal Instruction ????????????????????
```

Compilation directe sur le Pi1;

Installer xenomai-3.0.5

(cf .

<https://www.blaess.fr/christophe/2017/03/20/xenomai-sur-raspberry-pi-3-bilan-mitige/>)

```
cd xenomai
```

```
./scripts/bootstrap
```

```
./configure --disable-smp --with-core=cobalt
```

Attention, la compilation avec le compilateur gcc version 8.3.0 (Raspbian 8.3.0-6+rpi1) détecte de nombreuses erreurs qui n'étaient pas détectées par les compilateurs plus anciens.

Je les ai corrigées une à une mais il vaut mieux faire les corrections avant de compiler

```
make
```

-> **error:**

```
smokey_net_server.c: In function 'rtnet_rtcfg_add_client':
smokey_net_server.c:91:2: error: 'memcpy' forming offset [7, 32] is out of the bounds
[0, 6] of object 'mac' with type 'struct ether_addr' [-Werror=array-bounds]
  memcpy(cmd.args.add.mac_addr, mac.ether_addr_octet,
  ^~~~~~
  sizeof(cmd.args.add.mac_addr));
  ~~~~~
```

```
nano testsuite/smokey/net_common/smokey_net_server.c
```

commenter la ligne memcpy (! ok cette fonction sera buggée)

To remove this error:

```
In file included from calibration_ni_m.h:27:0,
                  from calibration_ni_m.c:26:
analogy_calibrate.h: In function '__debug':
analogy_calibrate.h:95:2: error: implicit declaration of function 'clock_gettime'
[-Werror=implicit-function-declaration]
```

Add #include "time.h" to utils/analogy/analogy_calibrate.h

-> **error**

```
rtcanrecv.c: In function 'main':
rtcanrecv.c:251:2: error: 'strncpy' specified bound 16 equals destination size
[-Werror=stringop-truncation]
  strncpy(ifr.ifr_name, argv[optind], IFNAMSIZ);
```

In utils/can/rtcanrecv.c commenter la ligne strncpy(ifr.ifr_name, argv[optind], IFNAMSIZ); (! ok cette fonction sera buggée)

Il serait mieux de changer IFNAMSIZ en IFNAMSIZ-1

-> **error:**

```
rtcansend.c:234:5: error: 'strncpy' specified bound 16 equals destination size
[-Werror=stringop-truncation]
  strncpy(ifr.ifr_name, argv[optind], IFNAMSIZ);
```

In utils/can/rtcansend.c commenter la ligne strncpy(ifr.ifr_name, argv[optind], IFNAMSIZ); (! ok cette fonction sera buggée)

Il serait mieux de changer IFNAMSIZ en IFNAMSIZ-1

-> **error:**

```
rtcanconfig.c:162:5: error: 'strncpy' specified bound 16 equals destination size
[-Werror=stringop-truncation]
  strncpy(iframe, argv[optind], IFNAMSIZ);
```


In `utils/can/rtcancanconfig.c` commenter la ligne `strncpy(ifname, argv[optind], IFNAMSIZ);` (! ok cette fonction sera buggée)
Il serait mieux de changer `IFNAMSIZ` en `IFNAMSIZ-1`

-> error:

```
cyclictest.c:1454:4: error: 'strncpy' specified bound 256 equals destination size
[-Werror=stringop-truncation]
    strncpy(tracer, optarg, sizeof(tracer));
    ^~~~~~
cyclictest.c:1349:4: error: 'strncpy' output truncated before terminating nul copying
as many bytes from a string as its length [-Werror=stringop-truncation]
    strncpy(fifopath, optarg, strlen(optarg));
    ^~~~~~
cyclictest.c:1349:4: error: 'strncpy' specified bound depends on the length of the
source argument [-Werror=stringop-overflow=]
    strncpy(fifopath, optarg, strlen(optarg)-1);
```

In `demo/posix/cyclictest/cyclictest.c` modifier les lignes

```
strncpy(tracer, optarg, sizeof(tracer)); en
strncpy(tracer, optarg, sizeof(tracer)-1); et
strncpy(fifopath, optarg, strlen(optarg)); en
strncpy(fifopath, optarg, sizeof(fifopath)-1);
```

-> error:

```
can-rtt.c:251:5: error: 'strncpy' specified bound 16 equals destination size
[-Werror=stringop-truncation]
    strncpy(ifr.ifr_name, rxdev, IFNAMSIZ);
    ^~~~~~
can-rtt.c:285:2: error: 'strncpy' specified bound 16 equals destination size
[-Werror=stringop-truncation]
    strncpy(ifr.ifr_name, txdev, IFNAMSIZ);
    ^~~~~~
```

In `demo/posix/cobalt/can-rtt.c` modifier les lignes

```
strncpy(ifr.ifr_name, rxdev, IFNAMSIZ); en
strncpy(ifr.ifr_name, rxdev, IFNAMSIZ-1); et
strncpy(ifr.ifr_name, txdev, IFNAMSIZ); en
strncpy(ifr.ifr_name, txdev, IFNAMSIZ-1);
```

-> error:

```
eth_p_all.c:75:3: error: 'strncpy' specified bound 16 equals destination size
[-Werror=stringop-truncation]
    strncpy(ifr.ifr_name, argv[1], IFNAMSIZ);
    ^~~~~~
```

In `demo/posix/cobalt/eth_p_all.c` modifier la ligne

```
strncpy(ifr.ifr_name, argv[1], IFNAMSIZ); en
strncpy(ifr.ifr_name, argv[1], IFNAMSIZ-1);
```

make

sudo make install

sudo /usr/xenomai/bin/latency

== Sampling period: 1000 us

== Test mode: periodic user-mode task

== All results in microseconds

warming up...

RTT| 00:00:01 (periodic user-mode task, 1000 us period, priority 99)

RTH	----lat min	----lat avg	----lat max	-overrun	---msw	---lat best	--lat worst
RTD	0.992	5.056	23.680	0	0	0.992	23.680
RTD	1.028	5.555	42.468	0	0	0.992	42.468
RTD	1.208	5.304	45.540	0	0	0.992	45.540
RTD	1.220	5.655	45.112	0	0	0.992	45.540
RTD	1.016	6.371	46.684	0	0	0.992	46.684
RTD	1.120	5.658	44.948	0	0	0.992	46.684
RTD	1.312	5.543	43.900	0	0	0.992	46.684
RTD	1.032	5.237	44.752	0	0	0.992	46.684
RTD	1.352	8.501	47.356	0	0	0.992	47.356
RTD	3.880	34.995	51.364	0	0	0.992	51.364
RTD	1.212	5.645	41.380	0	0	0.992	51.364
RTD	1.752	36.515	48.528	0	0	0.992	51.364
RTD	18.312	37.666	49.412	0	0	0.992	51.364
RTD	29.160	35.278	46.584	0	0	0.992	51.364
RTD	28.688	35.140	50.032	0	0	0.992	51.364
RTD	29.640	34.960	43.428	0	0	0.992	51.364
RTD	17.980	38.253	49.912	0	0	0.992	51.364
RTD	19.568	35.694	46.544	0	0	0.992	51.364
RTD	19.228	35.204	47.840	0	0	0.992	51.364
RTD	1.184	27.592	52.364	0	0	0.992	52.364
RTD	1.312	5.433	37.724	0	0	0.992	52.364

E. Exemples d'application XENOMAI 3 sur Raspberry PI 1, 2, 3 & Raspberry PI 4

Documentation Xenomai :

<https://xenomai.org/documentation/xenomai-3.0.7/html/xeno3prm/>

1. periodic_task

cf. <https://www.ashwinnarayan.com/post/xenomai-realtime-programming-part-2/>

Les fichiers sont dans l'archive `periodic_task.tar.bz2`

J'ai compilé directement sur le Pi1 mais il devrait être possible de faire de la cross-compilation si les user-space lib ont pu être cross-compilée (ce n'est pas le cas ici)

un make crée le fichier `cyclic_test`

```
./cyclic_test
Starting cyclic task...
Starting task cyclic_task with period of 10 ms ....
Loop count: 0, Loop time: 0.00813 ms
Loop count: 1, Loop time: 9.86126 ms
Loop count: 2, Loop time: 19.86726 ms
Loop count: 3, Loop time: 29.86578 ms
Loop count: 4, Loop time: 39.86638 ms
Loop count: 5, Loop time: 49.86560 ms
Loop count: 6, Loop time: 59.86516 ms
Loop count: 7, Loop time: 69.86752 ms
Loop count: 8, Loop time: 79.86580 ms
Loop count: 9, Loop time: 89.86692 ms
Loop count: 10, Loop time: 99.86628 ms
Loop count: 11, Loop time: 109.86648 ms
Loop count: 12, Loop time: 119.86582 ms
Loop count: 13, Loop time: 129.86613 ms
Loop count: 14, Loop time: 139.86586 ms
```

```
Loop count: 15, Loop time: 149.86605 ms
Loop count: 16, Loop time: 159.85953 ms
```

2. kernel_module

cf.

https://github.com/harcokuppens/xenomai3_rpi_gpio/blob/master/examples/xenomai3/kernel_modules/test-irq-in-kernel-mode-using-gpiolib/README.txt

Les fichiers sont dans l'archive `kernel_module.tar.bz2`

J'ai compilé directement sur le Pi1 mais il devrait être possible de faire de la cross-compilation si les user-space lib ont pu être cross-compilée (ce n'est pas le cas ici)

un make crée le fichier `button_toggles_led.ko`

```
sudo su
insmod ./button_toggles_led.ko
tail -11 /var/log/messages
```

```
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352182] irq number: 417
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352221] cpu: 0
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352236] gpio request in
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352271] gpio direction in
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352293] gpio request in
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352400] gpio direction out
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352430] set irq trigger: rising
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352511] irq request ,
IRQ_TYPE_EDGE_FALLING
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352572] globalCounter 0
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352591] handlerCPU: -1
Jun  3 09:09:32 raspberrypi kernel: [ 1858.352604] done
```