

INSTALLATION XENOMAI

1	INSTALLATION SUR LINUX.....	2
2	EXEMPLE UTILISATION.....	10

1 INSTALLATION SUR LINUX

Voici le guide permettant d'installer Xenomai sur linux.

1.1 Prérequis

Les paquets suivants sont à récupérer grâce à apt-get :

Git dh-autoreconf libncurses5 libncurses5-dev flex bison gcc

Noyau linux version 5.4.77, que l'on peut récupérer sur

<https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/linux-5.4.77.tar.xz>

Ubuntu 18.04

Ipipe_core , disponible à <https://xenomai.org/downloads/ipipe/v5.x/x86/ipipe-core-5.4.77-x86-2.patch>

Xenomai EDF est à récupérer sur le git suivant : <https://github.com/BraveMole/Xenomai-EDF.git>

Passer sur la branch final version

1.2 Installation

On se place dans le répertoire de Xénomai, et dans l'invite de commande on tape :

`./scripts/bootstrap`

Cela prépare le noyau xénomai avant la compilation, on doit obtenir ceci :

```
root@dell:/xeno_rtos/xenomai-3# ./scripts/bootstrap
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'config'.
libtoolize: copying file 'config/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'config'.
libtoolize: copying file 'config/libtool.m4'
libtoolize: copying file 'config/ltoptions.m4'
libtoolize: copying file 'config/ltugar.m4'
libtoolize: copying file 'config/ltversion.m4'
libtoolize: copying file 'config/lt~obsolete.m4'
configure.ac:80: installing 'config/compile'
configure.ac:66: installing 'config/config.guess'
configure.ac:66: installing 'config/config.sub'
configure.ac:68: installing 'config/install-sh'
configure.ac:105: installing 'config/missing'
demo/alchemy/Makefile.am: installing 'config/depcomp'
root@dell:/xeno_rtos/xenomai-3#
```

Ensuite, on applique le patch Ipipe sur le noyau Linux.

`scripts/prepare-kernel.sh --linux=../linux-5.4.77 --ipipe=../ipipe-core-5.4.77-x86-2.patch --arch=x86_64`

avec `--ipipe` le chemin pour trouver le fichier patch et `--linux` celui pour le noyau qui sera utilisé.

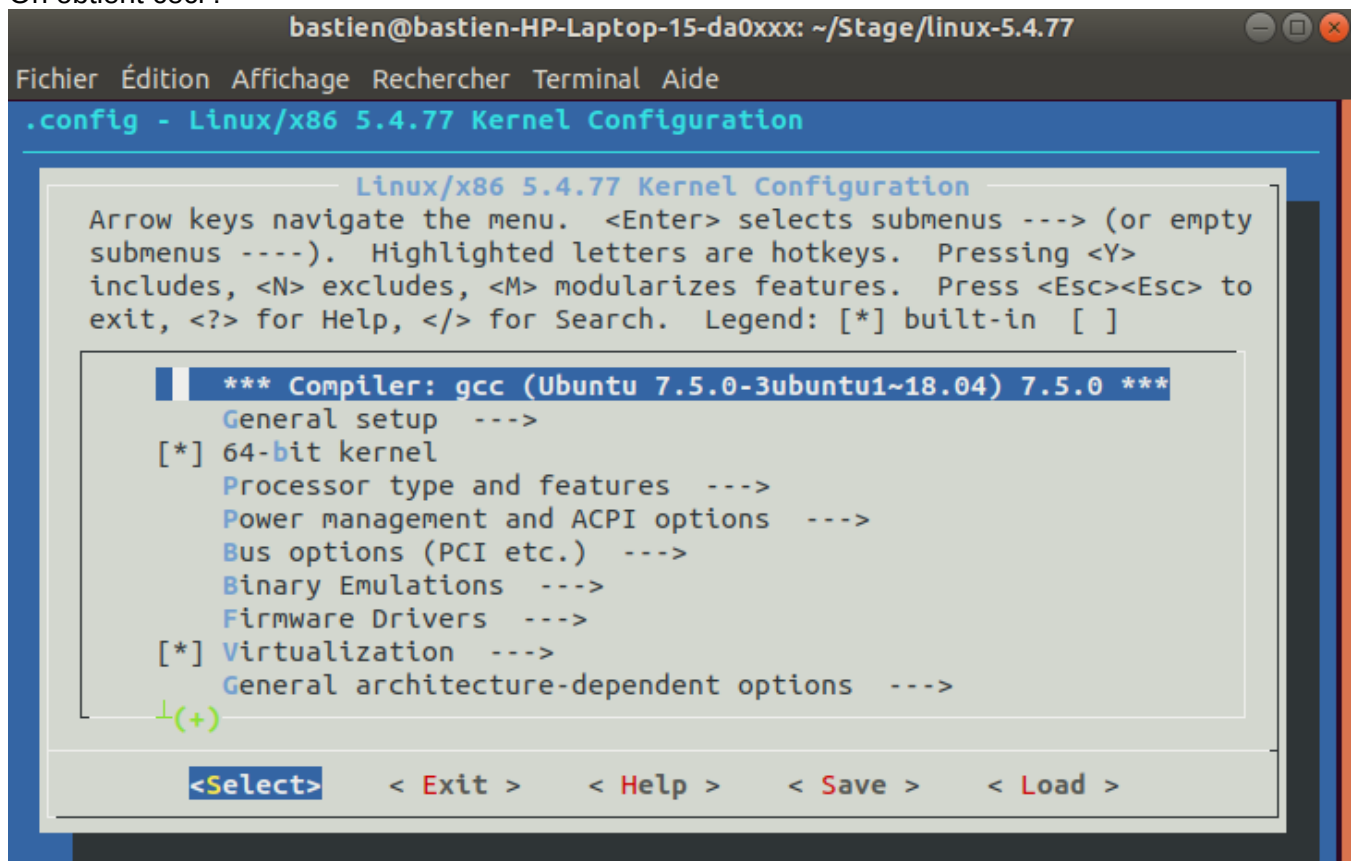
On doit obtenir ceci:

```
checking file kernel/trace/trace_clock.c
checking file kernel/trace/trace_functions.c
checking file kernel/trace/trace_functions_graph.c
checking file kernel/trace/trace_preemptirq.c
checking file lib/Kconfig.debug
checking file lib/atomic64.c
checking file lib/bust_spinlocks.c
checking file lib/dump_stack.c
checking file lib/ioremap.c
checking file lib/smp_processor_id.c
checking file mm/memory.c
checking file mm/mlock.c
checking file mm/mmu_context.c
checking file mm/mprotect.c
checking file mm/vmalloc.c
root@dell:/xeno_rtos/xenomai-3#
```

On se place alors dans le dossier du noyau linux pour configurer celui-ci avant compilation.

Dans l'invite de commande, taper make menuconfig.

On obtient ceci :



On peut alors Load la configuration en pièces jointes (ubuntu18.04Xenomai.config).

Voici la liste des modifications à apporter manuellement si le fichier n'est pas disponible :

* Processor type and features

--> Linux guest support (**exclude**)

--> CPU core priorities scheduler support (**exclude**)

* Power management and ACPI options

--> CPU Frequency scaling

--> CPU Frequency scaling (**exclude**)

--> ACPI (Advanced Configuration and Power Interface) Support

--> Processor (**exclude**)

--> CPU Idle

--> CPU idle PM support (**exclude**)

* Memory management options

--> Transparent Hugepage Support (**exclude**)

--> Contiguous Memory Allocation (**exclude**)

--> Allow for memory compaction (**exclude**)

--> Page Migration (**exclude**)

* Device Drivers

--> Input device support

--> Miscellaneous devices

--> PC Speaker support (**exclude**)

--> Staging drivers

--> Unisys SPAR driver support

--> Unisys visorbus driver (**exclude**)

Sauvegarder et quitter.

Il faut alors compiler le noyau linux .

Dans le répertoire linux :

sudo make -j8 (peut prendre plusieurs heures)

sudo make -j8 modules_install install

On peut alors installer xenomai sur le noyau linux.

Dans le répertoire de Xénomai

./configure && make && make install

On peut alors rebooter l'ordinateur, et choisir la version ubuntu modifiée avec xenomai.

On peut alors vérifier que l'installation s'est bien passée en réalisant un test de latence :

Il existe un utilitaire dans `usr/xeomai/bin` , que l'on peut lancer grâce à l'invite de commande :

```
# /usr/xenomai/bin/latency -t1 -p 1000
== Sampling period: 1000 us
== Test mode: in-kernel periodic task
== All results in microseconds
warming up...
RTT| 00:00:01 (in-kernel periodic task, 1000 us period, priority 99)
RTH|----lat min|----lat avg|----lat max|-overrun|---msw|---lat best|--lat worst
RTD|   -1.421|    8.716|   19.086|      0|    0|   -1.421|   19.086
RTD|   -2.142|    8.474|   19.048|      0|    0|   -2.142|   19.086
RTD|   -2.123|    8.615|   19.094|      0|    0|   -2.142|   19.094
RTD|   -2.130|    8.478|   19.969|      0|    0|   -2.142|   19.969
RTD|   -1.662|    8.694|   30.069|      0|    0|   -2.142|   30.069
RTD|   -1.637|    8.679|   29.371|      0|    0|   -2.142|   30.069
RTD|   -2.284|    1.019|   24.419|      0|    0|   -2.284|   30.069
RTD|   -2.278|   -2.055|   -1.286|      0|    0|   -2.284|   30.069
RTD|   -2.253|   -1.954|   -0.319|      0|    0|   -2.284|   30.069
RTD|   -2.248|   -1.911|   -0.851|      0|    0|   -2.284|   30.069
RTD|   -2.240|   -1.889|   -0.722|      0|    0|   -2.284|   30.069
RTD|   -2.226|   -1.867|   -0.975|      0|    0|   -2.284|   30.069
RTD|   -2.224|   -1.870|   -0.902|      0|    0|   -2.284|   30.069
RTD|   -2.213|   -1.859|   -0.987|      0|    0|   -2.284|   30.069
RTD|   -2.401|   -1.776|    2.124|      0|    0|   -2.401|   30.069
RTD|   -2.381|   -2.083|   -1.087|      0|    0|   -2.401|   30.069
RTD|   -2.381|   -2.069|   -1.175|      0|    0|   -2.401|   30.069
RTD|   -2.364|   -2.061|   -1.126|      0|    0|   -2.401|   30.069
RTD|   -2.358|   -2.049|   -1.160|      0|    0|   -2.401|   30.069
RTD|   -2.355|   -2.040|   -1.307|      0|    0|   -2.401|   30.069
RTD|   -2.329|   -2.021|   -0.876|      0|    0|   -2.401|   30.069
RTT| 00:00:22 (in-kernel periodic task, 1000 us period, priority 99)
RTH|----lat min|----lat avg|----lat max|-overrun|---msw|---lat best|--lat worst
RTD|   -2.323|   -2.022|   -1.222|      0|    0|   -2.401|   30.069
RTD|   -2.320|   -1.990|   -1.228|      0|    0|   -2.401|   30.069
RTD|   -2.308|   -1.991|   -0.749|      0|    0|   -2.401|   30.069
```

Si la latence est supérieure à 100 us, alors cela signifie que un problème ralentit l'exécution des tâches et que l'installation ne s'est pas passée correctement.

2 EXEMPLE UTILISATION

Pour créer une tâche avec une priorité dynamique, on utilise la commande :

`rt_task_create_dyna` (pointeur vers la tâche, nom de la tâche, taille de la pile, deadline relative en tick (1 secondes = 1^9 ticks, le mode de fonctionnement, classement 0)

Le fichier test correspondant est `periodicTaskTest1.c`

```
rt_task_create_dyna(&loop_task_prio, "prio", 0, 5 000 000, 0, 0); //Deadline in ticks
rt_task_create_dyna(&loop_task_non_prio, "non_prio", 0, 10 000 000, 0, 0);

//Since task starts in suspended mode, start task
rt_task_start(&loop_task_prio, &loop_task_proc, 0);
rt_task_start(&loop_task_non_prio, &loop_task_proc, 0);
```

On a donc deux tâches périodiques, avec la tâche `prio` qui sera toujours prioritaire sur la tâche `non_prio`

```
bastien@bastien-HP-Laptop-15-da0xxx:~/Stage/testsApplisXenomai/TestPeriodicTask$
./periodicTask
Starting cyclic task...
Starting task prio with period of 1s ....
Task name: prio , Loop count: 0, Loop time: 1000.00564 ms
Task name: non_prio , Loop count: 0, Loop time: 1000.01755 ms
Task name: prio , Loop count: 1, Loop time: 2000.00762 ms
Task name: non_prio , Loop count: 1, Loop time: 2000.01262 ms
Task name: prio , Loop count: 2, Loop time: 3000.02221 ms
Task name: non_prio , Loop count: 2, Loop time: 3000.02415 ms
Task name: prio , Loop count: 3, Loop time: 4000.00458 ms
Task name: non_prio , Loop count: 3, Loop time: 4000.01129 ms
Task name: prio , Loop count: 4, Loop time: 5000.01278 ms
Task name: non_prio , Loop count: 4, Loop time: 5000.01351 ms
Task name: prio , Loop count: 5, Loop time: 6000.01427 ms
Task name: non_prio , Loop count: 5, Loop time: 6000.01658 ms
Task name: prio , Loop count: 6, Loop time: 700.00751 ms
Task name: non_prio , Loop count: 6, Loop time: 7000.01514 ms
Task name: prio , Loop count: 7, Loop time: 8000.00465 ms
Task name: non_prio , Loop count: 7, Loop time: 8000.01277 ms
```