

Noyau linux - suite - les modules

Objectifs du TP :

1. Ecrire l'Hello World d'un module ;
2. Le tester sous Qemu.

1 Code source

Voici le code source d'un module basique, `hello.c`.

```

/*
 * hello.c - Utilisation des macros module_init() et module_exit().
 * Cette méthode est préférée à l'utilisation directe de
 * init_module() et cleanup_module().
 */
#include <linux/module.h>      /* Pour tous les modules. */
#include <linux/kernel.h>     /* Pour KERN_INFO. */
#include <linux/init.h>       /* Pour les macros. */

static int __init hello_init(void)
{
    printk(KERN_INFO "Salut le monde !\n");
    return 0;
}

static void __exit hello_exit(void)
{
    printk(KERN_INFO "Do vidjenja !\n");
}

module_init(hello_init);
module_exit(hello_exit);

```

Question 1.1. *Rappelez pourquoi la fonction `printk` est utilisée et non `printf`.*

2 Compilation

Créez un répertoire `hello` au niveau de la racine des sources de votre noyau et placez-y le code `hello.c` donné ci-dessus. Voici le `Makefile` à placer également dans le répertoire `hello` ; notez que seule la première ligne en fait vraiment nécessaire (vous auriez pu autrement exécuter la commande à la main) :

```

obj-m += hello-1.o

all:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

```

Compilez le module ... `make`.

3 Chargement et déchargement

Il est temps de charger le module. Il est bien entendu nécessaire d'être root pour réaliser cette opération. Au choix, craquez la machine hôte ou réalisez l'opération sous Qemu (le second choix est conseillé). Téléchargez le module dans votre image Qemu puis insérez-le dans le noyau :

```
insmod hello.ko
```

Question 3.1. Vérifiez ce qu'il s'est passé en regardant le fichier `/var/log/messages` :

```
tail /var/log/messages
```

Enfin, déchargez-le :

```
rmmod hello
```

4 Licence et documentation du module

Lors du chargement de votre module, vous avez sans doute remarqué un avertissement du type suivant :

```
kernel: hello: module license 'unspecified' taints kernel.
```

En effet, tout module non déclaré libre est considéré *a priori* comme propriétaire. Utilisez la macro suivante dans le source de votre module pour attribuer un licence GPL au module :

```
MODULE_LICENSE( "GPL" );
```

De même, référez votre nom et donnez une description du module avec les macros `MODULE_AUTHOR()` et `MODULE_DESCRIPTION()`. Utilisez la commande `modinfo` pour vérifier que cela a bien été pris en compte.

A Rendons à César ...

Ce document est inspiré de "The Linux Kernel Module Programming Guide" : <http://www.tldp.org/LDP/1>