

TD n° 3

Modules

1 Mon premier module: « Hello World »

1.1 Création du module « Hello World »

Mettre en place un module « Hello World » qui présente les caractéristiques suivantes :

- Affiche un message au chargement (« Hello World !») et au déchargement du module (« Gooby »)
- Prend un paramètre `n` entier en entrée qui répète le message d'entrée `n` fois
- Prend un paramètre `str` chaîne de caractère qui concatène celle-ci au message de sortie du module (par exemple « cruel world ! »)

Créer un `Makefile` permettant de compiler ce module. Tester ce module à l'aide de la commande `insmod`.

1.2 Chargement de module

Faites en sorte que le module précédent se charge à l'aide de la commande `modprobe`. Quand vous aurez trouvé les opérations qui le permettent, ajouter une entrée à votre `Makefile` de façon à automatiser la procédure.

Modifiez la configuration du système afin que, lors de son chargement avec la commande `modprobe`, le module ne soit plus chargé avec les valeurs par défaut de ses paramètres, mais avec d'autres valeurs de ces deux paramètres. Tester que tout fonctionne correctement.

1.3 Ajout de fonctionnalités au module « Hello World »

Ajouter les fonctionnalités suivantes à votre module initial :

- Affiche un message supplémentaire au chargement du module : le numéro de version du noyau sur lequel le module est chargé. On prendra soin de ne pas utiliser la macro C présente dans les sources ce qui aurait pour résultat de toujours afficher la version du noyau avec lequel le module a été compilé. Vous veillerez donc à utiliser une fonction.

Voir `linux/utsname.h`

- Affiche le temps écoulé depuis le chargement du module lorsqu'on le décharge. Vous pouvez utiliser la fonction `do_gettimeofday()` pour réaliser cela.

2 Dépendances entre modules

2.1 Module « Hello »

Créez un module `hello` qui :

- Affiche son nom lors de son chargement
- Affiche son nom lors de son déchargement
- Offre un service `print_hello()` qui consiste à afficher la chaîne "Hello" (export d'un symbole)

2.2 Module « World »

Créez un module `world` qui :

- est dépendant du module `hello` précédent

TD n° 3 Modules

- lors de son chargement :
 - affiche son nom
 - utilise le service `print_hello()`
 - affiche `"world !"`
- lors de son déchargement :
 - affiche son nom

2.3 Test de dépendances

Installer les modules et configurer le système afin de vérifier que quand vous chargez le module `world`, le noyau charge bien automatiquement le module `hello`.

3 Visualisation, création et modification d'une image `initrd`

Les images `initrd` contiennent un système de fichier compressé qui est chargé en mémoire par le noyau lors de son démarrage (avant le montage de la partition racine du système de fichier). Nous allons essayer de comprendre son fonctionnement en visualisant son contenu, en le modifiant et en essayant de charger celui-ci.

3.1 Visualisation du contenu d'une image `initrd`

Commencez par copier l'image `initrd` du noyau installé sur votre machine. Renommez le fichier en `.gz`. Vous pouvez maintenant décompresser ce fichier. Pour explorer le contenu du fichier, vous pouvez le désarchiver avec la commande `cpio`:

```
mkdir initrd
cd initrd
gzip -d -c /boot/initrd.img-3.2.0-4-686-pae | cpio -id
```

Vous avez maintenant accès à tout le contenu de ce fichier dans le répertoire.

3.2 Modification

Tentez de modifier le contenu de ce fichier `initrd` afin de le rendre plus compacte (supprimez des modules qui ne vous semblent pas nécessaires au démarrage).

3.3 Création d'un fichier `initrd`

3.3.1 Création manuelle

Pour recréer votre fichier, vous pourrez utiliser la commande suivante (attention de ne pas inclure le fichier `img` de départ). Pensez à compresser votre fichier à l'aide de `gzip` après avoir créé l'archive avec `cpio`.

```
find . | cpio -o -H newc | gzip > ../newinitrd.img-3.2.0-4-686-pae
```

Tentez de redémarrer votre système avec ce nouvel `initrd` pour voir si vous avez bien une image fonctionnelle pour votre système (et plus petite que l'image précédente).

3.3.2 Création avec `mkinitrd`

Utilisez la commande `mkinitramfs` pour créer votre fichier `newinitrd.img-3.2.0-4-686-pae`.

```
mkinitramfs -r dossier -o ../newinitrd.img-3.2.0-4-686-pae
```