



MA-CSEL1 (Construction de systèmes embarqués sous Linux) – Exercices
Programmation noyau : Modules noyau

Module noyau

1. Générez d'un module noyau « out of tree » pour la cible NanoPi :
 - a. Créez le squelette d'un module noyau et générez-le en dehors des sources du noyau à l'aide d'un Makefile. Le module devra afficher un message lors de son enregistrement et lors de sa désinstallation.
 - b. Testez sur la machine hôte la commande « modinfo » sur votre squelette de module et comparez les informations retournées avec celles du code source.
 - c. Installez le module (insmod) et contrôlez le log du noyau (dmesg)
 - d. Comparez les résultats obtenus par la commande « lsmod » avec ceux obtenus avec la commande « cat /proc/modules »
 - e. Désinstallez le module (rmmod).
 - f. Adaptez le Makefile du module pour autoriser l'installation du module avec les autres modules du noyau permettant l'utilisation de la commande « modprobe ». Le module devra être installé dans le root filesystem utilisé en nfs par la cible.
2. Adaptez le module de l'exercice précédent afin qu'il puisse recevoir deux ou trois paramètres de votre choix. Ces paramètres seront affichés dans la console. Adaptez également le rootfs afin de pouvoir utiliser la commande « modprobe ».
3. Trouvez la signification des 4 valeurs affichées lorsque l'on tape la commande « cat /proc/sys/kernel/printk »

Gestion de la mémoire, bibliothèque et fonction utile

4. Créez dynamiquement des éléments dans le noyau. Adaptez un module noyau, afin que l'on puisse lors de son installation spécifier un nombre d'éléments à créer ainsi qu'un texte initial à stocker dans les éléments précédemment alloués. Chaque élément contiendra également un numéro unique. Les éléments seront créés lors de l'installation du module et chaînés dans l'une liste. Ces éléments seront détruits lors de la désinstallation du module. Des messages d'information seront émis afin de permettre le debugging du module.

Accès aux entrées/sorties

5. A l'aide d'un module noyau, afficher le Chip-ID du processeur, la température du CPU et la MAC adresse du contrôleur Ethernet.
 - a. Les 4 registres de 32 bits du Chip-ID sont aux adresses 0x01c1'4200 à 0x01c1'420c
 - b. Le registre de 32 bits du senseur de température du CPU est à l'adresse 0x01c2'5080
 - c. Les 2 registres de 32 bits de la MAC adresse sont aux adresses 0x01c3'0050 et 0x01c3'0054



MA-CSEL1 (Construction de systèmes embarqués sous Linux) – Exercices
Programmation noyau : Modules noyaux

Pour obtenir la température du CPU, il faut, après lecture du registre, appliquer la formule suivante : $température = -1191 * register_value / 10 + 223'000$

Avant d'accéder aux registres du Chip-ID, veuillez réserver la zone mémoire correspondante aux registres du µP. Validez cette réservation à l'aide de la commande « `cat /proc/iomem` ».

La commande « `cat /sys/class/thermal/thermal_zone0/temp` » permet de valider la bonne lecture de la température. La commande « `ifconfig` » permet de valider la bonne lecture de la MAC adresse.

Threads du noyau

6. Développez un petit module permettant d'instancier un thread dans le noyau. Ce thread affichera un message toutes les 5 secondes. Il pourra être mis en sommeil durant ces 5 secondes à l'aide de la fonction « `ssleep(5)` » provenant de l'interface `<linux/delay.h>`.

Mise en sommeil

7. Développez un petit module permettant d'instancier deux threads dans le noyau. Le premier thread attendra une notification de réveil du deuxième thread et se remettra en sommeil. Le 2^{ème} thread enverra cette notification toutes les 5 secondes et se rendormira. On utilisera les `waitqueues` pour les mises en sommeil. Afin de permettre le debugging du module, chaque thread affichera un petit message à chaque réveil.

Interruptions

8. Développez un petit module permettant de capturer les pressions exercées sur les switches de la carte d'extension par interruption. Afin de permettre le debugging du module, chaque capture affichera un petit message.

Quelques informations pour la réalisation du module :

- a. Acquérir la porte GPIO avec le service
`gpio_request (<io_nr>, <label>);`
- b. Obtenir le vecteur d'interruption avec le service
`gpio_to_irq (<io_nr>);`
- c. Informations sur les switches de la carte d'extension
 - i. k1 - gpio: A, pin_nr=0, io_nr=0
 - ii. k2 - gpio: A, pin_nr=2, io_nr=2
 - iii. k3 - gpio: A, pin_nr=3, io_nr=3