OSEM

Compte Rendu TP3

David Toty (3000755) & Maxime Tran (3000738)

Dans ce TP nous allons nous intéresser aux code noyau plus précisément ajouter un service système et modifier un driver.

1. Installation et compilation d'une application utilisateur

Dans un premier temps nous récupérons les sources du noyau ALMOS via un scp.

La première chose est de « sourcer» deux fichiers SourceMe :

Le premier est dans le dossier almos-mk et le deuxième dans le dossier source.
Dans le deuxième il faut modifier le TO_BE_SET à **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk**

On peut dores et déjà tester le système, il y a un programme *hello* configurer dans **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/usr/apps/hello** , on le compile avec make TARGET=tsar et pour l'exécuter on utilise la commande **./exec.sh** dans le répertoire **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk**

2.Ajout d'un service système affichant la date (compteur de cycles)

Le but est de faire un service système kloc, dont le fonctionnement est identique au service clock.

Dans un premier temps nous créons le fichier kloc.c dans **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/usr/libs/src/dietlibc**

Le contenu de ce fichier sera le même que celui de clock.c à l'exception du numéro (nom) de service : ici *SYS_KLOC*.

Ensuite nous devons ajouter dans le fichier **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/usr/libs/src/dietlibc/include/sys/syscall.h** l'entrée de ce service (numéro) : *SYS_KLOC*

Ensuite rajouter sys_kloc dans le tableau sys_call_tbl du fichier : **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/kernel/kern/do_syscall.c**

Il nous reste maintenant à compiler les modifications efféctuées dans le *Makefile* de l'espace utilisateur et dans celui du kernel :
make :
/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/usr/libs/src/Makefile
/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/kernel/Makefile

Par la suite, modifions **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/kernel/kern/time.h** pour y ajouter *int sys_kloc (uint64_t *val)*; (on peut constater que int sys_clock est présent).

Il faut également rajouter l'entrée *SYS_KLOC* dans l'enum de **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/kernel/kern/syscall.h**

Dans le fichier **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/kernel/kern/sys_clock.c** il faut faire en sort d'afficher le nombre de cycle :
printk (INFO, «Nombre de cycles : %ld\n»,cycles) ;
avec *cycles = cpu_get_cycles(current_cpu);* (comme ce qui est déjà présent).

Tout est prêt pour une éxécution. Pour tester, on peut par exemple modifier le programme hello vu dans la partie 1 et remplacé :
printf("hello world at %u\n", clock());

par :

printf("hello world at %u\n", kloc());

3. Modification d'un driver

Le but de cette partie est de modifié le driver **/dsk/l1/misc/toty/almos-mk-tp3/almos-mk/source/kernel/drivers/soclib/soclib_dma.c** pour retirer sa liste de fragments.

Les fragments sont définis par : *dev_request_t *frag;*

Il faut recalculer la liste de fragments à chaque appel de requêtes sur le devices.