

Systèmes d'exploitation (SYE)

Profs Daniel Rossier, Alexandre Corbaz, Fiorenzo Gamba
Assistants : Anthony Jaccard, Basile Cuneo, Jean-Pierre Miceli

Shell, ISR et version de noyau

lab03 (Semaine du 02.10.23)

Objectifs de laboratoire

Il s'agit d'un laboratoire d'introduction à l'environnement de travail pour les laboratoires SYE. Dans ce cadre, chaque étudiant-e aura l'occasion de se familiariser plus avant avec le système d'exploitation SO3 et ses différentes applications.

Validation du laboratoire

Ce laboratoire sera à rendre selon la méthode vue lors du précédent laboratoire, à savoir

1. Ajouter les fichiers modifiés avec "*git add <vos_fichiers>*"
2. Commiter les changements avec "*git commit -m '<message de commit>'*"
3. Envoyer les changements sur votre repo gitlab avec "*git push -u origin*"

Etape 1 - Modification du shell

L'espace utilisateur définit l'ensemble du code qui s'exécute lorsque le processeur fonctionne en mode « *user* ». Cet espace comprend l'ensemble des applications de type *utilisateur*. Une application de base est le *shell* évoqué précédemment.

Cette étape consiste à modifier le *shell* afin que celui-ci affiche la liste des *tokens* à chaque entrée d'une commande. Par *token*, on entend chaque élément de la chaîne de caractères (entrée par l'utilisateur) séparée par un espace.

- a) Editer le fichier « *usr/src/sh.c* » et analyser le code de la fonction principale (« *main()* »).
- b) Compléter le fichier afin d'afficher la liste des *tokens* après chaque entrée de commande. L'affichage se présentera comme les exemples ci-dessous :

```
so3% echo a bc de
echo
a
bc
de
a bc de ← Résultat de l'application
so3% ls
ls ← Token (unique)
cat.elf
dev/
echo.elf
lorem.txt
ls.elf
more.elf
```

```
sh.elf
so3%
```

- ⇒ Afin de bien visualiser les erreurs de compilation, il est préférable d'effectuer la commande « *./build.sh* » dans le répertoire « *usr/* » avant de lancer *make* dans le répertoire principal. Il est de même pour la compilation du noyau (répertoire « *so3/* »)
- ⇒ Afin d'afficher les lignes sur le terminal, il ne faut pas oublier de terminer les chaînes de caractères par un retour à la ligne « *\n* ».

Etape 2 - Modification de la version du noyau

La version actuelle du noyau est le numéro **2023.6.0**. Afin de poursuivre l'exploration du noyau, il est proposé de changer la version du noyau à **2023.6.1**. Pour rappel, le numéro de version est affiché au démarrage du noyau.

- a) Editer le fichier du noyau contenant le numéro de version et définir la version **2023.6.1**
 - b) Compiler et vérifier le numéro de version à l'affichage.
- ⇒ L'utilisation de l'outil **grep** de la suite GNU permet la recherche d'un *pattern* dans les fichiers spécifiés. La commande ci-dessous va afficher toutes les occurrences de « *toto* » présentes dans tous les fichiers du dossier courant et des sous-répertoires (fouille récursive).

```
reds@reds2023:~/sy23_student$ grep -r toto
```

- ⇒ Les pages « *man* » fournissent de l'aide sur les commandes ; voici un exemple d'utilisation :

```
reds@reds2022:~/sy23_student$ man grep
```

Etape 3 - Modification de la routine de service (ISR) du *timer*

Dans SO3, le *timer* *arm_timer* est programmé à une fréquence de 80 Hz. Autrement dit, une interruption *timer* de type IRQ est générée au processeur toutes les 12.5 millisecondes.

- a) Démarrer SO3 en utilisant le *debugger* au niveau du noyau.
- b) Insérer un *breakpoint* à l'entrée de la routine de service (ISR) du *timer* dans le fichier « *so3/devices/timer/arm_timer.c* ».
- c) A gauche de l'environnement, examiner le chemin d'appel des fonctions (*backtrace*) au moment de l'arrêt du programme sur le *breakpoint*.
- d) Modifier la routine de service afin d'afficher toutes les secondes un compteur qui donne le nombre d'appels à cette routine.

- ⇒ Dans le noyau, la fonction « *printf()* » est remplacée par la fonction « *printk()* ».
- ⇒ N'hésitez pas à vous aider des variables déjà présentes.
- ⇒ Pour afficher une variable en 64bits, utiliser le formatteur *%lld*.

