

Systèmes d'exploitation (SYE)

Profs Daniel Rossier, Alexandre Corbaz, Fiorenzo Gamba
Assistants : [Anthony Jaccard](#), [Basile Cuneo](#), [Jean-Pierre Miceli](#)

Appels systèmes

lab04 (Semaine du 09.10.2023)

Objectifs de laboratoire

Ce laboratoire se focalise sur la notion d'appel système au niveau de l'espace utilisateur et du noyau. Il permettra de se familiariser avec la chaîne d'appels permettant d'exécuter du code noyau afin d'obtenir certaines informations ou fonctionnalités de la part de celui-ci

Récupération et rendu du laboratoire

Pour récupérer la branche de ce laboratoire, utilisez les commandes:

1. `"git fetch upstream"`
2. `"git checkout lab04"`

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"` ou `"git push --set-upstream origin"`

Etape 1 - Rajout d'un appel système dans SO3

Cette étape permettra l'introduction d'un nouvel appel système - appelé (directement) **sys_fork2** - permettant la création d'une copie de processus. Cet appel système sera une copie de l'appel système **sys_fork** avec un affichage d'informations supplémentaire. Ce qui nous intéresse surtout ici est le cheminement pour rajouter un nouvel appel système dans SO3.

- a) En vous basant sur la théorie donnée lors du cours et des appels systèmes déjà implémentés dans SO3, rajoutez l'appel système `sys_fork2`. Le code de ce dernier est exactement le même que le code de l'appel système `sys_fork`. Il vous est demandé d'afficher un message provenant du kernel lors de l'appel à `sys_fork2`. Ce message aura la forme suivante (merci de respecter ce format):

```
fork2() -> process test_fork2.elf with pid 2 forked to child with pid 3
```

⇒ L'équivalent de la fonction « **printf()** » dans le kernel est « **printk()** ».

⇒ La liste des fichiers à modifier est :

- o `so3/include/syscall.h`
- o `so3/include/process.h`
- o `so3/kernel/process.c`
- o `so3/kernel/syscalls.c`
- o `usr/lib/libc/include/syscall.h`
- o `usr/lib/libc/crt0.S`

À vous de trouver les modifications à faire et dans quel ordre.

- ⇒ Il est possible de tester le fonctionnement du syscall **sys_fork2** en exécutant le programme *test_fork2*.
- b) Dans le fichier *rapport.md*, expliquez à quoi servent les différents ajouts effectués dans les fichiers modifiés et le cheminement complet pour ajouter un appel système.

