

Systemes d'exploitation (SYE)

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

Processus

lab05 (Semaine du 16.10.2023)

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 davantage avec le code du *shell* et de l'ensemble des fichiers qui composent SO3. De plus, le laboratoire consacre une étape sur la création de processus et le chargement d'images binaires.

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 – Exécution de commandes en arrière-plan dans sh

Dans cet exercice, il vous est demandé d'implémenter la gestion du symbole "&" dans le shell de SO3. Dans linux, ce symbole permet d'exécuter la commande qui le précède en arrière-plan. Elle pourra donc continuer à s'exécuter tout en permettant que d'autres commandes soient lancées en parallèle

Dans la fonction `"process_cmd()"` de `"usr/src/sh.c"`, rajoutez du code afin de lancer un nouveau processus qui exécutera la commande entrée sans paralyser le shell durant son exécution

Note: Le symbole "&" sera le dernier paramètre possible d'une commande. Les paramètres entrés après devront être ignorés

Une fois cela fait, vous pouvez tester cette fonctionnalité en utilisant la commande `"sleep <s>"` avec "s" le nombre de seconde que vous souhaitez faire "dormir" le programme. Exécutez-la une première fois sans utiliser "&" et constatez que le shell est inopérant tant que la commande n'est pas terminée. Réessayez maintenant en utilisant le symbole "&" et tentez d'utiliser une autre commande (par exemple `"ls"`) dans le temps de sommeil de la commande précédente. Vous devriez être capable de lancer et d'observer les sorties de la nouvelle commande