

Preliminaires

De base

Si l'un de ces points est vrai, alors la soutenance est immédiatement terminée, et le groupe a 0.

- Pas de rendu (= rien dans le dépôt git)
- Pas de fichier auteur
- Fichier auteur invalide
- Faute de norme (par la norminette)
- Triche ou fonctions interdites

Durant le reste de la soutenance, si le programme a un comportement inapproprié (segfault, buserror, ...) la soutenance s'arrête.

✔ Yes

✘ No

Le shell

Testons les principales features de votre sh1.

Fork et execve

Fork et Execve sont au coeur de la base d'un shell minimaliste, tel que le ft_sh1.

Si il n'y a pas ces 2 appels système, c'est qu'il y a un problème de compréhension du sujet. Avec la liste de fonctions autorisées, il n'y a pas d'autre solution.

✔ Yes

✘ No

Les builtins

Les fonctionnalités précédentes correspondent plus à un programme capable de faire un fork et un execve en boucle qu'à un shell digne de ce nom. Un shell, même préhistorique, doit proposer certaines fonctionnalités.

- cd
- cd /path/to/dir
- exit

✔ Yes

✘ No

Gestion de l'environnement

les builtins

- env
- setenv
- unsetenv

fonctionnent, modifient l'environnement et passent le bon environnement aux programmes qui sont exécutés.
(vérifier par ex avec /usr/bin/env au minimum)

✔ Yes

✘ No

Gestion du PATH

L'exécution des binaires avec un path complet fonctionne.

('/bin/ls' par ex)

L'exécution des binaires dans les répertoires du PATH fonctionne

('ls' par ex)

Essayer des binaires dans chaque répertoire que comporte le PATH.

✔ Yes

✘ No

Shell fondamental

Votre ft_sh1 doit pouvoir assurer ces fonctionnalités de base :

- un prompt
- Une commande simple, par exemple un simple ls
- le shell attend la fin d'exécution de la commande lancée pour re-afficher le prompt.
- emacs (ou vim pour ceux qui ont des goûts douteux) fichier.c
- ls /bin /boot /sbin /dev /home
- /bin/ls (tester une commande avec un PATH complet)
- commande inconnue (comme trucmachin) 'Command not found' (ou un message d'erreur similaire)
- commande connue mais pas dans le PATH

Rappel : pas de pipe, ni de redirection.

✔ Yes

✘ No

Gestion de la ligne de commande

Les shell suivants, dont le 42sh, seront plus appropriés pour des parser-lexer et autres façons compliquées d'analyser la ligne de commande.

En revanche, un minimum de choses doit marcher.

- Des espaces et/ou des tabulations un peu partout dans la ligne de commande ne perturbent pas l'exécution de cette commande, de manière similaire à sh, bash ou zsh
- Une ligne de commande vide ne provoque pas d'erreur
- Une ligne de commande vide avec des espaces ne provoque pas d'erreur

✔ Yes

✘ No

Bonus

Signal

Ctrl-c ça marche ? ça arrête le programme ?

✔ Yes

✘ No

Droits dans le PATH

Votre ft_sh1 cherche-t-il à exécuter des fichiers qui sont dans le PATH mais qui sont pas des binaires ? (avec les droits d'exécution donc). Si c'est le cas, vous n'avez pas gagné les points et répondez alors non à cette question.

✔ Yes

✘ No

Completion

La completion marche-t-elle ?

✔ Yes

✘ No

point virgule

On peut séparer plusieurs commandes sur une même ligne avec un point-virgule.

✔ Yes

✘ No

Autres features

D'autres bonus ?

par ex. gestion du tilde, une version simple de l'édition de la ligne (déjà un backspace, éventuellement des flèches, ...)

A noter que pas mal de features seront au menu du sh2, sh3, et 42sh.