

Attachments

☐ Subject (<https://cdn.intra.42.fr/pdf/pdf/183/42sh.fr.pdf>)

file:///Users/cboussau/Desktop/42sh.htm

Testez l'existence et le fonctionnement du heredoc "«<"; avec ses options et cas particuliers.

☐ Yes

☐ No

Aggrégation des file descriptors

Testez l'existence et le fonctionnement de l'aggrégation des file descriptors.

N'oubliez pas de tester les cas d'erreur (file descriptor invalide ou récemment fermé) et l'argument "<" qui ferme le file descriptor spécifié avant l'opérateur de redirection.

☐ Yes

☐ No

Le globbing

Testez la presence et le bon fonctionnement du globbing ("*", "?*", "[?]", "[?]", "[?]", etc).

Regardez dans les sources l'implémentation du globbing, la fonction glob(3) ne doit pas être utilisée.

☐ Yes

☐ No

Back quotes

Testez la presence et le bon fonctionnement de la substitution de commande avec les back quotes "``".

La commande exécutée doit être dans un sous-shell. Testez donc que l'environnement d'exécution du shell principal n'est pas modifié par l'exécution de la commande substituée (en lançant une commande qui modifie l'environnement, par exemple).

☐ Yes

☐ No

Sous-shells

Testez la presence et le bon fonctionnement des sous shells "({)".

Pour cela, testez par exemple une builtin qui modifie l'environnement, et vérifiez que l'environnement n'est pas modifié dans le shell principal.

Testez également avec des commandes complexes.

☐ Yes

☐ No

Variables locales

Testez la presence et le bon fonctionnement des variables locales et des builtins unset et export.

N'oubliez pas de tester les cas d'erreur (arguments invalides, non-existants, variable précédemment unset, etc).

☐ Yes

☐ No

Historique

Testez la presence et le bon fonctionnement de l'historique des commandes et des builtins history et "!" avec toutes leurs options.

☐ Yes

☐ No

Edition de ligne

Testez la presence et le bon fonctionnement de l'édition de ligne telle que demandée dans le ft_sh3.

Merci de vous référer au sujet du ft_sh3 pour le détail des fonctionnalités attendues (partie obligatoire du ft_sh3 uniquement).

Les étudiants peuvent avoir choisi de mapper différemment les fonctionnalités aux touches clavier, ça n'a pas vraiment d'importance tant que le tout est cohérent.

☐ Yes

☐ No

Manipulation de fichiers

Testez l'existence et le bon fonctionnement de la builtin read avec toutes ses options. (au minimum, l'option < doit être présente)

Si le 42sh gère le shell script, pensez à vérifier le comportement de la builtin en mode non-interactif.

☐ Yes

☐ No

Complétion dynamique

Testez la presence et le bon fonctionnement de la complétion dynamique.

Au minimum, la complétion doit être fonctionnelle avec des fichiers et des dossiers, dans des sous-dossiers du répertoire courant.

☐ Yes

☐ No

Job control

Testez la presence et le bon fonctionnement du Job control et des builtins job, fg, bg, (avec toutes leurs options !) et de l'opérateur "&".

Comme d'habitude, testez d'abord toutes les options des builtins, en testant les cas d'erreur (arguments invalides, etc).

Testez également en envoyant divers signaux aux processus en background. Est-ce que leur status est correctement mis à jour dans le shell ?

Essayez de lancer une commande en background qui lit sur l'entrée standard. Le processus doit être interrompu par un signal généré par le TTY (un processus en background ne doit pas accéder au terminal de session). C'est bien le cas ?

Faites ensuite des tests avec quelques cas complexes, comme des pipelines ou les opérateurs logiques. Quel comportement a été choisi par les étudiants ? Est-ce que ce comportement est justifié et cohérent ?

Si le 42sh gère le shell script, pensez à vérifier le comportement de l'opérateur "&" avec un shell non-interactif.

☐ Yes

☐ No

Shell script

Testez la presence et le bon fonctionnement du shell script, c'est-à-dire l'implémentation d'un vrai langage de script shell dans le 42sh. Il ne s'agit pas de simplement "lire des fichiers avec des commandes" pour parler de shell script.

Testez d'abord que le parsing est sans failles : mettez des indentations et des sauts de ligne aléatoirement dans vos scripts de test (tout en respectant la syntaxe du langage shell, évidemment) et vérifiez que ça ne pose pas de problème.

Le shell script doit au minimum :

- implémenter les structures de contrôle (boucles, conditions)
- gérer les commandes en mode non-interactif
- gérer des variables

Vérifiez méticuleusement ces points.

☐ Yes

☐ No

Des bonus

D'autres fonctionnalités

Si le 42sh a des fonctionnalités supplémentaires, comptabilisez-les ici. Vous pouvez comptabiliser jusqu'à 5 fonctionnalités bonus. Les bonus doivent être 100% fonctionnels et ne pas mettre en cause la stabilité du shell.

5

Rate it from 0 (failed) through 5 (excellent)

Ratings

Don't forget to check the flag corresponding to the defense

☒ Ok

☐ Empty work

☐ Incomplete work

☐ No author file

☐ Invalid compilation

☐ Norme

☐ Cheat

☐ Crash

Conclusion

Leave a comment on this correction

* (required) Comment

Finish correction

