SCALE FOR PROJECT MINISHELL

Partie obligatoire

Compilation

- Utilisez "make -n" to pour vous assurer que le projet compile avec "-Wall -Wextra -Werror". Si ce n'est pas le cas, cochez le flag "invalid compilation".
- Le minishell compile sans aucune erreur. Si ce n'est pas le cas, cochez le flag.
- Le Makefile ne doit pas re-link. Si ce n'est pas le cas, cochez le flag.

Commande simple et variables globales

- Exécutez une commande simple avec un PATH absolu tel que /bin/ls ou n'importe quelle autre commande sans option.
- Combien de variables globales y a-t-il ? Pourquoi ? Demandez à la personne évaluée de vous donner un exemple concret pour démontrer que leur usage est obligatoire et cohérent.
- Vérifiez la variable globale. Celle-ci ne peut contenir une autre information ou donner l'accès à autre chose que la valeur d'un signal reçu.
- Testez une commande vide.
- Testez seulement des espaces et des tabs.
- En cas de crash, cochez le flag "crash".
- Si quelque chose ne marche pas, cochez le flag "incomplete work".

Arguments

- Exécutez une commande simple avec un PATH absolu tel que /bin/ls ou n'importe quelle autre commande, avec option mais sans '' (single quotes) ni "" (double quotes).
- Répétez ce test plusieurs fois avec différentes commandes et différents arguments.
- En cas de crash, cochez le flag "crash".
- Si quelque chose ne marche pas, cochez le flag "incomplete work".

---

echo

- Lancez la commande echo avec et sans argument ou options, ou avec l'option -n.
- Répétez ce test plusieurs fois avec différents arguments.
- En cas de crash, cochez le flag "crash".
- Si quelque chose ne marche pas, cochez le flag "incomplete work".

exit

- Lancez la commande exit avec et sans arguments.
- Répétez ce test plusieurs fois avec différents arguments.
- N'oubliez pas de relancer le minishell.
- En cas de crash, cochez le flag "crash".
- Si quelque chose ne marche pas, cochez le flag "incomplete work"

Valeur de retour d'un processus

- Exécutez des commandes simple avec un chemin absolu tel que /bin/ls ou n'importe quelle autre commande avec des arguments mais sans '' (single quotes) ni "" (double quotes), puis lancez "echo $?". Vérifiez la valeur affichée. Vous pouvez le refaire dans bash et comparer.
- Répétez ce test plusieurs fois avec différentes commandes et différents arguments.
- Utilisez des commandes qui ne fonctionnent pas telles que '/bin/ls fichiernul'.
- Essayez des expressions telles que $? + $?
- En cas de crash, cochez le flag "crash".
- Si quelque chose ne marche pas, cochez le flag "incomplete work"

Signaux

- ctrl-C dans un prompt vide devrait afficher une nouvelle ligne avec un nouveau prompt.
- ctrl-\ dans un prompt vide ne devrait rien faire.
- ctrl-D dans un prompt vide devrait quitter minishell. Ensuite, relancez-le.
- ctrl-C dans un prompt après avoir écrit des choses devrait afficher une nouvelle ligne avec un nouveau prompt.
- Également, le buffer devrait être vide. Appuyez sur "Entrée" afin de vous assurer que la ligne précédente a été exécutée.
- ctrl-D dans un prompt après avoir écrit des choses ne devrait rien faire.
- ctrl-\ dans un prompt après avoir écrit des choses ne devrait rien faire.
- Essayez ctrl-C après avoir lancé une commande bloquante, comme cat ou grep sans argument.
- Essayez ctrl-\ après avoir lancé une commande bloquante, comme cat ou grep sans argument.
- Essayez ctrl-D après avoir lancé une commande bloquante, comme cat ou grep sans argument.
- Répétez plusieurs fois en utilisant des commandes différentes.
- En cas de crash, cochez le flag "crash".
- Si quelque chose ne marche pas, cochez le flag "incomplete work"

Double Quotes

- Exécutez une commande simple avec des arguments, mais cette fois utilisez des guillemets (rajoutez des ';' et des espaces entre les guillemets).
- Essayez une commande comme : echo "cat lol.c | cat > lol.c"
- N'essayez pas $.
- En cas de crash, cochez le flag "crash".
- Si quelque chose ne marche pas, cochez le flag "incomplete work".

Single Quotes

- Exécutez des commandes avec des single quotes dans les arguments.
- Essayez des arguments vides.
- Faites des tests avec des variables d'environnement, des espaces, des pipes, des redirections entre les guillemets.
- echo '$USER' doit afficher "$USER"
- Rien ne devrait être interprété.

env

- Vérifiez qu'env vous affiche bien les variables d'environnement.

export

- Exportez des variables d'environnement, dont certaines pour remplacer les anciennes.
- Vérifiez le résultat avec env.

unset

- Exportez des variables d'environnement, dont certaines pour remplacer les anciennes.
- Utilisez unset pour en retirer.
- Vérifiez le résultat avec env.

cd

- Utilisez la commande cd pour vous déplacer dans l'arborescence et utilisez /bin/ls pour vérifier que vous êtes dans le bon répertoire.
- Répétez ce test plusieurs fois avec des cd qui fonctionnent et qui ne fonctionnent pas.
- Essayez aussi '.' et '..' en arguments.

pwd

- Utilisez la commande pwd, avec et sans argument.
- Répétez ce test plusieurs fois dans différents répertoires.
- Essayez '.' et '..' en arguments.

Chemin relatif

- Exécutez des commandes en utilisant un chemin relatif.
- Répétez ce test plusieurs fois dans d'autres dossier avec un chemin relatif complexe (beaucoup de ..).

PATH d'environnement

- Exécutez des commandes mais sans PATH (ls, wc, awk, etc...).
- Retirez le $PATH et vérifiez si les commandes ne fonctionnent plus.
- Mettez plusieurs répertoires à PATH (directory1:directory2) et vérifiez qu'ils sont bien évalués de gauche à droite.

Redirection

- Exécutez des commandes avec les redirections < et/ou >
- Répétez ce test plusieurs fois avec différentes commandes et différents arguments et, quelques fois, utilisez >> au lieu de >.
- Vérifiez si plusieurs instances de la même redirection échouent.
- Testez les redirections avec << (cela ne doit pas forcément mettre à jour l'historique).

Pipes

- Exécutez des commandes avec des pipes telles que 'cat file | grep bla | more'
- Répétez plusieurs fois avec différentes commandes et différents arguments.
- Essayez des commandes qui échouent telles que 'ls fichiernul | grep bla | more'
- Mixez les pipes et les redirections.

Soyons fous ! Et l'historique

- Entrez une commande, puis ctrl-C, et appuyez sur "Entrée". Le buffer devrait être vide et il ne devrait plus rien avoir à exécuter.
- Peut-on naviguer dans l'historique avec Haut et Bas (profitez-en pour relancer des commandes) ?
- Exécutez des commandes qui ne fonctionnent pas telles que 'dskdskdksd' et vérifiez que tout fonctionne comme prévu.
- 'cat | cat | ls' doit fonctionner.
- Essayez des commandes vraiment, vraiment longues avec des tonnes d'arguments.
- Amusez-vous avec ce superbe minishell et profitez-en.

Variables d'environnement

- Exécutez echo avec des variables d'environnement ($variable) en argument.
- Assurez-vous que $ est interprété correctement.
- Vérifiez que les guillemets autour des $variables fonctionnent correctement (comme dans bash).
- Si USER n'existe pas, définissez-la.
- Ainsi, echo "$USER" devrait afficher la valeur de $USER.

Partie bonus

Les bonus ne seront examinés que si la partie obligatoire est excellente. Cela signifie que la partie obligatoire doit avoir été réalisée du début à la fin, avec une gestion d'erreur parfaite même en cas d'usage inattendu. Si tous les points obligatoires n'ont pas été attribués pendant cette soutenance, aucun point bonus ne sera comptabilisé.

And, Or

- Lancez des commandes avec &&, || et des parenthèses, et vérifiez que tout fonctionne comme dans bash.

Wildcard

- Utilisez des wildcards en argument dans le répertoire courant.

Surprise ! Ou pas...

- Mettez une valeur à la variable USER.
- echo "'$USER'" doit afficher la valeur de la variable USER.
- echo '"$USER"' doit afficher "$USER".