



(https://profile.intra.42.fr)

(https://s

(https://profile.intra.42.fr/searches)  
Scale for project minishell (/projects/minishell)

You should correct 1 student in this team



## Git repository

vogsphere@vogsphere.42.fr:intra/2017/activities/minishell/mprevot



## Introduction

Nous vous demandons pour le bon déroulement de cette évaluation de respecter les règles suivantes :

- Restez courtois, polis, respectueux et constructifs en toutes situations lors de cet échange. Le lien de confiance entre la communauté 42 et vous en dépend.
- Mettez en évidence auprès de la personne (ou du groupe) notée les dysfonctionnements éventuels du travail rendu, et prenez le temps d'en discuter et d'en débattre.
- Acceptez qu'il puisse y avoir parfois des différences d'interprétation sur les demandes du sujet ou l'étendue des fonctionnalités. Restez ouvert d'esprit face à la vision de l'autre (a-t-il ou elle raison ou tort ?), et notez le plus honnêtement possible. La pédagogie de 42 n'a de sens que si la peer-évaluation est faite sérieusement.

## Guidelines

- Vous ne devez évaluer que ce qui se trouve sur le dépôt GiT de rendu de l'étudiant(e) ou du groupe.
- Prenez soin de vérifier que le dépôt GiT est bien celui correspondant à l'étudiant(e) ou au groupe, et au projet.
- Vérifiez méticuleusement qu'aucun alias malicieux n'a été utilisé pour vous induire en erreur et vous faire évaluer autre chose que le contenu du dépôt officiel.
- Tout script sensé faciliter l'évaluation fourni par l'un des deux partis doit être rigoureusement vérifié par l'autre parti pour éviter des mauvaises surprises.
- Si l'étudiant(e) correcteur/correctrice n'a pas encore fait ce projet, il est obligatoire pour cet(te) étudiant(e) de lire le sujet en entier avant de commencer cette soutenance.
- Utilisez les flags disponibles sur ce barème pour signaler un rendu vide, non fonctionnel, une faute de norme, un cas de triche, etc. Dans ce cas, l'évaluation est terminée et la note finale est 0 (ou -42 dans le cas spécial de la triche). Toutefois, hors cas de triche, vous êtes encouragés à continuer d'échanger autour du

travail effectué (ou non effectué justement) pour identifier les problèmes ayant entraîné cette situation et les éviter pour le prochain rendu.

# Attachments

 Subject (<https://cdn.intra.42.fr/pdf/pdf/107/minishell.fr.pdf>)

## Sections

### Partie obligatoire

Rappel : si à un moment ou un autre, le programme ne réagit pas correctement (bus error, segfault, etc..), ou bien si vous détectez une fuite mémoire, la soutenance est terminée et la note est 0. Pensez à utiliser les flags correspondants quand cela est nécessaire. Cette consigne est active d'un bout à l'autre de la soutenance.

#### Fichier auteur

Vérifiez que le fichier "auteur" est bien présent à la racine du dépôt et formaté tel que demandé dans le sujet. Dans le cas contraire, la soutenance est terminée et la note est 0.

 Yes

 No

#### Fuites mémoire

Pendant toute la durée de la soutenance, gardez un oeil sur la quantité de mémoire utilisée par le minishell (à l'aide de top par exemple). Cette quantité doit rester à peu près fixe commande après commande. Dans le cas contraire, il y a au moins une fuite mémoire, la note du projet est 0.

 Yes

 No

#### Fork et execve

"fork" et "execve" sont au coeur de la base d'un shell minimaliste, tel que le minishell. Si vous constatez que ces deux fonctions ne sont jamais appelées dans le code source du programme, c'est qu'il y a un problème de compréhension du sujet. La soutenance est terminée et la note est 0. En respectant la liste des fonctions autorisées, il n'y a pas d'autre solution.

Effectuez les 4 tests suivants. Si l'un au moins de ces tests échoue, alors toute la section est échouée, passez à la suivante :

- Lancez le minishell, puis lancez la commande "\$> foo". La commande doit échouer avec un message d'erreur adapté et rendre le prompt.

- Lancez la commande "\$> /bin/ls". Le binaire ls doit s'exécuter correctement, puis le prompt doit réapparaître.

- Lancez la commande "\$> /bin/ls -laF". Le binaire doit s'exécuter correctement avec les options -l, -a et -F, puis le prompt doit réapparaître,

- Lancez la commande "\$> /bin/ls -l -a -F". Le binaire doit s'exécuter correctement avec les options -l, -a et -F, puis le prompt doit réapparaître

✓ Yes

✗ No

## Les builtins

Nous allons évaluer dans cette section l'implémentation des builtins "exit", "echo" et "cd". Même préhistorique, un shell doit proposer certaines fonctionnalités.

Effectuez les 6 tests suivants. Si l'un au moins de ces tests échoue, alors toute la section est échouée, passez à la suivante :

- Lancez le minishell, puis lancez la commande "\$> exit". Le programme doit se terminer proprement et rendre la main au shell parent. Relancez le minishell.

- Lancez une commande telle que "\$> echo It works". Le message doit s'afficher correctement.

- Lancez une commande telle que "\$> cd /absolute/path/of/your/choice", puis lancez la commande "\$> /bin/pwd". La commande /bin/pwd doit confirmer que le dossier courant a bien été changé.

- Lancez une commande telle que "\$> cd relative/path/of/your/choice", puis lancez la commande "\$> /bin/pwd". La commande /bin/pwd doit confirmer que le dossier courant a bien été changé.

- Lancez la commande "\$> cd", puis lancez la commande "\$> /bin/pwd". La commande /bin/pwd doit confirmer que le dossier courant est bien le home de l'utilisateur.

- Lancez la commande "\$> cd -", puis lancez la commande "\$> /bin/pwd". La commande /bin/pwd doit confirmer que le dossier courant est bien le dossier relative/path/of/your/choice précédent.

✓ Yes

✗ No

## Gestion de l'environnement

Nous allons évaluer dans cette section l'implémentation des builtins "env", "setenv" et "unsetenv".

Effectuez les 6 tests suivants. Si l'un au moins de ces tests échoue, alors toute la section est échouée, passez à la suivante :

- Lancez la commande "\$> env". Les variables d'environnement doivent s'afficher sous la forme clef=valeur.

- Lancez une commande telle que "\$> setenv FOO bar" ou "\$> setenv FOO=bar" selon la syntaxe choisie pour setenv. Lancez ensuite la commande "\$> env". L'environnement doit afficher une variable FOO avec pour valeur bar.

- Lancez la commande "\$> /usr/bin/env". Le minishell doit

transmettre le bon environnement aux binaires qu'il lance. Le binaire `/usr/bin/env` doit donc afficher l'environnement incluant une variable `FOO` avec pour valeur `bar` en s'exécutant.

- Lancez la commande `"$> unsetenv FOO"`. Lancez ensuite la commande `"$> env"`. L'environnement ne doit plus afficher une variable `FOO` avec pour valeur `bar`.

- Lancez à nouveau la commande `"$> unsetenv FOO"`. Lancez ensuite à nouveau la commande `"$> env"`. L'environnement ne doit pas avoir été modifié.

- Lancez à nouveau la commande `"$> /usr/bin/env"`. Le binaire `/usr/bin/env` doit ne doit plus afficher l'environnement incluant une variable `FOO` avec pour valeur `bar` en s'exécutant.

✓ Yes

✗ No

## Gestion du PATH

Nous allons évaluer dans cette section la gestion de la variable d'environnement `PATH` par votre shell.

Effectuez les 6 tests suivants. Si l'un au moins de ces tests échoue, alors toute la section est échouée, passez à la suivante :

- Lancez la commande `"$> unsetenv PATH"`, puis la commande `"$> setenv PATH "/bin:/usr/bin""` ou `"$> setenv "PATH=/bin:/usr/bin""` selon la syntaxe choisie. Lancez ensuite la commande `"$> ls"`. Le binaire `/bin/ls` doit s'exécuter correctement.

- Lancez la commande `"$> emacs"`. Le binaire `/usr/bin/emacs` doit s'exécuter correctement.

- Lancez la commande `"$> unsetenv PATH"`, puis lancez la commande `"$> ls"`. L'exécution doit échouer.

- Lancez à présent la commande `"$> emacs"`. L'exécution doit échouer également.

- Lancez la commande `"$> /bin/ls"`. Le binaire `/bin/ls` doit s'exécuter correctement.

- Lancez la commande `"$> /usr/bin/emacs"`. Le binaire `/usr/bin/emacs` doit s'exécuter correctement.

✓ Yes

✗ No

## Gestion de la ligne de commande

Nous allons évaluer dans cette section la gestion de la ligne de commande. Effectuez les 4 tests suivants. Si l'un au moins de ces tests échoue, alors toute la section est échouée, passez à la suivante :

- Lancez une commande vide `"$> "`. Le shell ne doit rien faire et réafficher le prompt.

- Lancez une commande composée uniquement d'un seul espace `"$> "`. Le shell ne doit rien faire et réafficher le prompt.

- Lancez une commande composée uniquement d'espaces et de tabulations. Le shell ne doit rien faire et réafficher le prompt.

- Lancez une commande avec plusieurs espaces et tabulations avant le nom du binaire, entre chaque argument passé au binaire, et après le dernier argument. Ces espaces et tabulations inutiles ne doivent pas perturber l'exécution de la commande.

✓ Yes

✗ No

## Bonus

Rappel : si à un moment ou un autre, le programme ne réagit pas correctement (bus error, segfault, etc.), la soutenance est terminée et la note est 0. Pensez à utiliser les flags correspondants. Cette consigne est active d'un bout à l'autre de la soutenance. Les bonus ne doivent être évalués que si et seulement si la partie obligatoire est PARFAITE. Par PARFAITE, on entend bien évidemment qu'elle est entièrement réalisée, qu'il n'est pas possible de mettre son comportement en défaut, même en cas d'erreur, aussi vicieuse soit-elle, de mauvaise utilisation, etc. Concrètement, cela signifie que si la partie obligatoire n'a pas obtenu TOUS les points pendant cette soutenance, les bonus doivent être intégralement IGNORÉS.

### Signal

Nous allons évaluer dans cette section la gestion des signaux, et en particulier de Ctrl-C.

Effectuez les 3 tests suivants. Si l'un au moins de ces tests échoue, alors toute la section est échouée, passez à la suivante :

- Au lieu de taper une commande à l'invite du minishell, appuyez sur Ctrl-C. Le minishell doit réafficher le prompt.

- Entrez une commande quelconque. Au lieu de la valider avec la touche entrée, appuyez sur Ctrl-C. Le minishell doit réafficher le prompt sans réafficher la commande incomplète.

- Lancez la commande "\$> cat", puis pendant que cat attend de lire sur l'entrée standard, appuyez sur Ctrl-C. Le minishell doit tuer le processus de cat, puis réafficher le prompt.

✓ Yes

✗ No

### Droits dans le PATH

Nous allons évaluer dans cette section la gestion des droits dans le PATH. Effectuez le test suivant. Si ce test échoue, alors toute la section est échouée, passez à la suivante :

- Créez un répertoire /tmp/bin/ et ajoutez ce répertoire à la variable PATH de l'environnement. Créez un programme nommé 'test\_exec\_rights' dans ce répertoire dans le langage de votre choix qui affiche 'KO' lorsqu'on l'exécute. Donnez les droits 644 (donc pas de droits d'exécution) à ce programme. Depuis un autre répertoire que /tmp/bin, lancez la commande "\$> test\_exec\_rights". Vérifiez bien que le minishell refuse de lancer le programme à cause des droits.

Si ce test échoue, cette section du barème est comptée fautive et aucun point n'est gagné, passez à la section suivante.

✓ Yes

✗ No

## Completion

Nous allons évaluer dans cette section la complétion.  
Effectuez les 2 tests suivants. Si l'un au moins de ces tests échoue,  
alors toute la section est échouée, passez à la suivante :

- Tapez le début de commande "\$> ec", puis appuyez sur  
tabulation. Le minishell doit compléter la commande en  
"\$> echo".

- Tapez le début de commande "\$> ema", puis appuyez sur  
tabulation. Le minishell doit compléter la commande en  
"\$> emacs".

✓ Yes

✗ No

## point virgule

Nous allons évaluer dans cette section la gestion de plusieurs  
commandes d'affilé séparées par des points vigules.  
Effectuez les 2 tests suivants. Si l'un au moins de ces tests échoue,  
alors toute la section est échouée, passez à la suivante :

- Lancez la commande "\$> echo TOP ; ls ; echo MIDDLE ; ls ; echo BOTTOM".  
Les 5 commandes doivent s'exécuter normalement dans l'ordre ou  
elles ont été écrites.

- Lancez la commande "\$> ;". Le minishell doit soit ne rien  
faire du tout et réafficher le prompt, soit afficher une  
erreur de syntaxe et réafficher le prompt.

✓ Yes

✗ No

## Autres features

Si le rendu propose d'autres bonus qui vous semblent pertinents,  
comptabilisez les dans cette section.



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

## Ratings

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

✓ Ok

★ Outstanding project

■ Empty work

■ Incomplete work

💬 No author file

⚙ Invalid compilation

📄 Norme

📖 Cheat

💥 Crash

## Conclusion

Leave a comment on this evaluation

**\* (required) Comment**

Finish evaluation