

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

# SCALE FOR PROJECT 21 SH (/PROJECTS/21 SH)

You should evaluate 2 students in this team



Git repository

vogsphere@vogsphere.13 

## 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 special de la triche). Toutefois, hors cas de triche, vous êtes encouragés à continuer décharger 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/967/21sh.en.pdf>)

 Subject (<https://cdn.intra.42.fr/pdf/pdf/468/21sh.fr.pdf>)

## Partie obligatoire

*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.*

---

### 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 œil sur la quantité de mémoire utilisée par le 21sh (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 21sh. 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, la soutenance est terminée et la note est 0.

- Lancez le 21sh, 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, Dans le cas contraire, la soutenance est terminée et la note est 0.

- Lancez le 21sh, puis lancez la commande "\$> exit". Le programme doit se terminer proprement et rendre la main au shell parent. Relancez le 21sh.

- Lancer une commande telle que "\$> echo "It works"". Le message doit s'afficher correctement.
- Lancez une commande telle que "\$> echo It works". (sans les doubles quotes) 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.
- Lancez la commande "\$> cd ~/path/of/your/choice", puis lancez "\$> /bin/pwd". "\$> /bin/pwd". La commande /bin/pwd doit confirmer que le dossier courant a bien été changé.

 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, Dans le cas contraire, la soutenance est terminée et la note est 0.

- 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 21sh 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, Dans le cas contraire, la soutenance est terminée et la note est 0.

- 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 6 tests suivants. Si l'un au moins de ces tests échoue, Dans le cas contraire, la soutenance est terminée et la note est 0.

- 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

---

## 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, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Au lieu de taper une commande à l'invite du 21 sh, appuyez sur ctrl-c. Le 21 sh doit réafficher le prompt.
- Tapez une commande quelconque et au lieu de la valider avec la touche entrée, appuyez sur ctrl-c. Le 21 sh 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 21 sh doit tuer le processus de cat, puis réafficher le prompt.



Yes



No

## Pipes

Nous allons évaluer dans cette section la gestion des pipes. Effectuez les 3 tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez la commande "\$> ls | cat -e". Le résultat de cette commande doit être le contenu du dossier dont chaque ligne se termine par le caractère '\$'.

- Lancez la commande "\$> ls -r | sort | cat -e". Le résultat de cette commande doit être le contenu du dossier, trié, dont chaque ligne se termine par le caractère '\$'.

- Lancez la commande "\$> base64 /dev/urandom | head -c 1000 | grep 42 | wc -l | sed -e 's/1/Yes/g' -e 's/0/No/g'". La commande doit afficher "Yes" si la chaîne "42" a été détectée dans les caractères randoms, ou "No" sinon.

✓ Yes

✗ No

## Redirections

Nous allons évaluer dans cette section la gestion des redirections. Effectuez les 5 tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez la commande "\$> echo "Testing redirections," > /tmp/test.txt" et vérifiez que le fichier /tmp/test.txt contient bien la chaîne "Testing redirections".

- Lancez la commande "\$> echo "with multiple lines" >> /tmp/test.txt" et vérifiez que le fichier /tmp/test.txt contient bien les chaînes "Testing redirections," et "with multiple lines" sur deux lignes.

- Lancez la commande "\$> wc -c < /tmp/test.txt" et vérifiez que la valeur affichée est bien 42.

- Lancez la commande "\$> cat -e << EOF", puis entrez le texte suivant sans les triples guillemets mais en respectant les sauts de lignes :

""""

Roses are red

Violets are blue  
All my base are belong to you  
And so are you  
""

Appuyez ensuite sur ctrl+d pour terminer votre entrée.

La sortie de la commande doit être exactement :

Roses are red\$  
Violets are blue\$  
All my base are belong to you\$  
And so are you\$

- Lancez la commande "\$> cat -e << EOF >> /tmp/test.txt" puis entrez à nouveau le poème du test précédent. Vérifiez que le fichier /tmp/test.txt contient bien 6 lignes :

Testing redirections, \$  
with multiple lines\$  
Roses are red\$  
Violets are blue\$  
All my base are belong to you\$  
And so are you\$

☒ Yes

☐ No

---

## Plusieurs commandes de suite

Nous allons évaluer dans cette section la gestion de plusieurs commandes d'affilé avec le séparateur ';'. Effectuez le test suivant. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez la commande "\$> ls -l ; touch newfile; ls -l". Les deux ls doivent être exécutés et la seule différence entre les deux doit être le fichier "newfile".

☒ Yes

☐ No

---

## Un peu de tout

Nous allons évaluer dans cette section l'utilisation de pipes, de redirections et de ';' tous ensembles. Effectuez les 2 tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez la commande "\$> mkdir test ; cd test ; ls -a ; ls | cat | wc -c > fifi ; cat fifi"



La sortie doit être :

. ..

5

- Lancez la commande "\$> cd /tmp; sort << EOF | cat -e > sorted\_poem ; sed -e 's/Roses/Turnips/' < sorted\_poem > better\_poem; cd -; echo "I prefer turnips anyway" >> /tmp/better\_poem; cat /tmp/better\_poem" et entrez le poème suivant à l'invite du heredoc (sans les triples guillemets) :

""""

Roses are red

Violets are blue

All my base are belong to you

I love you

""""

La sortie doit être (toujours sans les triples guillemets) :

""""

All my bases are belong to you\$

I love you\$

Turnips are red\$

Violets are blue\$

I prefer turnips anyway

""""

☒ Yes

☐ No

---

## Aggrégation des descripteurs de fichiers

Nous allons évaluer dans cette section l'aggrégation des descripteurs de fichiers. Effectuez les 3 tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Lancez la commande "\$> rm nosuchfile 2>&- " en vous assurant au préalable qu'il n'existe pas de fichier nommé 'nosuchfile' dans le répertoire courant. Le message d'erreur attendu ne doit pas s'afficher.

- Lancez la commande "\$> rm nosuchfile 2>&1 | cat -e " en vous assurant au préalable qu'il n'existe pas de fichier nommé 'nosuchfile' dans le répertoire courant. La sortie doit être :  
"rm: nosuchfile: No such file or directory\$"

- Lancez la commande "echo "No dollar character" 1>&2 | cat -e". La sortie doit être "No dollar character".

☒ Yes

☐ No

---

## Edition de ligne simple

Nous allons évaluer dans cette section l'édition de ligne simple. Effectuez les 3 tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Il doit être possible de déplacer le curseur vers la gauche et vers la droite dans la ligne courante avec les touches flèche gauche et flèche droite.
- Il doit être possible d'éditer la ligne courante à l'endroit où se trouve le curseur.
- Il doit être possible de sauter vers le début et la fin de ligne avec les touches home et end.
- Il doit être possible de naviguer dans l'historique avec les touches flèches haut et bas.

✓ Yes

✗ No

---

### Edition de ligne avancée

Nous allons évaluer dans cette section l'édition de ligne avancée. Effectuez les 3 tests suivants. L'évaluation est incrémentale, échouer un de ces tests n'empêche pas de réussir les autres.

- Il doit être possible de se déplacer d'un mot vers la gauche ou vers la droite dans la ligne courante avec la touche ctrl + la touche flèche gauche ou droite. (1 point)
- Il doit être possible de Copier/coller de tout ou partie de la ligne avec un raccourci clavier. (2 points)
- Il doit être possible d'écrire ou éditer une commande sur plusieurs lignes en même temps. (2 points)

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



---

**ctrl+D et ctrl+C**

Nous allons évaluer dans cette section la gestion de ctrl+D et de ctrl+C. Effectuez les 3 tests suivants. Si l'un au moins de ces tests échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Faites un ctrl+D lorsque la ligne de commande est vide. Le shell doit quitter proprement.

- Faites un ctrl+D lorsque la ligne de commande n'est pas vide. Il ne doit rien se passer.

- Lancez la commande "\$> cat", tapez quelques caractères puis appuyez sur ctrl+D deux fois. Le premier appui doit afficher les caractères tapés, le second doit rendre le prompt.

- Appuyez sur ctrl+C quand la ligne de commande est vide et quand la ligne de commande contient des caractères. Dans les deux cas, le shell doit rendre le prompt.

- Lancez la commande "\$> cat" puis appuyez sur ctrl+C. Le shell doit tuer le processus de cat et rendre le prompt.

✓ Yes

✗ No

---

## Gestion des quotes

Nous allons évaluer dans cette section la bonne gestion des commandes quotées. Effectuez le test suivant. Si ce test échoue, cette section du barème est échouée et aucun point n'est gagné, passez à la suivante.

- Entrez la commande '\$> echo "', puis appuyez sur entrée. Le shell doit revenir à la ligne et attendre la fin de la commande. Tapez quelques lignes supplémentaires, puis fermez les guillemets et appuyez sur entrée. Le shell doit récupérer la commande et la traiter correctement. Par exemple :

""""

\$> echo "

\*> Roses are red

\*> Violets are blue

\*> All my base are belong to you

\*> I love you

\*> "

Roses are red

Violets are blue  
 All my base are belong to you  
 I love you

\$>

|||||

✓ Yes

✗ No

## Bonus

*Rappel : si a 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.*

### Autres features

D'autres bonus ? Vous pouvez en comptabiliser jusqu'à cinq.



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

👤 Incomplete group

💧 Leaks

🚫 Forbidden function

## Conclusion

Leave a comment on this evaluation

**Finish evaluation**

General term of use of the site (<https://signin.intra.42.fr/legal/terms/6>)  
Privacy policy (<https://signin.intra.42.fr/legal/terms/5>)  
Legal notices (<https://signin.intra.42.fr/legal/terms/3>)  
Declaration on the use of cookies (<https://signin.intra.42.fr/legal/terms/2>)  
Terms of use for video surveillance (<https://signin.intra.42.fr/legal/terms/1>)  
Rules of procedure (<https://signin.intra.42.fr/legal/terms/4>)