

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

## SCALE FOR PROJECT MINISHELL (/PROJECTS/42CURSUS-MINISHELL)

You should evaluate 2 students in this team



Git repository

`git@vogsphere-v2.42sp.org.br:vogsphere/intra-uuid-0e5ca0c1-9c`

### Introduction

Por favor, siga as seguintes regras:

- Mantenha-se educado, cortês, respeitoso e construtivo durante todo o processo de avaliação. O bem-estar da comunidade depende disso.
- Identifique as possíveis disfunções no projeto do aluno ou grupo cujo trabalho está sendo avaliado. Reserve um tempo para discutir e debater os problemas que podem ter sido identificados.
- Você deve considerar que pode haver algumas diferenças em como seus colegas podem ter compreendido as instruções do projeto e o escopo de suas funcionalidades. Mantenha sempre a mente aberta e os avalie da forma mais honesta possível. A pedagogia só é útil se a avaliação por pares for feita seriamente.

### Guidelines

- Avalie apenas o trabalho submetido no repositório Git do aluno ou grupo avaliado.
- Verifique novamente se o repositório Git pertence ao(s) aluno(s). Certifique-se de que o projeto é o esperado. Além disso, verifique se o 'git clone' foi usado em um diretório vazio.
- Verifique cuidadosamente se nenhum alias malicioso foi usado para enganá-lo e fazê-lo avaliar algo que não seja o conteúdo do repositório oficial.
- Para evitar surpresas, e se aplicável, revise em conjunto quaisquer scripts usados para facilitar a classificação, como scripts de teste ou automação.
- Se você não concluiu a tarefa que vai avaliar, você deve ler todo o assunto antes de iniciar o processo de avaliação.
- Utilize as flags disponíveis para relatar um repositório vazio, um programa não funcional, um erro de Norma, trapaça, etc.  
Nestes casos, o processo de avaliação termina e a nota final é 0, ou -42 em caso de trapaça. No entanto, exceto em casos de trapaça, os alunos são fortemente encorajados a rever o trabalho submetido juntos para identificar quaisquer erros que não devem ser repetidos no futuro.
- Você também deve verificar se não há vazamentos de memória. Qualquer memória alocada no heap deve ser liberada corretamente antes do término da execução do programa. Você pode usar qualquer uma das ferramentas disponíveis no computador, como leaks, valgrind ou e\_fence. Em caso de vazamentos de memória, marque a flag apropriada.

### Attachments

 [subject.pdf \(https://cdn.intra.42.fr/pdf/pdf/198772/pt\\_br.subject.pdf\)](https://cdn.intra.42.fr/pdf/pdf/198772/pt_br.subject.pdf)

### Parte Obrigatória

README e Verificação de Conformidade

O repositório contém um arquivo README.md em sua raiz, e o arquivo inclui todo o seguinte?

- A primeira linha está em itálico e formatada exatamente como: *Este projeto foi criado com da 42 por <login1>[, <login2>[, <login3>[...]]]*.
- Uma seção "Descrição" explicando o propósito do projeto e fornecendo uma breve visão
- Uma seção "Instruções" com detalhes relevantes sobre compilação, instalação e/ou exec
- Uma seção "Recursos" listando referências (documentação, tutoriais, etc.) e explicando c especificando para quais tarefas e quais partes do projeto.

Se quaisquer dos elementos requeridos acima estiver faltando, a nota é 0.

✓ Yes

✗ No

Compilação

- Use "make -n" para verificar se a compilação usa "-Wall -Wextra -Werror". Se não, use a i compilation".
- minishell deve compilar sem erros. Se não, selecione a flag "invalid compilation".
- O Makefile não deve dar relink. Se não, selecione a flag.

✓ Yes

✗ No

Comando simples e variáveis globais

- Rode um comando simples usando um caminho absoluto, como /bin/ls, ou qualquer outrc opções.
- Quantas variáveis globais são usadas? Por que? Peça ao estudante avaliado para dar ur da razão pela qual parece obrigatório ou lógico esse uso.
- Verifique a variável global. Esta variável global não deve fornecer nenhuma outra informa dados além do número de um sinal recebido.
- Teste um comando vazio.
- Teste comandos que contenham somente espaços ou tabs.
- Se algo quebrar, selecione a flag "crash".
- Se algo não funcionar, selecione a flag "incomplete work".

✓ Yes

✗ No

Argumentos

- Execute um comando simples com um caminho absoluto como /bin/ls, ou qualquer outro argumentos mas sem quaisquer aspas ou aspas duplas.
- Repita o teste múltiplas vezes com diferentes comandos e argumentos.
- Se algo quebrar, selecione a flag "crash".
- Se algo não funcionar, selecione a flag "incomplete work".

✓ Yes

✗ No

echo

- Execute o comando echo sem argumentos ou a opção -n.
- Repita múltiplas vezes com diferentes argumentos.
- Se algo quebrar, selecione a flag "crash".
- Se algo não funcionar, selecione a flag "incomplete work".

✓ Yes

✗ No

exit

- Execute o comando exit com ou sem argumentos.
- Repita múltiplas vezes com diferentes argumentos.
- Não se esqueça de reiniciar o minishell
- Se algo quebrar, selecione a flag "crash".
- Se algo não funcionar, selecione a flag "incomplete work"..

✓ Yes

✗ No

Valor de retorno de um processo

Intra Projects minishell Edit

- Execute um comando simples com um caminho absoluto como /bin/ls, ou qualquer outro argumentos mas sem quaisquer aspas ou aspas duplas. Então execute echo \$?
- Verifique o valor impresso. Você pode fazer o mesmo no bash para comparar os resultados
- Repita múltiplas vezes com diferentes comandos e argumentos. Tente alguns comandos '/bin/ls arquivoquenaonaoexiste'.
- Tente expressões como expr \$? + \$?
- Se algo quebrar, selecione a flag "crash".
- Se algo não funcionar, selecione a flag "incomplete work".

✓ Yes

✗ No

Sinais

- ctrl-C num prompt vazio deve mostrar uma nova linha com um novo prompt.
- ctrl-\ num prompt vazio não deve fazer nada.
- ctrl-D num prompt vazio deve sair do minishell --> REINICIE!
- ctrl-C num prompt em que você escreveu algo deve mostrar uma nova linha com um novo prompt
- O buffer deve estar limpo também. Pressione "Enter" para se certificar de que nada da linha foi executado.
- ctrl-D num prompt em que você escreveu algo não deve fazer nada.
- ctrl-\ num prompt em que você escreveu algo não deve fazer nada.
- Tente dar ctrl-C depois de rodar um comando de bloqueio como cat sem argumentos ou cat > "qualquercoisa".
- Tente dar ctrl-\ depois de rodar um comando de bloqueio como cat sem argumentos ou cat > "qualquercoisa".
- Tente dar ctrl-D depois de rodar um comando de bloqueio como cat sem argumentos ou cat > "qualquercoisa".
- Repita múltiplas vezes usando diferentes comandos.
- Se algo quebrar, selecione a flag "crash".
- Se algo não funcionar, selecione a flag "incomplete work".

✓ Yes

✗ No

Aspas Duplas

- Execute um comando simples com argumentos e, desta vez, use também aspas duplas (para incluir espaços em branco também).
- Tente um comando como : echo "cat lol.c | cat > lol.c"
- Tente qualquer coisa exceto \$.
- Se algo quebrar, selecione a flag "crash".
- Se algo não funciona, selecione a flag "incomplete work".

✓ Yes

✗ No

Aspas simples

- Execute comandos com aspas simples como argumentos.
- Tente comandos vazios.
- Tente usar variáveis de ambiente, espaços em branco, pipes, redirecionamentos dentro de comandos
- echo '\$USER' deve imprimir "\$USER".
- Nada deve ser interpretado.

✓ Yes

✗ No

env

- Verifique se env mostra as variáveis de ambiente atuais.

✓ Yes

✗ No

export

- Exporte variáveis de ambiente, crie novas e substitua as antigas.
- Verifique os resultados com env.

✓ Yes

✗ No

unset

- Exporte variáveis de ambiente, crie novas e substitua as antigas.
- Use unset para remover algumas delas.

- Verifique os resultados com env.

☒ Yes☐ No

#### cd

- Use o comando cd para mover o diretório de trabalho e verifique se você está no diretório
- Repita múltiplas vezes com comandos cd que funcionam e que não funcionam
- Tente também '.' e '..' como argumentos.

☒ Yes☐ No

#### pwd

- Use o comando pwd.
- Repita múltiplas vezes em diferentes diretórios.

☒ Yes☐ No

#### Caminho relativo

- Execute comandos mas desta vez usando um caminho relativo.
- Repita múltiplas vezes em diferentes diretórios com um caminho relativo complexo (vários níveis de diretórios).

☒ Yes☐ No

#### Environment path

- Execute comandos mas desta vez sem usar um caminho (ls, wc, awk e por aí vai).
- Dê um unset no \$PATH assegure-se de que os comandos não estejam mais funcionando
- Dê um set no \$PATH para um valor de múltiplos diretórios (diretório1:diretório2) e assegure-se de que os comandos são verificados em ordem da esquerda para a direita.

☒ Yes☐ No

#### Redirecionamento

- Execute comandos com redirecionamentos < e/ou >
- Repita múltiplas vezes com diferentes comandos e argumentos e de vez em quando mude o redirecionamento.
- Verifique se múltiplas tentativas do mesmo redirecionamento falham.
- Teste redirecionamentos com << (ele não precisa atualizar o histórico).

☒ Yes☐ No

#### Pipes

- Execute comandos com pipes com 'cat arquivo | grep bla | mais'
- Repita múltiplas vezes com diferentes comandos e argumentos
- Tente alguns comandos errados como 'ls arquivoquenaonaoexiste | grep bla | mais'
- Tente misturar pipes e redirecionamentos.

☒ Yes☐ No

#### Seja louco e histórico

- Digite uma linha de comando, então use ctrl-C e pressione "Enter". O buffer deve estar livre para ser executado.
- Conseguimos navegar o histórico usando as setas para cima e para baixo? Conseguimos executar um comando?
- Execute comandos que não funcionam como 'dsbksdgbksdghsd'. Assegure-se de que o erro seja exibido e imprima uma mensagem de erro.
- 'cat | cat | ls' deve se comportar de uma "maneira normal".
- Tente executar um comando longo com um monte de argumentos.
- Divirta-se com esse minishell lindo e aproveite!

☒ Yes☐ No

#### Variáveis de ambiente

Intra Projects minishell Edit

- Execute echo com algumas variáveis de ambiente (\$variável) como argumentos.
  - Verifique se \$ é interpretado como uma variável de ambiente.
  - Verifique se aspas duplas interpolam \$.
  - Verifique se USER existe. Caso não exista, configure-a.
  - echo "\$USER" deve imprimir o valor da variável USER.
- ☒ Yes
- ☐ No

Bonus

Avalie a parte bônus se, e somente se, a parte obrigatória estiver completa e perfeitamente feita, de erros lida com usos inesperados ou mau uso. No caso de todos os pontos obrigatórios não p defesa, os pontos de bônus devem ser totalmente ignorados.

E, Ou

- Use &&, ||, e parêntesis com comandos para verificar se o minishell se comporta da mesr bash.
- ☒ Yes
- ☐ No

Coringa

- Use coringas nos argumentos no diretório atual de trabalho.
- ☒ Yes
- ☐ No

Surpresa! (ou talvez não...)

- Defina a variável de ambiente USER.
  - echo "\$USER" deve imprimir o valor da variável USER.
  - echo "\$USER" deve imprimir "\$USER".
- ☒ Yes
- ☐ No

Ratings

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

☒ Ok

☐ Outstanding project

Empty work

☐ Incomplete work

☐ Invalid compilation

☐ Norme

☐ Cheat

☐ Incomplete group

☐ Concerning situation

☐ Leaks

☐ Fort

☐ Can't support / explain code

Conclusion

Leave a comment on this evaluation ( 2048 chars max )

Finish evaluation