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

nquecedo

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

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

You should evaluate 2 students in this team



Git repository

git@vogsphere.42urduliz.com:vogsphere/intra-uuid-3825ebfd-a6e6-4a52-



## Introduction

Por favor respeta las siguientes normas:

- Sé educado, cordial, respetuoso y constructivo durante el proceso de evaluación. El bienestar de la comunidad depende de ello.
- Identifica con la persona o grupo evaluado, las posibles disfuncionalidades del trabajo. Tómate el tiempo de discutir y debatir los problemas que puedes haber identificado.
- Debes considerar la posibilidad de diferir en el entendimiento de las instrucciones del proyecto y el alcance de la funcionalidad. Mantén siempre una mente abierta y evalúa de forma sincera. La pedagogía sólo es válida si las evaluaciones se toman con seriedad.

## Guidelines

- Evalúa sólo el trabajo entregado en el repositorio Git del estudiante o grupo.
- Comprueba dos veces que el repositorio Git pertenece al estudiante o grupo evaluado. Asegúrate de que el trabajo entregado es el solicitado por el proyecto y que "git clone" es utilizado en una carpeta vacía.
- Comprueba cuidadosamente que no existan alias maliciosos usados para engañarte y hacerte evaluar trabajo subido a un repositorio no oficial.
- Para evitar sorpresas, comprueba cuidadosamente que tanto el estudiante evaluado como el evaluador han revisado los posibles scripts utilizados durante la entrega para facilitar la evaluación.
- Si como evaluador no has completado este proyecto todavía, es obligatorio leer el enunciado entero antes de empezar la evaluación.
- Utiliza los flags disponibles en esta evaluación para señalar un repositorio vacío, un programa disfuncional, un fallo de norma, trampas, etc. En estos casos, la evaluación termina y la nota final es 0 (o -42 en caso de trampa). Sin embargo, a excepción de trampa, se recomienda seguir discutiendo el trabajo (incluso si no está terminado) para identificar posibles fallos y evitar repetirlos en el futuro.
- Recuerda que durante la defensa, ningún segfault, ni otros comportamientos prematuros, descontrolados o cierres del programa se toleran. En caso contrario, la nota final es 0. Utiliza el flag apropiado.

Nunca debes editar ningún archivo salvo el de configuración si existe. Si quieres editar un archivo, tómate el tiempo de explicar las razones al estudiante evaluado y asegúrate de que ambos estáis de acuerdo.

 Debes también verificar la ausencia de leaks. Toda la memoria asignada en el heap debe liberarse previamente al final de la ejecución.
 Tienes permitido usar distintas herramientas disponibles en el ordenador, tales como leaks, valgrind o e\_fence. En caso de leaks, utiliza el flag apropiado.

## **Attachments**

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/135595/es.subject.pdf)

# Parte obligatoria

#### Compila

- Utiliza make -n para verificar que la compilación utilice -Wall -Werror -Wextra, si no utiliza el flag de compilación inválida.
- minishell compila sin errores, si no, utiliza el flag apropiado.
- El Makefile no debe hacer relink. Si no, utiliza el flag apropiado.

 $igoremsize{arphi}$  Yes igotimes No

#### Comandos simples y la variable global

- Ejecuta un comando sencillo con una ruta absoluta de la forma /bin/ls, o cualquier otro comando sin argumentos.
- ¿Cuántas variables globales utiliza? ¿Por qué? Debe dar un ejemplo concreto de por qué le parece necesario o lógico.
- Compruebe la variable global. Esta variable global no puede proporcionar otra información o acceso a datos que el número de una señal recibida.
- Prueba un comando vacío.
- Prueba solo tabuladores o espacios.
- Si el programa termina, utiliza el flag de crash.
- Si algo no funciona, utiliza el flag de trabajo incompleto.

### **Argumentos**

- Ejecuta un comando simple con una ruta absoluta como /bin/ls, u otro comando con argumentos pero sin comillas simples y comillas dobles.
- Hazlo varias veces con distintos comandos y argumentos.
- Si el programa termina, utiliza el flag de crash.
- Si algo no funciona, utiliza el flag de trabajo incompleto.

⊗ Yes ×No

#### echo

- Ejecuta el comando echo con o sin argumentos, y con o sin -n.
- Hazlo múltiples veces con distintos argumentos.
- Si el programa termina, utiliza el flag de crash.
- Si algo no funciona, utiliza el flag de trabajo incompleto.

 ${\it ext{ iny Yes}}$ 

#### exit

- Ejecuta el comando exit con o sin argumentos.
- · Hazlo varias veces con distintos argumentos.
- No olvides lanzar de nuevo minishell.
- Si el programa termina, utiliza el flag de crash.
- Si algo no funciona, utiliza el flag de trabajo incompleto.

 ${f ec{arphi}}$  Yes

#### Valor de retorno de un proceso

- Ejecuta una prueba simple con una ruta absoluta del tipo /bin/ls, o algún otro comando con argumentos pero sin comillas simples o comillas dobles. Después ejecuta echo \$?
- Comprueba el valor devuelto. Puedes hacer lo mismo en bash y comparar ambos resultados.
- Hazlo varias veces, con distintos comandos y argumentos. Ejecuta algunos comandos que fallen como '/bin/ Is archivo\_que\_no\_existe'.
- Algo como 'expr \$? + \$?'.
- Si el programa termina, utiliza el flag de crash.
- Si algo no funciona, utiliza el flag de trabajo incompleto.

#### Señales

- Verifica que ctrl-C en una entrada limpia muestra una nueva línea con una entrada limpia.
- Verifica que ctrl-\ en una entrada limpia no hace nada.
- Verifica que ctrl-D en una entrada limpia termina minishell, ejecútalo de nuevo.
- Verifica que ctrl-C en una entrada con texto, muestra una nueva línea con una entrada limpia.
- El buffer debería limpiarse correctamente, pulsa "enter" para verificar que nada de lo que habías escrito se ejecute.
- Verifica que ctrl-D en una entrada con texto no hace nada.
- Verifica que ctrl-\ en una entrada con texto no hace nada.
- Verifica que ctrl-C después de ejecutar un comando bloqueante como cat sin argumentos o grep "algo".
- Verifica que ctrl-\ después de ejecutar un comando bloqueante como cat sin argumentos o grep "algo".
- Verifica que ctrl-D después de ejecutar un comando bloqueante como cat sin argumentos o grep "algo".
- · Hazlo varias veces con distintos comandos.
- Si el programa termina, utiliza el flag de crash.
- Si algo no funciona, utiliza el flag de trabajo incompleto.

⊗ Yes × No

### Comillas dobles

- Ejecuta un comando simple con argumentos, esta vez con comillas dobles (deberías probar a incluir espacios en blanco también).
- Un comando como echo "cat lol.c | cat > lol.c"
- Cualquier cosa, salvo \$.
- Si el programa termina, utiliza el flag de crash.
- Si algo no funciona, utiliza el flag de trabajo incompleto.

⊗ Yes × No

#### **Comillas simples**

- Ejecuta comandos con comillas simples como argumento.
- Prueba argumentos vacíos.
- Prueba variables de entorno, espacios en blanco, pipes y redirecciones en las comillas simples.
- echo '\$USER' debe imprimir \$USER.
- Nada debe interpretarse.

⊗ Yes		imesNo
env		
Comprueba si env muestra la	s variables de entorno actual	es
2/11		
⊗ Yes		imesNo
export		
Exporta variables de entorno		ejas.
Comprueba que funcione cor	rectamente con env.	
⊘ Yes		imesNo
unset		
<ul> <li>Exporta variables de entorno</li> </ul>		ejas.
<ul> <li>Utiliza unset para eliminar alg</li> <li>Verifica el resultado con env.</li> </ul>	junas de ellas.	
vermea er resultado con env.		
⊗ Yes		imesNo
cd		
Utiliza el comando cd para ca	ambiar de directorio de trabajo	o, y asegúrate de que estás en el directorio
correcto con /bin/ls.		
<ul><li>Repite esto varias veces, algo</li><li>Prueba '.' y '' como argumen</li></ul>		ue no.
• Frueba . y como argumer	ilos.	
⊗ Yes		imesNo
pwd		
Utiliza el comando pwd.		
Repite esto varias veces en r	núltiples directorios.	
✓ Yes		×No
⊗ Yes		∕ NO
Rutas relativas		
• Ejecuta comandos, pero esta	vez con rutas relativas.	
<ul> <li>Repite esto múltiples veces e</li> </ul>	n múltiples directorios con ru	tas relativas complejas (muchos)
⊗ Yes		imesNo
La variable de entorno PATH		
<ul> <li>Ejecuta comandos, pero esta</li> </ul>	vez sin rutas (ls, wc, awk, etc	o).
Elimina \$PATH y verifica que		
<ul> <li>Establece \$PATH para varios de izquierda a derecha.</li> </ul>	directorios (directorio1:direct	orio2) y valida que los directorios se comprueban
⊗ Yes		imesNo
Redirecciones		
Ejecuta comandos con redire	cciones < v/o >.	
,	J 1	

• Repite esto varias veces con diferentes comandos y argumentos, cambia de vez en cuando > por >>.

Comprueba si varias redirecciones del mismo tipo fallar	1.
• Prueba la redirección << (no necesita actualizar el histo	orial).
⊗ Yes	imesNo
lipes	
Ejecuta comandos con pipes como 'cat file   grep bla   r	
<ul> <li>Haz esto varias veces cambiando comandos y argume</li> <li>Prueba algunos comandos que fallen como 'ls archivoq</li> </ul>	
<ul> <li>Prueba a mezclar pipes y redirecciones.</li> </ul>	dendexiste   grep bia   more .
♂ Yes	×No
uélvete loco, y el historial	
<ul> <li>Escribe una línea de comandos, utiliza ctrl-C y pulsa er ejecutarse.</li> </ul>	nter. El buffer deberá estar vacío y nada deberá
<ul> <li>¿Puedes navegar en el historial con las flechas de arrib</li> <li>Ejecuta comandos que no deberían funcionar como 'wji y muestra un error.</li> </ul>	
"cat   cat   ls" se comporta de forma "normal".	
<ul> <li>Prueba a ejecutar un comando largo con muchísimos a</li> </ul>	rgumentos.
Pásatelo bien con este increíble minishell y disfruta.	
⊘ Yes	imesNo
ariables de entorno	
Ejecuta echo con algunas \$ variables como argumento:	S.
Verifica que \$ se interpreta como una variable de entore	no.
Verifica que las comillas dobles interpolan \$.	
Verifica que \$USER existe o créalo.	
echo "\$USER" deberá imprimir el valor de \$USER.	
⊗ Yes	imesNo
Bonus	
Se tendrá en cuenta tus bonus si y solo si la parte obligatoria naturalmente decir que la parte obligatoria está completa, de p ncluso en casos retorcidos o de mal uso. Si no has conseguio nonus será completamente ignorada.	orincipio a fin, y la gestión de errores es impecable,
and, Or	
Utiliza &&,    y paréntesis con comandos y verifica que f	funciona como en bash.
⊘ Yes	imesNo
/ildCard	
Utiliza wildcards en argumentos para el directorio actua	ıl.
⊗ Yes	imesNo
Sorpresa! (O no)	
<ul> <li>Establece la variable de entorno USER.</li> </ul>	

	eba echo ""\$USER	"", debera impri	IMIL \$09EK .			
	⊗ Yes			imesNo		
Ratir	ngs					
Don't forge	t to check the flag c	orresponding to	the defense			
<b>✓</b> Ok			★ Outstanding project			
Empty w	vork <b>L</b> Incom	plete work	Invalid compilation	Norme	Cheat	Crash
<b>♣</b> Inc	complete group	▲ Conce	cerning situation <b>b</b> Leaks		<b>⊘</b> Forbidden function	
Cond	clusion		<b>⊅</b> Can't support / explain o	code		
	<b>Clusion</b>			oode		
				code		