

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

SCALE FOR PROJECT PIPEX (/PROJECTS/PIPEX)

You should evaluate 1 student in this team



Git repository

git@vogsphere-v2.42madrid.com:vogsphere/intra-uuid-6675d936



Introduction

Por favor, cumple las siguientes reglas:

- Sé educado, cordial, respetuoso y constructivo a lo largo del proceso de evaluación. El bienestar de la comunidad depende de ello.
- Identifica con el estudiante o grupo evaluado los posibles fallos de funcionalidad en su proyecto. Tómate el tiempo de discutir y debatir los problemas que puedan ser identificados.
- Debes considerar que puede haber diferencias en la manera en la que tus compañeros entienden lo que se pide en el enunciado. Mantén siempre una mente abierta y evalúa de la forma más honesta posible. La pedagogía solo es útil si la evaluación entre compañeros se hace bien.

Guidelines

- Evalúa solo el trabajo entregado en el repositorio de Git del estudiante o grupo evaluado.
- Comprueba dos veces que el repositorio de Git pertenece al estudiante o grupo evaluado. Asegúrate de que el proyecto es el que se tiene que evaluar. Comprueba también que "git clone" se utiliza en un directorio vacío.
- Comprueba cuidadosamente que no hay alias maliciosos usados para engañarte y hacerte evaluar algo que no pertenece al repositorio oficial.
- Para evitar sorpresas, revisad juntos los posibles scripts utilizados para facilitar la evaluación (por ejemplo, de pruebas o automatización).

- Si no has completado el proyecto que vas a evaluar, tienes que leer el subject entero antes de empezar el proceso de evaluación.

- Utiliza los posibles flags para reportar un repositorio vacío, un programa no funcional, con fallos de Norma, trampas, etc.

En estos casos, el proceso de evaluación termina y la nota final es 0, o -42 en caso de trampas. Sin embargo, excepto para cuando se detectan trampas, se recomienda revisar juntos el trabajo entregado, para identificar los posibles fallos y evitar cometerlos de nuevo en el futuro.

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/65605/es.subject.pdf>)

Preliminares

Si se sospechan intentos de trampa, el proceso de evaluación termina. Utiliza el flag de "Cheat" para reportarlo. Toma esta decisión con calma, inteligencia, y por favor, con cuidado.

Pruebas preliminares

- La defensa sólo se puede realizar si el estudiante o grupo evaluado está presente. De esta manera, todos aprenden compartiendo conocimiento.
- Si no se ha entregado ningún trabajo (o el archivo, el directorio o el nombre del archivo es erróneo), la nota es 0 y la evaluación se termina.
- Si alguno de los siguientes supuestos ocurre, la evaluación termina:
 - El repositorio de Git está vacío.
 - Hay fallo de Norma.
 - Se detectan trampas, que resulta en -42.
 - Existen fallos de compilación. El Makefile no debe hacer relink.

Si todos los puntos están bien, marca "Sí" y continúa la evaluación.

En el caso contrario, usa el flag apropiado de acuerdo con las indicaciones al final de este scale.

 Yes

 No

Instrucciones generales

Instrucciones generales

- Si el programa principal muestra un error (segfault, bus error, sinsentidos, etc). Utiliza el flag de "Crash".
- El Makefile compila el ejecutable y tiene las normas requeridas.
- El nombre del ejecutable es "pipex".
- No se utilizan funciones prohibidas.

☒ Yes☐ No

Parte obligatoria

El comando `"/pipex archivo1 comando1 comando2 archivo2"` debe actuar como el comando `"< archivo1 comando1 | comando2 archivo2"`.

Gestión de errores y de argumentos

- El programa acepta exactamente 4 argumentos, ni más, ni menos (salvo para los bonus) y solamente en el orden correcto.
- La gestión de errores es correcta; archivos existentes, permisos de archivos, el binario del comando existe, etc.

Si todos los puntos se respetan, marca "Sí" y continúa la evaluación. En caso contrario, la evaluación recibe o "Incomplete work" o el flag que corresponda.

☒ Yes☐ No

El programa

El programa hace lo que se le pide, sin mostrar más información/pasos que el comando shell que debe replicar.

Ejecuta tus propios tests y compara los resultados de shell al salir y el resultado de shell en comparación al del programa.

Si no tienes ideas, mira los ejemplos del enunciado.

Si no se detectan errores, marca "Sí" y continúa. En caso contrario, la evaluación se termina aquí.

☒ Yes☐ No

Extras

Evalúa los bonus si, y solo si, la parte obligatoria está perfecta y completa; y la gestión de errores maneja comportamientos inesperados o de mal uso. En caso de que algún punto de la parte obligatoria no pasara la evaluación, toda la parte extra será completamente ignorada.

múltiples pipes

El programa es capaz de gestionar el uso de múltiples pipes, uno tras otro. Para la parte obligatoria, prueba con pequeños comandos de shell y luego compara el resultado con el original.

Si funciona con dos pipes pero no con cinco, el bonus no cuenta.

✓ Yes

✗ No

<< y >>

El programa replica y permite el uso de << y >>. Prueba varias veces con cosas como : CMD << STOP_VALUE | CMD1 >> file1

✓ 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

💥 Crash

⚠ Concerning situation

💧 Leaks

🚫 Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

Terms of use for video surveillance (<https://profile.intra.42.fr/legal/terms/1>)

Rules of procedure (<https://profile.intra.42.fr/legal/terms/4>)

Declaration on the use of cookies (<https://profile.intra.42.fr/legal/terms/2>)

General term of use of the site (<https://profile.intra.42.fr/legal/terms/6>)

Legal notices (<https://profile.intra.42.fr/legal/terms/3>)

Privacy policy (<https://profile.intra.42.fr/legal/terms/5>)