

From: Pyoyoso pyoyoso@gmail.com
Subject:
Date: February 24, 2022 at 1:16 PM
To: pyoyoso pyoyoso@gmail.com

P

Introduction

Por favor, respeta las siguientes directrices:

- Sé educado, cordial, respetuoso, y constructivo a lo largo del proceso de evaluación. El bienestar de la comunidad depende de ello.
- Identifica con la persona o grupo evaluados los posibles fallos de funcionalidad del trabajo. Tómame el tiempo de discutir y debatir los problemas identificados.
- Debes considerar la existencia de diferencias en cómo tus compañeros pueden haber entendido las instrucciones del proyecto y el alcance de las funcionalidades solicitadas.

Guidelines

- Evalúa solo el trabajo entregado en el repositorio Git de la persona o grupo evaluado en la rama master y en el último commit.
- Comprueba dos veces que el repositorio de Git pertenece al estudiante o grupo evaluado. Asegúrate de que el trabajo entregado es el relativo al proyecto evaluado y que el comando "git clone" se utiliza en una carpeta vacía.
- Comprueba cuidadosamente que no se han utilizado alias maliciosos que puedan engañarte y hacerte evaluar trabajo ajeno al del repositorio oficial.
- Para evitar sorpresas, comprueba cuidadosamente que tanto el evaluador como el evaluado han revisado los scripts utilizados para facilitar la evaluación.
- Si el evaluador no ha completado este proyecto en particular todavía, es obligatorio que lea el subject por completo antes de empezar la evaluación.
- Utiliza las flags disponibles en esta evaluación para indicar un repositorio vacío, un programa disfuncional, con crash, fallos de Norma, compilación inválida, cheating, etc.
- Debes también verificar la ausencia de leaks de memoria. Toda la memoria alocada en el heap debe liberarse correctamente.

Attachments

[subject.pdf](#) [checker_Mac](#) [checker_linux](#)

Parte obliaatoria

Como recordatorio, durante la defensa no se toleran segfaults u otros comportamientos que terminen la ejecución del programa de forma inesperada, temprana o descontrolada. Si esto pasa, tu nota será un 0. Utiliza la flag apropiada. Esta activa se aplica a lo largo de toda la defensa.

Memory leaks

A lo largo de la defensa, presta atención a la cantidad de memoria utilizada por push_swap (utilizando el comando top por ejemplo) para detectar cualquier anomalía y asegurarte de que la memoria reservada es correctamente liberada. Si hay algún memory leak, la nota final es 0.

Yes

No

Gestión de errores

En esta sección, vamos a evaluar la gestión de errores del push_swap. En caso de que cualquiera falle, no se otorgarán puntos a esta sección. Pasa a la siguiente.

- Ejecuta push_swap con parámetros no numéricos. El programa debe mostrar "Error".
- Ejecuta push_swap con un parámetro numérico duplicado. El programa debe mostrar "Error".
- Ejecuta push_swap solo con parámetros numéricos, pero incluyendo uno superior al MAXINT. El programa debe mostrar "Error".
- Ejecuta push_swap sin parámetros. El programa no deberá mostrar nada y deberá devolver el control al usuario.

Yes

No

Push_swap - Prueba de identidad

En esta sección, evaluaremos el comportamiento de push_swap cuando recibe una lista ya ordenada. Ejecuta los siguientes 3 tests. Si cualquiera falla, no se otorgarán puntos a esta sección. Pasa a la siguiente.

- Ejecuta el siguiente comando "\$>./push_swap 42". El programa no deberá mostrar nada (0 instrucciones).
- Ejecuta el siguiente comando "\$>./push_swap 0 1 2 3". El programa no deberá mostrar nada (0 instrucciones).
- Ejecuta el siguiente comando "\$>./push_swap 0 1 2 3 4 5 6 7 8 9". El programa no deberá mostrar nada (0 instrucciones).

Yes

No

Push_swap - Versión simple

Si las siguientes pruebas fallan, no se darán puntos en esta sección. Pasa a la siguiente. Utiliza el binario checker dado adjunto en el proyecto.

- Ejecuta '\$>ARG="2 1 0"; ./push_swap \$ARG | ./checker_OS \$ARG'. Verifica que el checker muestra "OK" y que el tamaño de la lista de instrucciones del push_swap es 2 o 3. De otro modo esta

prueba es fallida.

Yes

No

Otras pruebas simples.

Ejecuta las dos siguientes pruebas. Si cualquiera falla, no se darán puntos a esta sección. Pasa a la siguiente. Utiliza el checker dado como archivo adjunto.

- Ejecuta '\$>ARG="1 5 2 4 3"; ./push_swap \$ARG | ./checker_OS \$ARG'.

Verifica que el checker muestra "OK" y que el tamaño de la lista de instrucciones del push_swap no supera las 12. El tamaño más pequeño posible de la lista es 8.

- Ejecuta '\$>ARG="<5 valores aleatorios>"; ./push_swap \$ARG | ./checker_OS \$ARG' y reemplaza el texto dado por 5 números aleatorios.

Verifica que el checker muestra "OK" y que el tamaño de la lista de instrucciones del push_swap no es mayor que 12. En caso contrario, se considera un fallo. Tendrás que comprobar que este programa no se ha diseñado explícitamente para responder correctamente a pruebas de esta magnitud. Repite esta prueba varias veces antes de validarla.

Yes

No

Push_swap - Nivel intermedio

Si cualquiera de las siguientes pruebas falla no se otorgarán puntos en esta sección. Pasa a la siguiente. Utiliza el checker dado adjunto.

- Ejecuta '\$>ARG="<100 valores aleatorios>"; ./push_swap \$ARG | ./checker_OS \$ARG' y reemplaza el texto predeterminado por 100 valores válidos aleatorios. Comprueba que el programa checker muestra "OK" y puntúa en base al tamaño de la lista de instrucciones.

- 5 puntos para menos de 700

- 4 puntos de 701 a 900 instrucciones

- 3 puntos de 901 a 1100 instrucciones

- 2 puntos de 1101 a 1300 instrucciones

- 1 punto de 1301 a 1500 instrucciones

Deberás comprobar que el programa no se ha escrito con el objetivo de contestar correctamente a esta prueba. Deberás repetir esta prueba un par de veces con distintas permutaciones para validarla.

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



Push_swap - Versión Avanzada

Si alguna de las siguientes pruebas falla, no se otorgarán puntos a este ejercicio. Pasa al siguiente. Utiliza el checker dado como archivo adjunto.

- Ejecuta '\$>ARG="<500 valores aleatorios>"; ./push_swap \$ARG |

./checker_OS \$ARG' y reemplaza el texto dado por 500 números válidos aleatorios (con suficiente tamaño para llenar la lista / lista sorted). Verifica

aleatorios (por que si no te idas a llamar john/jane script). verifica que el programa checker muestra "OK" y puntúa en base al tamaño de la lista de instrucciones.

- 5 puntos si es inferior a 5500 instrucciones
- 4 puntos para un tamaño comprendido entre 5501 y 7000 instrucciones
- 3 puntos si la lista tiene de 7001 a 8500 instrucciones
- 2 puntos en caso de que su tamaño sea de 8501 a 10000 instrucciones
- 1 punto si, finalmente, el tamaño es de 10001 a 11500 instrucciones

Deberás comprobar que el programa no se desarrolló específicamente para contestar correctamente a esta prueba. Realiza esto con dos permutaciones distintas antes de validar el ejercicio.

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



Extras

Recuerda: durante la duración de la defensa, ningún segfault u otras formas de comportamiento indefinido, prematuro, descontrolado o inesperado deberá ser tolerado. En caso de que así haya ocurrido, la nota final será 0. Utiliza la flag pertinente. Esta norma aplicará a lo largo de todo el proceso de evaluación. Tus bonus serán evaluados siempre y cuando tu parte obligatoria esté PERFECTA. Esto implica tener la parte obligatoria completa, de principio a fin, y tu gestión de errores debe ser impecable, incluso en casos de mal uso. Por lo que, si la parte obligatoria no consigue todos los puntos, los bonus deberán ser totalmente ignorados.

Checker - gestión de errores

En esta sección, evaluaremos la gestión de errores del checker. Si cualquiera de estas falla, no se otorgarán puntos en esta sección. Continúa con la siguiente.

- Ejecuta el checker con parámetros no numéricos. El programa deberá mostrar "Error".
- Ejecuta el checker con parámetros numéricos duplicados. El programa deberá mostrar "Error".
- Ejecuta el checker con solo parámetros numéricos incluyendo uno superior a MAXINT. El programa mostrará "Error".
- Ejecuta el checker sin parámetros. El programa no deberá mostrar nada y deberá devolver la entrada al usuario.
- Ejecuta el checker con parámetros válidos, y escribe una acción que no exista durante la fase de instrucciones. El programa deberá mostrar "Error".
- Ejecuta el checker con parámetros válidos, y escribe una acción con uno o varios espacios antes y/o después de la acción en la fase de instrucciones. El programa deberá mostrar "Error".

YES

NO

Checker - pruebas falsas

En esta sección, evaluaremos la capacidad del checker para gestionar una lista de instrucciones que no ordenen la lista. Ejecuta las siguientes dos pruebas. Si una falla, no des puntos en esta sección y avanza a la siguiente.

No olvides presionar Ctrl+D para terminar la lectura en la fase de instrucciones.

- Ejecuta el checker con el siguiente comando "\$>./checker 0 9 1 8 2 7 3 6 4 5" y escribe la siguiente lista de acciones "[sa, pb, rrr]". El checker debería mostrar "KO".

- Ejecuta el checker con una list válida de parámetros de tu elección y escribe una lista de instrucciones válida que no ordene los números. El checker deberá mostrar "KO". Deberás verificar que el programa no se desarrolló para contestar correctamente a las pruebas escritas en esta evaluación. Prueba un par de veces con combinaciones distintas antes de validar el punto.

Yes

No

Checker - pruebas reales

En esta sección, evaluaremos la capacidad del checker para gestionar una lista de instrucciones que ordenen la lista. Ejecuta las siguientes dos pruebas. Si una falla, no des puntos en esta sección y avanza a la siguiente.

No olvides presionar Ctrl+D para terminar la lectura en la fase de instrucciones.

- Ejecuta el checker con el siguiente comando "\$>./checker 0 1 2". Pulsa Ctrl+D sin escribir ninguna instrucción. El programa deberá mostrar "OK".

- Ejecuta el checker con el siguiente comando "\$>./checker 0 9 1 8 2". Después escribe la siguiente lista de instrucciones "[pb, ra, pb, ra, sa, ra, pa, pa]". El checker deberá mostrar "OK". Deberás verificar que el programa no se desarrolló para contestar correctamente a las pruebas escritas en esta evaluación. Prueba un par de veces con combinaciones distintas antes de validar el punto.