UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Herramientas para manejar errores (par 2)

Oscar Evanilson Gutiérrez Pérez

Carrera: Ingeniería en Computación

Código de estudiante: 219748308

1 de febrero de 2022

En la parte 1 de esta actividad, investigamos acerca de herramientas para manejar errores en un lenguaje de programación, en mi caso fue en el lenguaje de C.

Las herramientas que encontré fueron perror y try-catch por lo que me adentré a mi editor de texto y realicé unos ejemplos muy sencillos utilizando estas herramientas, a continuación lo realizado:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  #define try bool __HadError=false;
6  #define catch(x) ExitJmp:if(__HadError)
7  #define throw(x) __HadError=true;goto ExitJmp;
```

Lo primero de todo fue incluir las librerías necesarias para estas herramientas, en este caso la única adicional es stdbool que nos ayuda en la declaración de try, catch y throw.

Después de eso se declaran estas tres palabras y se les asigna la función que deben de realizar al llamarlas.

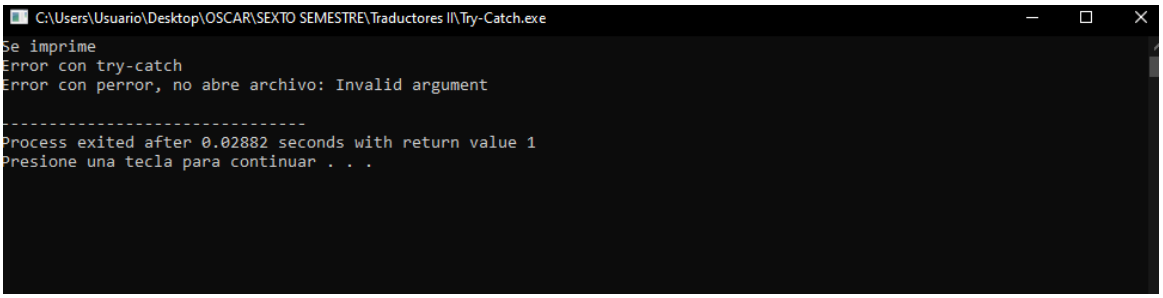
```
9   int main(int argc, char *argv[])
10  {
11      try
12      {
13          printf("Se imprime\n");
14          throw();
15          printf("No se imprime\n");
16      }
17      catch(...)
18      {
19          printf("Error con try-catch\n");
20      }
```

Ahora dentro de la función principal del programa utilizamos primero la herramienta de try y catch, para esto ponemos dentro de las llaves de try lo que queremos que realice el programa y en caso de que se encuentre un error se llama a la función de catch que dará un salto al programa hacia la sección de catch, donde debemos de indicar que se debe de hacer en caso de haber ese error.

```
22      FILE *fout;
23
24    if ((fout = fopen(argv[1], "w")) == NULL) {
25        perror("Error con perror, no abre archivo");
26        return EXIT_FAILURE;
27    }
28     return EXIT_SUCCESS;
29  }
```

El segundo ejemplo dentro de este programa es el de perror donde se indica con un if, el error que se quiere manejar y dentro de la función perror, lo que el programa debe de hacer en caso de que suceda este error.

Captura de los mensajes de ambos errores:



Link de supositorio en GitHub para revisar el código:

https://github.com/oscarevanilson/Computacion-Tolerante-a-Fallas

Conclusiones

Esta actividad de dos partes me pareció muy completa para comprender el tema de las herramientas para manejar errores en los lenguajes de programación. Me interesa conocer más acerca de las herramientas que investigue y utilice en el ejemplo porque, aunque le di un uso muy sencillo para los ejemplos de esta actividad, siento que pueden ser muy útiles a la hora de programar un software de

manera cotidiana y creo que me pueden ayudar a prevenir muchos errores en mis tareas y futuros trabajos. Junto a esto espero que a lo largo del semestre y del transcurso de esta materia conozca más herramientas en los demás lenguajes de programación que uso cotidianamente, así como profundizar el uso de estas herramientas para que domine las opciones que estas me pueden llegar a brindar en los programas.

Me gustó esta actividad y estoy satisfecho con los resultados que obtuve de esta, así como los nuevos aprendizajes al realizarla y espero que estos continúen llegando en las próximas actividades.