

Universidad Central de Venezuela

Facultad de Ciencias

Computación

Organización y Estructura del Computador II

Proyecto #2

Jesús Martínez, CI: 21.194.440

Alejandra Vento, CI: 26.648.099

Informe

En el presente informe se detallan los métodos y partes en las que fue realizado el proyecto de simulación de detección de errores, nuestra solución se encuentra estructurada de 5 fases:

Primeramente en la sección **.data**, fueron especificadas las variables: **"si"**, que posee el mensaje de alerta que se mostrara de haber sido modificado el archivo, **"no"**, que posee el mensaje de que no ocurrió ningún error si no fue modificado el archivo, **"error"**, que posee el mensaje inicial que te permite decidir si vas a insertar un error o no que será guardado en la variable **"N"** para luego poder comparar, **"fout"**, que posee el nombre del archivo que se va a leer, **"fin"**, que posee el nombre del archivo que se va a crear y donde se va a escribir, **"buffer"**, que reserva un espacio de 1024 donde se va a almacenar la información leída.

-Fase 1: En esta fase, se imprime por consola el mensaje que te permite escoger si se va a insertar un error o no, y se lee la selección como **"s"** o **"n"**, que corresponden a si y no respectivamente.

-Fase 2 (Open File): En esta fase, se hace la llamada al sistema que permite abrir el archivo especificado en **"fout"**, y se guarda el apuntador a éste en un registro tipo **S**, para poder llevar a cabo las siguientes fases.

-Fase 3 (Read From File): En esta fase, se hace la llamada al sistema que permite leer el archivo que fue previamente abierto y se copia toda la información contenida en él, a la variable **"buffer"**, luego de esto en un ciclo se cuenta la cantidad de caracteres contenidos en el archivo para conocer su tamaño y a la vez se registra su valor ASCII en una suma total que necesitaremos más adelante para comparar. Terminado el ciclo se calcula el carácter localizado en la mitad del archivo donde se fija el apuntador que dependiendo si la selección fue **"s"** o **"n"**, de ser **"s"** modificara el valor del carácter por el que este 10 posiciones después en el código ASCII (**esta variación corresponderá a ser el error insertado**), o de ser **"n"** lógicamente este carácter no se verá afectado, para cualquiera de las dos opciones al terminar acaba esta fase y se avanza a la siguiente.

-Fase 3 (Write The File): En esta fase, se hace la llamada al sistema que permite escribir en el archivo especificado en **"fin"**, que será creado en ese momento y almacenara la información contenida en la variable **"buffer"**.

-Fase 4 (Open File de Salida): En esta fase, se hace la llamada al sistema que permite abrir el archivo **"fin"**, que fue previamente creado y que almacena la información modificada en la fase anterior. También se guarda el apuntador al archivo para poder proseguir en la ejecución.

-Fase 5 (Read From File): En esta fase, se hace la llamada al sistema que permite leer el archivo especificado en **"fin"** y se recorre con el objetivo de calcular su tamaño y poder comparar con el tamaño del archivo anterior **"fout"**, que ya fue previamente calculado en la **Fase 3**, y así determinar si fue insertado algún error en éste o no. Si se detecta un error en el archivo, se determina que ocurrió un error en la transmisión, por lo que muestra por consola el mensaje: **"Alerta! Ocurrió un error en la transmisión"**, almacenado en el string **"si"**, pero de no ser así se mostrara el mensaje **"No hubo error"**, almacenado en el string **"no"**, ya luego de mostrar el mensaje que corresponda, se hace el llamado al sistema para finalizar la ejecución del programa y éste termina.