INSTITUTO POLITECNICO NACIONAL

Escuela Superior de Cómputo

PRACTICA 1

Autor:
Gomez Hernandez Alan Javier

12 octubre, 2021—

PROGRAMA COMBINATORIO

RESUMEN

Realizar un programa donde el usuario ingrese n número para obtener cadenas

$$2^n$$

y solo se aceptará del 0 a 1000, si se detecta que el usuario no ingreso nada se le dará un numero aleatorio, y por cadena resultante se contaran los unos para hacer una gráfica. Para terminar el programa se le preguntara si se ingresara otro número o saldrá del programa.

INTRODUCCION

Para realizar este programa se requieren de conocimientos previos como es el manejo de archivos, math, recursividad, funciones, uso de graficas con LabView, y uso de cadenas para almacenar datos en C. Pero antes de esto el conocimiento previo de el universo de las cadenas binarias (Sigma). Dada una "n". Para empezar un archivo es un conjunto de datos estructurados en una colección de entidades elementales o básicas denominadas registros que son de igual tipo y constan a su vez de diferentes entidades de nivel más bajos denominadas campos. Hay dos tipos de archivos, archivos de texto y archivos binarios. Un archivo de texto es una secuencia de caracteres organizadas en líneas terminadas por un carácter de nueva línea. En estos archivos se pueden almacenar canciones, fuentes de programas, base de datos simples, etc. Los archivos de texto se caracterizan por ser planos, es decir, todas las letras tienen el mismo formato y no hay palabras subrayadas, en negrita, o letras de distinto tamaño o ancho.Y para este trabajo sera el que ocuparemos.

Nombre	Función
fopen()	Abre un archivo.
fclose()	Cierra un archivo.
fgets()	Lee una cadena de un archivo.
fputs()	Escribe una cadena en un archivo
fseek()	Busca un byte especifico de un archivo.
fprintf()	Escribe una salida con formato en el archivo.
fscanf()	Lee una entrada con formato desde el archivo.

Figure 1: librería STDIO.H para el uso de esta funcion

Por consiguiente lo que es recursividad a la técnica consistente en definir una función en términos de sí misma. Puesto que en C una función puede llamar a

otras funciones, se permite que una función también pueda llamarse a sí misma. Despues tenemos la funcion que nos ayudara a separar el cuerpo principal del codigo con respecto a las operaciones que se realizarán para obtener el resultado. Finalmente lo que es un arreglo Los arrays son variables estructuradas, donde cada elemento se almacena de forma consecutiva en memoria. Ya que obtendremos valores y en general el proposito es generar una grafica se utilizara LabView que es realizado para ese proposito.

DESARROLLO

0.1 PROPUESTA DE SOLUCION

Se propone realizar de primera instancia un if que nos ayudara a ver si el usuario ingreso un numero entero o bien un 0 aleatorio, suponiendo que quiera ingresar un numero no valido fuera del parámetro se le pediré ingresar ese valor otra vez con do while. Ahora entraremos a un ciclo con do while para realizar la sencilla operación de salir o no del programa en ambos casos se les asigna una variable. Pasando esta sección de pedir valores y asignar las librerías include ¡stdio.h¿, include ¡math.h¿ para el procedimiento, se crean ficheros para almacenar el numero de 1 generados por cadena y finalmente graficarlos, siendo dos casos uno para n y números de 1, y otro fichero con el numero de 1. Pero bueno estructuralmente se crea +w el fichero en modo write y validamos si esta o no con NULL, mandamos mensaje en caso de error y sale del programa. Si se crea seguimos con el programa, a hora iniciamos un arreglo donde guardaremos los números de unos y variables, y con la función pow aumentamos el numero del usuario o aleatorio, en 2 elevado a ese numero, esto nos ayudara ya que teniendo ese dato es el numero de binarios que pediremos como limite al programa, esto se logra con un For que obtendrá esa cadena de números hasta que termine el número elevado (sigma). Siguiendo con el programa entramos a la función recursiva que nos ayudará para obtener el numero binario y cada ves que sea vea uno lo estará contando, se hace la llamada a la función y la almacenamos en una variable. La función pide dos valores una de control para la suma de los unos y otra donde estaremos dividiendo el numero en dos, cada ves que se entre a la función si el residuo es igual a 1 significa que es impar es decir un 1 y en caso contrario se dará un 0, pero como lo que nos interesa son los 1, contamos cuantas veces se nos dará n porcentaje 2 == 1 y eso es lo que se guardara en la variable que llama a la función, evidentemente guardamos este dato ya que la recursividad deje de llamarse a si misma, llamamos el espacio en memoria del fichero y con fprintf guardamos el n numero que lleva el ciclo y el valor que dio la función (es decir el numero de unos en ese numero que entro), esto para la gráfica. Iremos mostrando los datos de 0 y 1 que conforman el numero binario, separados por comas, antes dicho esto y por estética los unos por n se irán guardando en un arreglo y mostraremos con otro for, aparte de que ya están guardadas en el fichero correspondiente. Finalmente preguntamos si se sigue con el programa o no con el do while antes explicado con un valor

entero y otro while que valida si es alguno de estos dos, si se ingresa otra ves limpiamos la pantalla.

0.2 CODIGO

```
#include <stdio.h>
#include <math.h>


int decabin (int n,int cont) {

    if (n) {
      if((n%2)==1)
         cont++;
        cont = decabin(n / 2,cont);
        printf("%d", n % 2);
    }
    return cont;
}

void main(){
int var=0;
do
{

/*CREACION DE FICHEROS*/
FILE *fichero;
    fichero = fopen("cordenadas1.txt", "w+");
    if(fichero == NULL ) {
        printf("No fue posible abrir/crear el archivo\n");
        return;
   }
   FILE *fichero2;
   fichero2 = fopen("cordenadas2.txt", "w+");
    if(fichero == NULL ) {
        printf("No fue posible abrir/crear el archivo\n");
        return;
   }

/*INSTANCIA DE VALORES Y OBTENCION DE DATOS*/
    int numero=0,j=0,n=0;
    char unos[100000];
    do{/*while que nos ayudara a validar valores fuera del parametro*/

    printf("Ingrese un entero si se ingresa 0 se dara un numero aleatorio: ");
```

```
    scanf("%d", &numero);

    }while(numero<0 || numero>1000);

    if( numero == 0){
numero = rand() % 1000;
printf("Numero aleatorio:%d\n",numero);
}

/*REALIZACION DEL DESARROLLO */
int pt = pow(2,numero);

printf("{0");
    for (j = 0;j < pt ; j++)
    {
    n = decabin(j,n);/*-----------> LLAMADA A LA FUNCION*/
    /*GUARDAMOS EN EL FICHERO*/
    fprintf(fichero,"%d       %d\n",j+1,n);
    fprintf(fichero2,"%d\n",n);

     printf(",");
     unos[j] = n;
     n=0;
    }
printf("]\n");

/*MOSTRAR DATOS*/

printf("Numero de unos por numero:\n{");
for (j = 0;j < pt ; j++){
printf("%d,",unos[j]);
}
printf("}\n");

/*VALIDACION SI SE INGRESARA OTRO NUMERO*/
printf("\n\n ¿Se ingresara otra cadena? \nsi = presione 1 \nno = presione 0\n");
scanf("%d",&var);
system("cls");

}while(var==1);


}
```

0.3 FUNCIONAMIENTO Y PRUEBAS

0.3.1 Prueba con n=25

Como se observara se empezara a correr el programa y siendo un numero n = 25 sera algo grande lo que presentara el compilador y por ende tardara por el procesador y se vera de la sigueinte manera hasta acabar.

Figure 2: Compilación de codigo

Esta parte de las coordenadas las pasamos a Excel para crear las respectivas tabulaciones entre columnas y filas, y asi poder hacer la grafica.

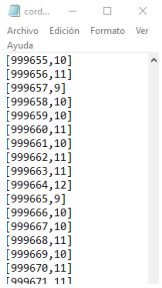

Figure 3: Creando el diagrama

0.4 GRAFICA

Utilizaremos LabView para extraer los datos del txt que obtuvimos de la hoja de excel con los tabuladores respectivos, lo que nos queda de la sigueinte manera, insertando matrices de la tabla, grafica, path y arrays para insertarlos en las salidas respectivas.

En esta parte crearemos un nuevo proyecto. Se ven diferentes carpetas, pero será una creación normal.

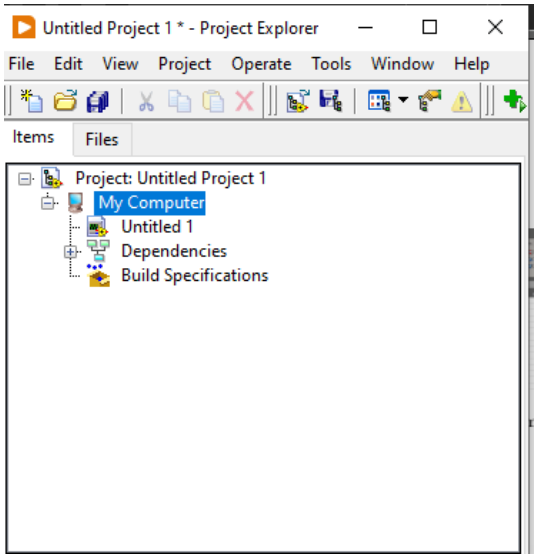

Figure 4: Creando el diagrama

Uno ves creemos este, iremos construyendo el diagrama de conexiones para que queden interconectados y funcionen como una gráfica. Primero entramos a las funciones con clic derecho y buscamos path (este nos ayudara a poner la ruta donde están guardados los datos para graficar), después con clic derecho usaremos una matriz para pasar los números o coordenadas del txt a esta matriz por ello la conectamos así en la primera entrada, después la salida de la matriz ira conectada a un all rows que mostrara esa matriz de coordenadas, después se conecta a un index de arreglo que acomodara las filas y columnas a enteros, esta salida del Index va conectada en dos partes pero ambas en el build Array, que junta los datos, pero por separado y los manda a la gráfica finalmente.

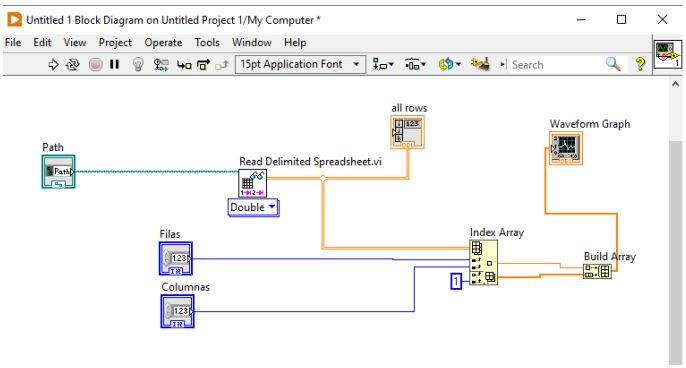

Figure 5: El entorno de conexiones queda de esta manera

Lo antes se ira acomodando en el entorno grafico evidentemente no se verán estas conexiones que ya se hicieron pero lo primero que haremos es que con el PATH exportamos los datos del txt y automáticamente se cargara todo lo demás y queda de la siguiente manera.

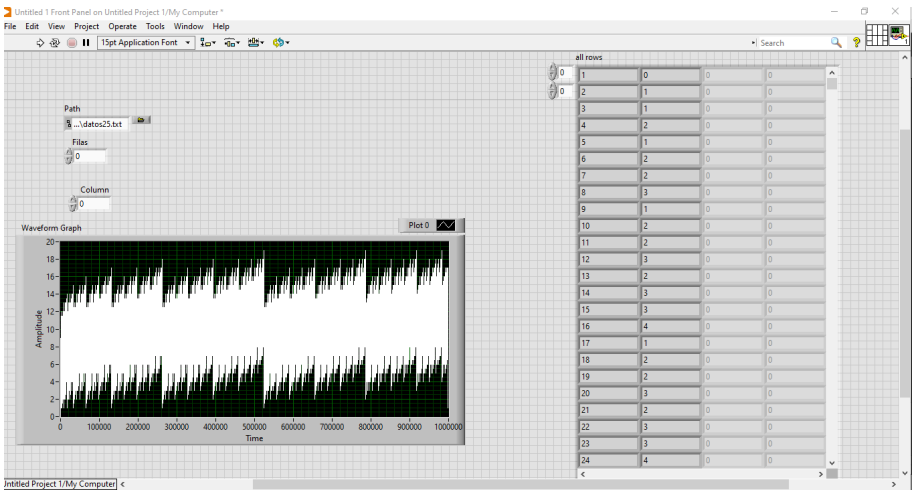

Figure 6: Finalmente el entorno de graficacion queda de esta manera

CONCLUSION

A lo largo de la práctica y realización de este proyecto, se encontró una complejidad en la cual, pasaba por varios problemas, evidentemente el exceso de datos pedidos cuando se ingresa un numero tan grande se notaba en el equipo, aparte de que se podía observar como trabajaba ´para encontrar el numero binario hasta el límite de sigma n, ingresado, aparte de esto también esta el cómo organizar la información para poder convertir un dato grande a uno con el que se pueda trabajar como son los 1 y 0, binarios. Bueno abordando mas la complejidad y como se optó por la solución, fue acertada para obtener ese numero de unos para graficar, y mostrar el numero binario, se tuvieron algunos problemas por si usar recursividad o funcional, por lo que tener una ordinación nos ayudó para encontrar algún fallo dentro del código, el exportar los datos también fue de gran ayuda para la gráfica, ya que el gran problema que se obtuvo fue extraer este dato, sin embargo con el tiempo y observación se detectó que solo era irnos por los números impares con residuo en 1, después de este análisis los archivos fueron esenciales para guardar el registro pasarlo a Excel y LabVIEW. Sobre todo, nos ayudó a comprender como puede trabajar la maquina con recursos y el tiempo de funcionalidad con este algoritmo.