

Universidad de Costa Rica

Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
Plataformas Abiertas
I ciclo 2019

Informe
Laboratorio 4 Y 5

Structs en C

Esteban Naranjo Garita B55004

Grupo 03

Profesor: Javier Pacheco

Fecha de entrega

Índice

1. Introducción	1
2. Nota teórica	2
3. Desarrollo	3
3.1. Lista de control sencilla	3
3.2. Lista compleja con recursividad	5
4. Conclusiones	8

Índice de figuras

1. Introducción

Para el presente trabajo se pide realizar un programa en C que lleve un registro de estudiantes, el laboratorio 4 se pedía un programa sencillo, con una función set y una print, un struct, un arreglo y un enum, para el laboratorio 5 se pide hacerlo más complejo añadiendo una función que me pida el número para darle el tamaño al array, además de preguntas como si los datos están bien o si desea terminar o volver a llenar y otras funciones.

2. Nota teórica

En esta sección se explicaran algunas funciones utilizadas para realizar el laboratorio, se explicaran mejor en los comentarios del código.

A) Struct:

Un struct es una función de C que permite hacer una clase donde se guardan datos de forma ordenada, se puede ver como casillas tipo fila y se puede acceder a ella con punteros de forma muy sencilla, además que se puede hacer un arreglo de structs y de esta forma reducir muchísimo el código en ciertos casos como el de este laboratorio.

B) Enum:

Enum funciona para enumerar una lista de cosas y poder darle un nombre a ese numero, de esta forma facilita la construcción del código para el programador y el entendimiento y la lectura para el lector.

3. Desarrollo

3.1. Lista de control sencilla

```

1  #include <stdio.h>
2
3  #define SIZE 5
4  //enum enumera una lista de palabras, de esta forma se puede hacer una pregunta cerrada al usuario
5  //de una forma mas rapida y facil y asi tambien eliminar el factor de que el usuario escriba mal
6  //el
7  //nombre del dato que necesito
8  enum enfasis {
9  __ELECTRONICA=1,
10 __REDES,
11 __POTENCIA,
12 __CONTROL
13 };
14 //con struct se crea una estructura en forma de vector fila donde puedo guardar distintas
15 //variables para programar
16 //una lista de una forma mas ordenada y practica
17 struct datos_persona {
18 __char nombre[15];
19 __char apellido[15];
20 __int telefono;
21 __char carnet[10];
22 __enum enfasis Enfasis;
23 };
24
25 //se crean una funcion set que pida valores al usuario y que con el puntero del estruct los vaya
26 //almacenando en las
27 //casillas correspondientes
28 void set_struct (struct datos_persona *ptr_struct, int selector){
29 __printf("\nDigite los datos del estudiante %d\n", selector+1);
30 __printf("Digite su nombre.\n");
31 __scanf("%s", ptr_struct->nombre);
32 __printf("Digite su apellido.\n");
33 __scanf("%s", ptr_struct->apellido);
34 __printf("Digite su numero de telefono.\n");
35 __scanf("%d", &ptr_struct->telefono);
36 __printf("Digite su carnet.\n");
37 __scanf("%s", ptr_struct->carnet);
38
39 __printf("Elija el numero que corresponde a su enfasis:\n1.ELECTRONICA.\n2.REDES.\n3.POTENCIA.\n4.
40 CONTROL.\n");
41 __int seleccion=0;
42 __scanf("%d", &seleccion);
43 __switch (seleccion){
44 __case 1:
45 __ptr_struct -> Enfasis = ELECTRONICA;
46 __break;
47 __case 2:
48 __ptr_struct -> Enfasis = REDES;
49 __break;
50 __case 3:
51 __ptr_struct -> Enfasis = POTENCIA;
52 __break;
53 __case 4:
54 __ptr_struct -> Enfasis = CONTROL;
55 __break;
56 __} //fin del switch
57
58 //fin del set
59
60 //se crea una funcion print que imprima los datos de las casillas del struct tambien con ayuda del
61 //puntero
62 void print_struct (struct datos_persona *ptr_struct, int selector){
63 __printf("\nEstudiante %d:\n", selector+1);

```

```

63  __printf("\nNombre:_%s", ptr_struct->nombre);
64  __printf("\nApellido:_%s", ptr_struct->apellido);
65  __printf("\nNumero_de_telefono:_%d", ptr_struct->telefono);
66  __printf("\nCarnet:_%s", ptr_struct->carnet);

67  //como el enum da el numero de lugar donde esta la palabra entonces aqui se pasa ese numero a el
    dato que se quiere mostrar
    //en este caso el enfasis
68  __switch (ptr_struct->Enfasis){
69  __case 1:
70  ____printf("\nEnfasis:_Electronica\n");
71  __break;
72  __case 2:
73  ____printf("\nEnfasis:_Redes\n");
74  __break;
75  __case 3:
76  ____printf("\nEnfasis:_Potencia\n");
77  __break;
78  __case 4:
79  ____printf("\nEnfasis:_Control\n");
80  __break;
81  __default:
82  ____printf("\nEnfasis:_Se_ingreso_un_numero_equivocado\n");

83  __} //fin del switch
84  } //fin del print
85  //en el main se llaman las funciones para su ejecucion
86  int main(){
87  __//se crea la variable dato tipo array para decirle al puntero que inicie en la primer casilla de
    el struct
88  __struct datos_persona dato[SIZE];
89  __//el puntero tipo struct datos_persona para poder moverme entre las "casillas" del struct
90  __struct datos_persona *ptr_struct;
91  __ptr_struct=&dato[0];
92  __int selector;
93  __//se hace un for para que escriba los valores recibidos
94  __for(selector=0;selector<SIZE;selector++){
95  __set_struct(ptr_struct,selector);
96  __//al aumentar el valor del puntero se cambia el espacio del arreglo
97  __ptr_struct++;
98  __}

101  __//se vuelve a mandar el puntero a la posicion inicial en el struct para imprimir desde el inicio
102  __ptr_struct=&dato[0];

103  __for(selector=0;selector<SIZE;selector++){
104  __print_struct(ptr_struct,selector);
105  __ptr_struct++;
106  __}
107  __return 0;
108  } //fin del main
109  } //fin del main

```

3.2. Lista compleja con recursividad

```

1  #include <stdio.h>
3  #include <stdbool.h>
4  #include <stdlib.h>
5
7  //enum enumera una lista de palabras, de esta forma se puede hacer una pregunta cerrada al usuario
8  //de una forma mas rapida y facil y asi tambien eliminar el factor de que el usuario escriba mal
9  //el
10 //nombre del dato que necesito
11 enum enfasis {
12     __ELECTRONICA=1,
13     __REDES,
14     __POTENCIA,
15     __CONTROL
16 };
17 //con struct se crea una estructura en forma de vector fila donde puedo guardar distintas
18 //variables para programar
19 //una lista de una forma mas ordenada y practica
20 struct datos_persona {
21     __char nombre[15];
22     __char apellido[15];
23     __int telefono;
24     __char carnet[10];
25     __enum enfasis Enfasis;
26 };
27 //esta funcion va a modificar el tamaño del arreglo de structs dependiendo del número que digite
28 //el usuario y se extrae con un puntero
29 void num_datos(int *ptr_datos){
30     __printf("\nDigite la cantidad de estudiantes que desea registrar\n");
31     __scanf("%i", ptr_datos);
32 }
33 //se crean una funcion set que pida valores al usuario y que con el puntero del struct los vaya
34 //almacenando en las
35 //casillas correspondientes
36 void set_struct (struct datos_persona *ptr_struct, int selector){
37     __printf("\nDigite los datos del estudiante_%d\n", selector+1);
38     __printf("\nDigite su nombre.\n");
39     __scanf("%s", ptr_struct->nombre);
40     __printf("\nDigite su apellido.\n");
41     __scanf("%s", ptr_struct->apellido);
42     __printf("\nDigite su número de teléfono.\n");
43     __scanf("%d", &ptr_struct->telefono);
44     __printf("\nDigite su carnet.\n");
45     __scanf("%s", ptr_struct->carnet);
46     __int seleccion=0;
47     __do{
48         __printf("Elija el número que corresponde a su enfasis:\n1.ELECTRONICA.\n2.REDES.\n3.POTENCIA.\n4.CONTROL.\n");
49         __scanf("%d", &seleccion);
50         __if(seleccion>4){
51             __printf("\nSe ingreso un número equivocado, intente nuevamente\n\n");
52         } else{
53             __switch (seleccion){
54                 __case 1:
55                     __ptr_struct -> Enfasis = ELECTRONICA;
56                     __break;
57                 __case 2:
58                     __ptr_struct -> Enfasis = REDES;
59                     __break;
60                 __case 3:
61                     __ptr_struct -> Enfasis = POTENCIA;
62                     __break;
63                 __case 4:
64                     __ptr_struct -> Enfasis = CONTROL;
65                     __break;
66             } //fin del switch

```



```

65  ____}
66  ____} while ((seleccion <= 0) || (seleccion > 4));
67  } //fin del set
    //se crea una funcion print que imprima los datos de las casillas del struct tambien con ayuda del
    puntero
69  void print_struct (struct datos_persona *ptr_struct, int selector){
    __printf("\nEstudiante:_%d:\n", selector+1);
71  __printf("\nNombre:_%s", ptr_struct->nombre);
    __printf("\nApellido:_%s", ptr_struct->apellido);
73  __printf("\nNumero_de_telefono:_%d", ptr_struct->telefono);
    __printf("\nCarnet:_%s", ptr_struct->carnet);
75
    //como el enum da el numero de lugar donde esta la palabra entonces aqui se pasa ese numero a el
    dato que se quiere mostrar
77  //en este caso el enfasis
    __switch (ptr_struct->Enfasis){
79  ____case 1:
        ____printf("\nEnfasis:_Electronica\n");
81  ____break;
        ____case 2:
83  ____printf("\nEnfasis:_Redes\n");
        ____break;
85  ____case 3:
        ____printf("\nEnfasis:_Potencia\n");
87  ____break;
        ____case 4:
89  ____printf("\nEnfasis:_Control\n");
        ____break;
91  ____} //fin del switch
93  } //fin del print

95  //esta funcion contiene la logica del programa, se saca del main para ordenar el programa y
    practicar con los parametros y las funciones
    void logica(int selector, struct datos_persona *ptr_struct, _Bool datos_ok, _Bool terminar, _Bool
        todos, int bool_datos, int bool_todos, int bool_terminar, int estudiante, int cantidad, struct
        datos_persona dato[]){
97
    ____ptr_struct=&dato[0];
99  ____//se hace un for para que escriba los valores recibidos
    ____for(selector=0; selector<cantidad; selector++){
101  ____set_struct(ptr_struct, selector);
        ____//al aumentar el valor del puntero se cambia el espacio del arreglo
103  ____ptr_struct++;
        ____}
105  ____//se crea un ciclo que pregunte si se desea salir y se repite mientras que terminar sea 0 o
        false
        ____do{
107  ____//se utilizan variables booleanas para facilitar la comprension del codigo
        ____//este switch se utiliza en forma de menu para imprimir o corregir
109  ____printf("\nLos_datos_estan_correctos?\n0.No\n1.Si\n");
        ____scanf("%d",&bool_datos);
111  ____datos_ok=bool_datos;
        ____switch(datos_ok){
113  ____case false:
            ____printf("\nDesea_corregir_los_datos_de_todos_los_estudiantes?\n0.No\n1.Si\n");
115  ____scanf("%d",&bool_todos);
            ____todos=bool_todos;
117  ____if((todos==false)){
                ____//con el numero de estudiante que se ingresa se modifica el valor del arreglo al que
                apunta el puntero
119
                ____do{
121  ____printf("\nIngrese_el_numero_del_estudiante_que_desea_corregir\n");
                    ____scanf("%d",&estudiante);
123  ____ptr_struct=&dato[estudiante-1];
                    ____selector=estudiante-1;
125  ____if(estudiante>cantidad){
                        ____printf("\nEl_estudiante_%d_no_existe,intente_nuevamente!!!\n\n",estudiante);
127  ____} else {
                            ____set_struct(ptr_struct, selector);

```

```

129 _____}
130 _____}while(estudiante>cantidad);
131 _____} //fin if
132 _____else{
133 _____ptr_struct=&dato[0];
134 _____for(selector=0;selector<cantidad;selector++){
135 _____set_struct(ptr_struct,selector);
136 _____//al aumentar el valor del puntero se cambia el espacio del arreglo
137 _____ptr_struct++;
138 _____} //fin for
139 _____}
140 _____break;
141
142 _____case true:
143 _____printf("\nDesea_imprimir_los_datos_de_todos_los_estudiantes?\n0.No\n1.Si\n");
144 _____scanf("%d",&bool_todos);
145 _____todos=bool_todos;
146 _____if((todos==false)){
147
148 _____do{
149 _____printf("\nIngrese_el_numero_del_estudiante_que_desea_imprimir\n");
150 _____scanf("%d",&estudiante);
151 _____ptr_struct=&dato[estudiante-1];
152 _____selector=estudiante-1;
153 _____if(estudiante>cantidad){
154 _____printf("\nEl_estudiante_%d_no_existe,intente_nuevamente!!!\n\n",estudiante);
155 _____} else{
156 _____print_struct(ptr_struct,selector);
157 _____}
158 _____}while(estudiante>cantidad);
159 _____} //fin if
160 _____else{
161 _____ptr_struct=&dato[0];
162 _____for(selector=0;selector<cantidad;selector++){
163 _____print_struct(ptr_struct,selector);
164 _____ptr_struct++;
165 _____} //fin for
166 _____} //fin else
167 _____break;
168 _____} //fin swtich
169 _____printf("\nDesea_terminar?\n0.No\n1.Si\n");
170 _____scanf("%d",&bool_terminar);
171 _____terminar=bool_terminar;
172
173 _____}while(terminar==false);
174 _____}
175
176 _____//en el main se llaman las funciones para su ejecucion
177 int main(){
178 _____//decloracion de variables*/
179 _____//decloracion de variables*/
180 _____//decloracion de variables*/
181 _____int cantidad=0,bool_datos=0,bool_todos=0,bool_terminar=0,estudiante=0,selector=0;
182 ______Bool datos_ok=false,todos=false,terminar=false;
183 _____int *ptr_datos=NULL;
184 _____//el puntero tipo struct datos_persona para poder moverme entre las "casillas" del struct
185 _____struct datos_persona *ptr_struct=NULL;
186 _____ptr_datos=&cantidad;
187
188 ______num_datos(ptr_datos);
189 _____//se crea la variable dato tipo array para hacer un arreglo de structs
190 _____struct datos_persona dato[cantidad];
191
192 _____logica(selector, ptr_struct, datos_ok, terminar,todos,bool_datos, bool_todos, bool_terminar,
193 _____estudiante, cantidad, dato);
194 _____return 0;
195 _____} //fin del main

```

4. Conclusiones

Se concluye que este laboratorio es muy útil para practicar código en C, al final en la parte 2, o sea el laboratorio 5 se decidió agregar al código mas funciones para practicar aún más por lo que agregué situaciones de fallo para que el que ingresa datos pueda continuar solo si los ingresa correctamente, también se puede ver que los structs y enums sirven para ordenar el código y hacerlo más facil de leer y de modificar, también para que sea menos enredado a la hora de leerlo y por último que los punteros son una herramienta muy importante para C y hay que dominarlos porque pueden ser muy poderosos si se saben utilizar.

Referencias

- [1] Structures in C - GeeksforGeeks"GeeksforGeeks, 2019. [Online]. Available: <https://www.geeksforgeeks.org/structures-c/>. [Accessed: 27- May- 2019].
- [2] D. Villa, "3.8.3. Programas mas claros gracias a enum", Arco.esi.uclm.es, 2019. [Online]. Available: [http://arco.esi.uclm.es/david.villa/pensar en C++/vol1/ch03s08s03.html](http://arco.esi.uclm.es/david.villa/pensar%20en%20C%2B%2B/vol1/ch03s08s03.html). [Accessed: 27- May- 2019].