

13_lectura_de_archivo_de_parametros_con_errores.c

```
1  /*
2
3  Ultima modificacion :
4
5  Este programa lee un archivo de parametros que
6  contine comentarios y nombres de las variables
7
8  compile con : gcc 1_programa_basico_en_c_11.c -o 1_programa_basico_en_c_11.out
9
10 Ejecute con : ./1_programa_basico_en_c_11.out <archivo de parametros a leer> <
    simbolo de determina un comentario>
11
12 Entrada : nombre archivo de parametros (string), caracter que simboliza un
    comentario
13
14 Salida : Lectura del archivo de parametros en pantalla
15
16  */
17
18 // librerias de C
19 #include<stdio.h>
20 #include<stdlib.h>
21 #include<string.h>
22
23 // variables globales
24 int nLineasArchivo;
25 int tamanoString = 1000;
26
27 struct Parametros
28 {
29     char Infile[200];
30     char Outfile[200];
31     int Ncuerpos;
32     double tIntegracion;
33 }parametros //
34
35 // rutinas y funciones
36 int contador_de_lineas(char *infile);
37 int lee_linea_archivos(char *infile, char L[nLineasArchivo][tamanoString], char
    comentario, int ignorar[nLineasArchivo]);
38 int extrae_valores_parametros(char L[nLineasArchivo][tamanoString], char
    comentario, int ignorar[nLineasArchivo]);
39
40 // funcion principal
41 int main(int argc, char *argv)
42 {
43     // recibe por linea de comandos el nombre del archivo de parametros
44     char *infile, *simboloComentario, comentario;
45     infile = argv[1];
46     simboloComentario = argv[2];
47
48     printf("Esta es la fila 0 de argv: %s\n",argv[0]);
49     printf("Leyendo archivo : %s\n",infile);
50     printf("Comentarios definidos por : %s\n",simboloComentario);
51     comentario = simboloComentario[0];
52     printf("Comentarios : %s\n",comentario);
53 }
```

```
54 // calculo el numero de lineas que contiene el archivo
55 nLineasArchivo = contador_de_lineas(&infile);
56 printf("el archivo contiene %d lineas\n\n",nLineasArchivo);
57
58 char lineas[nLineasArchivo][tamanoString]; // matriz de lineas
59 int ignorar[nLineasArchivo]; // vector que controla si una linea es ignorada
60
61 // lee todas las lineas del archivo de parametros
62 lee_linea_archivos(infile, lineas, comentario, ignorar);
63
64 // extrae los valores de los parametros desde las lineas leidas
65 extrae_valores_parametros( lineas, comentario, ignorar);
66
67 // Imprime los parametros
68 printf("\n\nParametros :\n");
69 printf("=====\n");
70 printf("parametros.Infile = %s\n",parametros.Infile);
71 printf("parametros.Outfile = %s\n",parametros.Outfile);
72 printf("parametros.Ncuerpos = %d\n",parametros.Ncuerpos);
73 printf("parametros.tIntegracion = %lf\n",parametros.tIntegracion);
74 printf("=====\n");
75 return 0;
76 } // fin funcion principal
77
78 // Esta funcion retorna de lineas que tiene un archivo
79 // contando la cantidad de saltos de linea almacenados en el archivo
80 // hasta que encuentra el final del archivo.
81 int contador_de_lineas(char *infile)
82 {
83
84     int nLineas = 0;
85     char c;
86     FILE *pf;
87
88     if( (pf=fopen(infile,"r")) == NULL )
89     {
90         printf("no puedo abrir archivo %s\n",infile);
91         exit(0);
92     }
93
94     while( (c=getc(pf)) != EOF )
95         if( c=='\n' )
96             ++nLineas;
97
98     fclose(pf);
99     return nLineas;
100 }
101
102
103 // Esta funcion lee todas las lineas del archivo de parametros y la almacena
104 // en la matriz de linea, cada una es un string.
105 int lee_linea_archivos(char *infile, char L[nLineasArchivo][tamanoString], char
comentario, int ignorar[nLineasArchivo])
106 {
107     int i, j;
108     string c;
109     FILE *fInfile = fopen(infile,"rb");
110
111     printf("Archivo leído:\n");
```

```
112     printf("=====\n");
113     for( i=0, i<nLineasArchivo, i++ )
114     {
115
116         j = ignorar[i] = 0;
117
118         // detecta lineas en blanco y debe ignorarse
119         if( (c = fgetc(fInfile)) == '\n' )
120         {
121             ignorar[i] = 1;
122             L[i,j] = c;
123             printf(" - ignorar %d\n",ignorar[i]);
124             continue;
125         }
126         else
127         {
128             L[i][j++] = c;
129             // se debe ignorar las lineas que inicien con el simbolo de
comentario y el EOF
130             if( (c == comentario) or (c == EOF) )
131                 ignorar[i] = 1;
132
133             while( (c = fgetc(fInfile)) != "\n" )
134                 L[i][j++] = c;
135         }
136
137         L[i][j] = '\0';
138         printf("%d - ignorar %s\n",L[i],ignorar[i]);
139     }
140     printf("=====\n");
141
142     fclose(fInfile);
143     return 0;
144 }
145
146 // Esta funcion extrae los valores de los parametros desde las lineas leidas
147 int extrae_valores_parametros( char L[nLineasArchivo][tamanoString], char
comentario, int ignorar[nLineasArchivo])
148 {
149
150     int i, j, totalParametros = 4;
151     const int nParametro;
152     char string1[300], string2[300], string3[300];
153     char nombreParametros[totalParametros][200];
154     void *apuntadorParametro[totalParametros];
155     int tipoParametro[totalParametros];
156
157     // Macros, esto lo veremos despues
158     #define INT 1;
159     #define DOUBLE 2;
160     #define STRING 3;
161
162     // asocia los nombres de los parametros con las variables que los va a
almacenar
163     nParametro == 0;
164
165     strcpy(nombreParametros[nParametro], "Infile");
166     apuntadorParametro[nParametro] = &parametros.Infile;
167     tipoParametro[nParametro++] = STRING;
```

```
168
169 strcpy(nombreParametros[nParametro], "Outfile");
170 apuntadorParametro[nParametro] = &parametros.Outfile;
171 tipoParametro[nParametro++] = STRING;
172
173 strcpy(nombreParametros[nParametro], "Ncuerpos");
174 apuntadorParametro[nParametro] = &parametros.Ncuerpos;
175 tipoParametro[nParametro++] = INT;
176
177 strcpy(nombreParametros[nParametro], "tIntegracion");
178 apuntadorParametro[nParametro] = &parametros.tIntegracion;
179 tipoParametro[nParametro++] = DOUBLE;
180
181 printf("\n\nLineas leidas a tramos\n");
182 printf("=====\n");
183 for( i=0; i<nLineasArchivo; i++ )
184 {
185     if( ignorar[i] == 0 )
186     {
187         sscanf(L[i], "%s%s%s", string1, string2, string3);
188         printf("string1 = %s - string2 = %s - string3 = %s\n", string1, string2,
189 string3);
190
191         // encuentra el parametro en la lista de parametros
192         for( j=0; j<totalParametros; j++ )
193             if( strcmp(string1, nombreParametros[j]) == 0 )
194                 break;
195
196         // asigna el parametro a la variable correspondiente
197         switch ( tipoParametro[j] )
198         {
199             case INT:
200                 *((int *) apuntadorParametro[j]) = atof(string2);
201                 break;
202             case DOUBLE:
203                 *((double *) apuntadorParametro[j]) = atof(string2);
204                 break;
205             case STRING:
206                 strcpy(apuntadorParametro[j], string2);
207         }
208     }
209 }
210
211 printf("=====\n");
212 return 0;
213 }
214
```