



Curso: Sistemas Operativos

Profesora: Mireya Paredes

Reporte 1: Error Printing

Miembros del equipo:

Rafael Agustín López Hernández – 160805

Alexander Díaz Ruiz - 160046

Jose Ashamat Jaimes Saavedra- 158320

# Práctica 1

## Introducción

El objetivo de esta práctica es familiarizarse con la arquitectura del sistema operativo UNIX y específicamente con el manejo de errores de Linux. Además, se aprenderá a cómo programar y compilar desde la terminal de Linux utilizando el compilador GCC.

UNIX es un *kernel* de un sistema operativo de tiempo compartido, el cual fue desarrollado por Dennis Ritchie en la década de los 60s. Un *kernel* es un programa que siempre está residente en memoria; por estar en memoria principal todo el tiempo, éste tiene que ser pequeño, usando lo mínimo de almacenamiento sin perder la capacidad para dar los servicios requeridos por otras aplicaciones. Los servicios que corre son para el manejo de memoria, manejo de tareas y manejo de disco.(Beal, n.d.)

Linux, un sistema operativo *open source*, se utilizará para esta actividad, enfocando el uso en el uso del lenguaje C y correrlo en el programa *shell*. *Shell* es un programa que toma comandos del teclado y da al sistema operativo para realizarlos. (*Learning the shell—Lesson 1: What is the shell?*, n.d.) En la mayoría de las distribuciones el shell que se usa es BASH (Bourne Again Shell), pero en este caso se usaron BASH y ZSH (Z Shell).

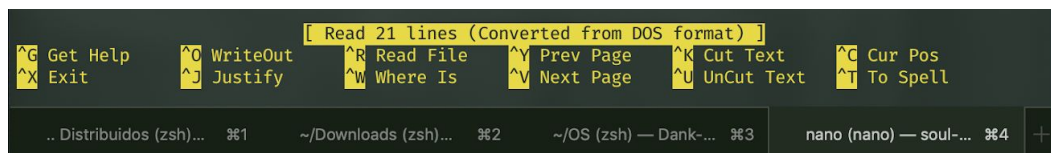
# Desarrollo

El programa se realizó en equipo, y las siguientes capturas de pantalla explican el proceso de desarrollo:



**Figura 1.** Comando para abrir un archivo .c «ej1\_1» en nano.

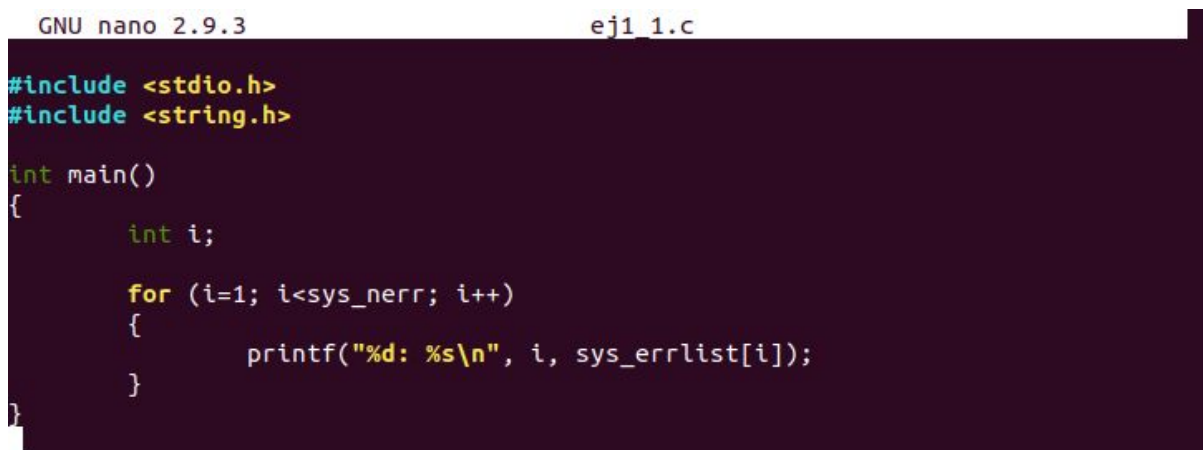
En la *Figura 1* se aprecia el comando que se usó para abrir y modificar el código. Nano es un editor de texto incluido en UNIX, el cual es muy amigable para el usuario, teniendo los comandos de guardar, editar, cortar y más en la barra de herramientas--vistas a continuación en la *Figura 2*:



**Figura 2.** Barra de herramientas del editor de texto Nano.

El primer ejercicio a resolver constó de imprimir en la terminal todos los códigos de error que maneja el sistema de UNIX, precedidos por su índice de identificación.

Francisco Manuel Márquez García, en su libro *Unix-Programación Avanzada* (Márquez García, 2004), sugiere imprimir cada elemento de `sys_errlist[ ]`, arreglo de strings, el cual contiene todas las descripciones de los códigos de error, dentro de un ciclo `for` con límite superior de `sys_nerr` (el número de descripciones contenidas por `sys_errlist[ ]`), como se aprecia en el código de la siguiente *Figura 3*:



**Figura 3.** Solución para el primer ejercicio ej1\_1.c , tal como ilustrado en el ejemplo provisto por Márquez García.

Sin embargo, al tratar de compilar el programa con GCC, la siguiente salida fue producida por Bash:

```
alex@alex-VirtualBox:~/Documents$ nano ej1_1.c
alex@alex-VirtualBox:~/Documents$ gcc ej1_1.c -o programa
/tmp/ccS3N70T.o: In function `main':
ej1_1.c:(.text+0x21): warning: `sys_errlist' is deprecated; use `strerror' or `strerror_r' instead
ej1_1.c:(.text+0x45): warning: `sys_nerr' is deprecated; use `strerror' or `strerror_r' instead
alex@alex-VirtualBox:~/Documents$
```

*Figura 4. Error al tratar de compilar el código presente en la Figura 3.*

, haciendo referencia a la naturaleza obsoleta del ejemplo provisto en el libro.

En lugar de emplear `sys_errlist[ ]`, éste terminó siendo reemplazado por la función **`strerror(int)`**, misma que fue sugerida por GCC; esta última función siendo capaz de recuperar un string referente al mensaje de error correspondiente al índice ingresado como parámetro de la misma.

Sustituir `sys_nerr` fue más sencillo aún, debido a que representa una constante: el número de errores manejados por UNIX.

Es por ello que se sustituyó por 134--mismo número de errores reconocidos por el *kernel*.

Ambos cambios al programa mencionados con anterioridad son reflejados a continuación, en la *Figura 5*:

```
GNU nano 2.9.3 ej1_1.c

/* Se importa el header <string.h> para poder implementar la función strerror
Ésta regresa un apuntador a un string referente al mensaje de error cuyo
identificador se ingresa como argumento de la función.
*/
#include <stdio.h>
#include <string.h>

int main()
{
    int i;

    /* Nótese que se optó por ocupar strerror() en lugar de sys_nerr y
    sys_errlist[], dado que el mismo compilador GCC los considera obso-
    letos.
    */

    for (i=1; i<134; i++)
    {
        printf("%d: %s\n", i, strerror(i));
    }
}
```

*Figura 5. ej1\_1.c corregido.*

La ejecución del programa se reserva para la última sección del reporte: Resultados del programa.

```
GNU nano 2.0.6 File: libmic.a

extern int errno ;

void error(char *nfichero, int nro_linea, char *mensaje){
char *errores []={"", "EPERM", "ENOENT", "ESRCH", "EINTR", "EIO", "ENXIO", "E2BIG", "ENOEXEC", "EBADF", "ECHILD", "EAGAIN",
"ENOMEN", "EACCES", "EFAULT", "ENOTBLINK", "EBUSY", "EEXIST", "EXDEV", "NODEV", "NOTDIR", "EISDIR", "EINVAL", "ENFILE", "EMFILE",
"NOTTY", "ETXTBSY", "EFBIG", "ENOSPC", "ESPIPE", "EROFS", "EMLINK", "EPIPE", "EDOM", "ERANGE", "EDEADLK", "ENAMETOOLONG",
"ENOLCK", "ENOSYS", "ENOTEMPTY", "ELOOP", "EWOULDBLOCK", "ENOMSG", "EIDRM", "ECHRNG", "EL2NSYNC", "EL3HLT", "EL3RST",
"ELNRNG", "EUNATCH", "ENOCSI", "EL2HLT", "EBADE", "EBADR", "EXFULL", "ENOANO", "EBADRQC", "EBADSLT", "EDEADLOCK", "EBFONT",
"ENOSTR", "ENODATA", "ETIME", "ENOSR", "ENONET", "ENOPKG", "EREMOTE", "ENOLINK", "EADV", "ESRMNT", "ECOMM", "EPROTO",
"EMULTIHOP", "EDOTDOT", "EBADMSG", "EOVERFLOW", "ENOTUNIQ", "EBADFD", "EREMCHG", "ELIBACC", "ELIBBAD", "ELIBSCN", "ELIBMAX",
"ELIBEXEC", "EILSEQ", "ERESTART", "ESTRPIPE", "EUSERS", "ENOTSOCK", "EDESTADDRREQ", "EMSGSIZE", "EPROTOTYPE", "ENOPROTOOPT",
"PROTONOSUPPORT", "ESOCKTNOSUPPORT", "EOPNOTSUPP", "EPFNOSUPPORT", "EAFNOSUPPORT", "EADDRINUSE", "EADDRNOTAVAIL", "ENETDOWN",
"ENETUNREACH", "ENETRESET", "ECONNABORTED", "ECONNRESET", "ENOBUFS", "EISCONN", "ENOTCONN", "ESHUTDOWN", "ETOOMANYREFS",
"ETIMEDOUT", "ECONNREFUSED", "EHOSTDOWN", "EHOSTUNREACH", "EALREADY", "EINPROGRESS", "ESTALE", "EUCLEAN", "ENOTNAM", "ENAVAIL",
"EISNAM", "EREMOTEIO", "EDQUOT", "ENOMEDIUM", "EMEDIUMTYPE", "ECANCELED", "ENOKEY", "EKEYEXPIRED", "EKEYREVOKED",
"EKEYREJECTED", "OWNERDEAD", "NOTRECOVERABLE", "EHWPOISON"};

printf("%s(%d). %s %s, %s %s %s\n", nfichero, nro_linea, "ERROR:", errores[errno], strerror(errno), mensaje);
}
```

**Figura 6.** El archivo *libmic.a* abierto en nano.

```
GNU nano 2.0.6 File: mic.h

#include <stdio.h>
#include<string.h>
#include<errno.h>
void error(char *nfichero, int nro_linea, char *mensaje);
#include "libmic.a"
```

**Figura 7.** El archivo *mic.h* abierto en nano.

```
GNU nano 4.5 ejercicioProgramacion1.c
#include "mic.h"

int main ()
{
    char mensaje[50];
    int i;
    for(i = 1; i<133; i++){
        sprintf(mensaje, "%s %d", "Error nro.", i);
        errno = i;
        error(__FILE__, __LINE__, mensaje);
    }

    return 0;
}
```

Figuras 9-13. Los 133 códigos de error que manejan las llamadas a UNIX

**Figura 8.** Archivo ej1\_2.c abierto en nano

```
rafaellopez@soul-release ~/Documents/OS gcc ej1_1.c -o ej10ou.out
rafaellopez@soul-release ~/Documents/OS ./ej10ou.out
```

**Figura 9.** Se compila el archivo ej1\_1.c con GCC, y creando un archivo de salida (out) usando el -o y luego poniendo el nombre del archivo de salida. Al final, el programa se corre con el comando ./[nombre del out file]



# Resultados del programa (BASH & ZSH)

La ejecución exitosa del primer ejercicio se presenta a continuación:

```
alex@alex-VirtualBox:~$ cd Documents
alex@alex-VirtualBox:~/Documents$ nano ej1_1.c
alex@alex-VirtualBox:~/Documents$ gcc ej1_1.c -o programa
alex@alex-VirtualBox:~/Documents$ ./programa
```

```
1: Operation not permitted
2: No such file or directory
3: No such process
4: Interrupted system call
5: Input/output error
6: No such device or address
7: Argument list too long
8: Exec format error
9: Bad file descriptor
10: No child processes
11: Resource temporarily unavailable
12: Cannot allocate memory
13: Permission denied
14: Bad address
15: Block device required
16: Device or resource busy
17: File exists
18: Invalid cross-device link
19: No such device
20: Not a directory
21: Is a directory
22: Invalid argument
23: Too many open files in system
24: Too many open files
25: Inappropriate ioctl for device
```

```
25: Inappropriate ioctl for device
26: Text file busy
27: File too large
28: No space left on device
29: Illegal seek
30: Read-only file system
31: Too many links
32: Broken pipe
33: Numerical argument out of domain
34: Numerical result out of range
35: Resource deadlock avoided
36: File name too long
37: No locks available
38: Function not implemented
39: Directory not empty
40: Too many levels of symbolic links
41: Unknown error 41
42: No message of desired type
43: Identifier removed
44: Channel number out of range
45: Level 2 not synchronized
46: Level 3 halted
47: Level 3 reset
48: Link number out of range
49: Protocol driver not attached
50: No CSI structure available
51: Level 2 halted
52: Invalid exchange
53: Invalid request descriptor
```

```
53: Invalid request descriptor
54: Exchange full
55: No anode
56: Invalid request code
57: Invalid slot
58: Unknown error 58
59: Bad font file format
60: Device not a stream
61: No data available
62: Timer expired
63: Out of streams resources
64: Machine is not on the network
65: Package not installed
66: Object is remote
67: Link has been severed
68: Advertise error
69: Srmount error
70: Communication error on send
71: Protocol error
72: Multihop attempted
73: RFS specific error
74: Bad message
75: Value too large for defined data type
76: Name not unique on network
77: File descriptor in bad state
78: Remote address changed
79: Can not access a needed shared library
80: Accessing a corrupted shared library
81: .lib section in a.out corrupted
```

```
81: .lib section in a.out corrupted
82: Attempting to link in too many shared libraries
83: Cannot exec a shared library directly
84: Invalid or incomplete multibyte or wide character
85: Interrupted system call should be restarted
86: Streams pipe error
87: Too many users
88: Socket operation on non-socket
89: Destination address required
90: Message too long
91: Protocol wrong type for socket
92: Protocol not available
93: Protocol not supported
94: Socket type not supported
95: Operation not supported
96: Protocol family not supported
97: Address family not supported by protocol
98: Address already in use
99: Cannot assign requested address
100: Network is down
101: Network is unreachable
102: Network dropped connection on reset
103: Software caused connection abort
104: Connection reset by peer
105: No buffer space available
106: Transport endpoint is already connected
107: Transport endpoint is not connected
108: Cannot send after transport endpoint shutdown
109: Too many references: cannot splice
```



```
109: Too many references: cannot splice
110: Connection timed out
111: Connection refused
112: Host is down
113: No route to host
114: Operation already in progress
115: Operation now in progress
116: Stale file handle
117: Structure needs cleaning
118: Not a XENIX named type file
119: No XENIX semaphores available
120: Is a named type file
121: Remote I/O error
122: Disk quota exceeded
123: No medium found
124: Wrong medium type
125: Operation canceled
126: Required key not available
127: Key has expired
128: Key has been revoked
129: Key was rejected by service
130: Owner died
131: State not recoverable
132: Operation not possible due to RF-kill
133: Memory page has hardware error
alex@alex-VirtualBox:~/Documents$
```

**Figuras 10-14.** Los 133 códigos de error que manejan las llamadas a UNIX

```
rafaellopez@soul-release ~/Documents/OS ./ej10ou.out
ej1_1.c(16). ERROR: EPERM, Operation not permitted Error # 1
ej1_1.c(16). ERROR: ENOENT, No such file or directory Error # 2
ej1_1.c(16). ERROR: ESRCH, No such process Error # 3
ej1_1.c(16). ERROR: EINTR, Interrupted system call Error # 4
ej1_1.c(16). ERROR: EIO, Input/output error Error # 5
ej1_1.c(16). ERROR: ENXIO, Device not configured Error # 6
ej1_1.c(16). ERROR: E2BIG, Argument list too long Error # 7
ej1_1.c(16). ERROR: ENOEXEC, Exec format error Error # 8
ej1_1.c(16). ERROR: EBADF, Bad file descriptor Error # 9
ej1_1.c(16). ERROR: ECHILD, No child processes Error # 10
ej1_1.c(16). ERROR: EAGAIN, Resource deadlock avoided Error # 11
ej1_1.c(16). ERROR: ENOMEM, Cannot allocate memory Error # 12
ej1_1.c(16). ERROR: EACCES, Permission denied Error # 13
ej1_1.c(16). ERROR: EFAULT, Bad address Error # 14
ej1_1.c(16). ERROR: ENOTBLK, Block device required Error # 15
ej1_1.c(16). ERROR: EBUSY, Resource busy Error # 16
ej1_1.c(16). ERROR: EEXIST, File exists Error # 17
ej1_1.c(16). ERROR: EXDEV, Cross-device link Error # 18
ej1_1.c(16). ERROR: ENODEV, Operation not supported by device Error # 19
ej1_1.c(16). ERROR: ENOTDIR, Not a directory Error # 20
ej1_1.c(16). ERROR: EISDIR, Is a directory Error # 21
ej1_1.c(16). ERROR: EINVAL, Invalid argument Error # 22
ej1_1.c(16). ERROR: ENFILE, Too many open files in system Error # 23
ej1_1.c(16). ERROR: EMFILE, Too many open files Error # 24
ej1_1.c(16). ERROR: ENOTTY, Inappropriate ioctl for device Error # 25
ej1_1.c(16). ERROR: ETXTBSY, Text file busy Error # 26
ej1_1.c(16). ERROR: EFBIG, File too large Error # 27
ej1_1.c(16). ERROR: ENOSPC, No space left on device Error # 28
ej1_1.c(16). ERROR: ESPIPE, Illegal seek Error # 29
ej1_1.c(16). ERROR: EROFS, Read-only file system Error # 30
ej1_1.c(16). ERROR: EMLINK, Too many links Error # 31
ej1_1.c(16). ERROR: EPIPE, Broken pipe Error # 32
ej1_1.c(16). ERROR: EDOM, Numerical argument out of domain Error # 33
ej1_1.c(16). ERROR: ERANGE, Result too large Error # 34
ej1_1.c(16). ERROR: EDEADLK, Resource temporarily unavailable Error # 35
ej1_1.c(16). ERROR: ENAMETOOLONG, Operation now in progress Error # 36
ej1_1.c(16). ERROR: ENOLCK, Operation already in progress Error # 37
ej1_1.c(16). ERROR: ENOSYS, Socket operation on non-socket Error # 38
ej1_1.c(16). ERROR: ENOTEMPTY, Destination address required Error # 39
ej1_1.c(16). ERROR: ELOOP, Message too long Error # 40
ej1_1.c(16). ERROR: EWOULDBLOCK, Protocol wrong type for socket Error # 41
ej1_1.c(16). ERROR: ENOMSG, Protocol not available Error # 42
ej1_1.c(16). ERROR: EIDRM, Protocol not supported Error # 43
ej1_1.c(16). ERROR: ECHRNG, Socket type not supported Error # 44
ej1_1.c(16). ERROR: EL2NSYNC, Operation not supported Error # 45
ej1_1.c(16). ERROR: EL3HLT, Protocol family not supported Error # 46
ej1_1.c(16). ERROR: EL3RST, Address family not supported by protocol family Error # 47
ej1_1.c(16). ERROR: ELNRNG, Address already in use Error # 48
ej1_1.c(16). ERROR: EUNATCH, Can't assign requested address Error # 49
ej1_1.c(16). ERROR: ENOCSI, Network is down Error # 50
ej1_1.c(16). ERROR: EL2HLT, Network is unreachable Error # 51
ej1_1.c(16). ERROR: EBADE, Network dropped connection on reset Error # 52
ej1_1.c(16). ERROR: EBADR, Software caused connection abort Error # 53
ej1_1.c(16). ERROR: EXFULL, Connection reset by peer Error # 54
```

**Figura 15.** Errores desde el #1 al #54



```
ej1_1.c(16). ERROR: EXFULL, Connection reset by peer Error # 54
ej1_1.c(16). ERROR: ENOANO, No buffer space available Error # 55
ej1_1.c(16). ERROR: EBADRQC, Socket is already connected Error # 56
ej1_1.c(16). ERROR: EBADSLT, Socket is not connected Error # 57
ej1_1.c(16). ERROR: EDEADLOCK, Can't send after socket shutdown Error # 58
ej1_1.c(16). ERROR: EBFONT, Too many references: can't splice Error # 59
ej1_1.c(16). ERROR: ENOSTR, Operation timed out Error # 60
ej1_1.c(16). ERROR: ENODATA, Connection refused Error # 61
ej1_1.c(16). ERROR: ETIME, Too many levels of symbolic links Error # 62
ej1_1.c(16). ERROR: ENOSR, File name too long Error # 63
ej1_1.c(16). ERROR: ENONET, Host is down Error # 64
ej1_1.c(16). ERROR: ENOPKG, No route to host Error # 65
ej1_1.c(16). ERROR: EREMOTE, Directory not empty Error # 66
ej1_1.c(16). ERROR: ENOLINK, Too many processes Error # 67
ej1_1.c(16). ERROR: EADV, Too many users Error # 68
ej1_1.c(16). ERROR: ESRMNT, Disc quota exceeded Error # 69
ej1_1.c(16). ERROR: ECOMM, Stale NFS file handle Error # 70
ej1_1.c(16). ERROR: EPROTO, Too many levels of remote in path Error # 71
ej1_1.c(16). ERROR: EMULTIHOP, RPC struct is bad Error # 72
ej1_1.c(16). ERROR: EDOTDOT, RPC version wrong Error # 73
ej1_1.c(16). ERROR: EBADMSG, RPC prog. not avail Error # 74
ej1_1.c(16). ERROR: EOVERFLOW, Program version wrong Error # 75
ej1_1.c(16). ERROR: ENOTUNIQ, Bad procedure for program Error # 76
ej1_1.c(16). ERROR: EBADF, No locks available Error # 77
ej1_1.c(16). ERROR: EREMCHG, Function not implemented Error # 78
ej1_1.c(16). ERROR: ELIBACC, Inappropriate file type or format Error # 79
ej1_1.c(16). ERROR: ELIBBAD, Authentication error Error # 80
ej1_1.c(16). ERROR: ELIBSCN, Need authenticator Error # 81
ej1_1.c(16). ERROR: ELIBMAX, Device power is off Error # 82
ej1_1.c(16). ERROR: ELIBEXEC, Device error Error # 83
ej1_1.c(16). ERROR: EILSEQ, Value too large to be stored in data type Error # 84
ej1_1.c(16). ERROR: ERESTART, Bad executable (or shared library) Error # 85
ej1_1.c(16). ERROR: ESTRPIPE, Bad CPU type in executable Error # 86
ej1_1.c(16). ERROR: EUSERS, Shared library version mismatch Error # 87
ej1_1.c(16). ERROR: ENOTSOCK, Malformed Mach-o file Error # 88
ej1_1.c(16). ERROR: EDESTADDRREQ, Operation canceled Error # 89
ej1_1.c(16). ERROR: EMSGSIZE, Identifier removed Error # 90
ej1_1.c(16). ERROR: EPROTOTYPE, No message of desired type Error # 91
ej1_1.c(16). ERROR: ENOPROTOOPT, Illegal byte sequence Error # 92
ej1_1.c(16). ERROR: EPROTONOSUPPORT, Attribute not found Error # 93
ej1_1.c(16). ERROR: ESOCKTNOSUPPORT, Bad message Error # 94
ej1_1.c(16). ERROR: EOPNOTSUPP, EMULTIHOP (Reserved) Error # 95
ej1_1.c(16). ERROR: EPNOSUPPORT, No message available on STREAM Error # 96
ej1_1.c(16). ERROR: EAFNOSUPPORT, ENOLINK (Reserved) Error # 97
ej1_1.c(16). ERROR: EADDRINUSE, No STREAM resources Error # 98
ej1_1.c(16). ERROR: EADDRNOTAVAIL, Not a STREAM Error # 99
ej1_1.c(16). ERROR: ENETDOWN, Protocol error Error # 100
ej1_1.c(16). ERROR: ENETUNREACH, STREAM ioctl timeout Error # 101
ej1_1.c(16). ERROR: ENETRESET, Operation not supported on socket Error # 102
ej1_1.c(16). ERROR: ECONNABORTED, Policy not found Error # 103
ej1_1.c(16). ERROR: ECONNRESET, State not recoverable Error # 104
ej1_1.c(16). ERROR: ENOBUFS, Previous owner died Error # 105
ej1_1.c(16). ERROR: EISCONN, Interface output queue is full Error # 106
ej1_1.c(16). ERROR: ENOTCONN, Unknown error: 107 Error # 107
ej1_1.c(16). ERROR: ESHUTDOWN, Unknown error: 108 Error # 108
```

**Figura 16.** Errores desde #54 hasta #108

```
ej1_1.c(16). ERROR: ESHUTDOWN, Unknown error: 108 Error # 108
ej1_1.c(16). ERROR: ETOOMANYREFS, Unknown error: 109 Error # 109
ej1_1.c(16). ERROR: ETIMEDOUT, Unknown error: 110 Error # 110
ej1_1.c(16). ERROR: ECONNREFUSED, Unknown error: 111 Error # 111
ej1_1.c(16). ERROR: EHOSTDOWN, Unknown error: 112 Error # 112
ej1_1.c(16). ERROR: EHOSTUNREACH, Unknown error: 113 Error # 113
ej1_1.c(16). ERROR: EALREADY, Unknown error: 114 Error # 114
ej1_1.c(16). ERROR: EINPROGRESS, Unknown error: 115 Error # 115
ej1_1.c(16). ERROR: ESTALE, Unknown error: 116 Error # 116
ej1_1.c(16). ERROR: EUCLEAN, Unknown error: 117 Error # 117
ej1_1.c(16). ERROR: ENOTNAM, Unknown error: 118 Error # 118
ej1_1.c(16). ERROR: ENAVAIL, Unknown error: 119 Error # 119
ej1_1.c(16). ERROR: EISNAM, Unknown error: 120 Error # 120
ej1_1.c(16). ERROR: EREMOTEIO, Unknown error: 121 Error # 121
ej1_1.c(16). ERROR: EDQUOUT, Unknown error: 122 Error # 122
ej1_1.c(16). ERROR: ENOMEDIUM, Unknown error: 123 Error # 123
ej1_1.c(16). ERROR: EMEDIUMTYPE, Unknown error: 124 Error # 124
ej1_1.c(16). ERROR: ECANCELED, Unknown error: 125 Error # 125
ej1_1.c(16). ERROR: ENOKEY, Unknown error: 126 Error # 126
ej1_1.c(16). ERROR: EKEYEXPIRED, Unknown error: 127 Error # 127
ej1_1.c(16). ERROR: EKEYREVOKED, Unknown error: 128 Error # 128
ej1_1.c(16). ERROR: EKEYREJECTED, Unknown error: 129 Error # 129
ej1_1.c(16). ERROR: EOWNERDEAD, Unknown error: 130 Error # 130
ej1_1.c(16). ERROR: ENOTRECOVERABLE, Unknown error: 131 Error # 131
ej1_1.c(16). ERROR: EHWPOISON, Unknown error: 132 Error # 132
```

**Figura 17.** Errores desde #108 hasta #132

# Bibliografía

Beal, V. (n.d.). *What is Kernel (Operating System Kernel)? Webopedia Definition*. Retrieved

February 4, 2020, from <https://www.webopedia.com/TERM/K/kernel.html>

*Learning the shell—Lesson 1: What is the shell?* (n.d.). Retrieved February 6, 2020, from

[http://linuxcommand.org/lc3\\_lts0010.php](http://linuxcommand.org/lc3_lts0010.php)

Márquez García, F. M. (2004). Capítulo 1 Introducción. In *UNIX: PROGRAMACIÓN*

*AVANZADA* (3ra edición). Alfaomega Grupo Editor.