

Listar los archivos dentro de un directorio

Escrito por mcggiber - 06/12/2014 17:12

Se presenta un problema al momento de enlistar los archivos en el directorio, se agradecen los aportes.

```
#include
#include
#include
#include
#include
#include
#include

/* Función para devolver un error en caso de que ocurra */
void error(const char *s);

/* Calculamos el tamaño del archivo */
long fileSize(char *fname);

/* Sacamos el tipo de archivo haciendo un stat(), es como el stat de la línea de comandos */
unsigned char statFileType(char *fname);

/* Función que hace algo con un archivo, pero le pasamos el dirent completo, usaremos más datos */
void procesoArchivo(char *ruta, struct dirent *ent);

int main(int argc, char *argv)
{
    /* Con un puntero a DIR abriremos el directorio */
    DIR *dir;
    /* en *ent habrá información sobre el archivo que se está "sacando" a cada momento */
    struct dirent *ent;

    if (argc != 2)
    {
        error("Uso: ./directorio_2 n");
    }
    /* Empezaremos a leer en el directorio actual */
    dir = opendir (argv);

    /* Miramos que no haya error */
    if (dir == NULL)
        error("No puedo abrir el directorio");

    /* Una vez nos aseguramos de que no hay error, ¡vamos a jugar! */
    /* Leyendo uno a uno todos los archivos que hay */
    while ((ent = readdir (dir)) != NULL)
    {
        /* Nos devolverá el directorio actual (.) y el anterior (..), como hace ls */
        if ( ( strcmp(ent->d_name, ".")!=0) && ( strcmp(ent->d_name, "..")!=0) )
        {
```

```
    /* Una vez tenemos el archivo, lo pasamos a una función para procesarlo. */
    procesoArchivo(argv, ent);
}
}
closedir (dir);

return EXIT_SUCCESS;
}

void error(const char *s)
{
    /* perror() devuelve la cadena S y el error (en cadena de caracteres) que tenga errno */
    perror (s);
    exit(EXIT_FAILURE);
}

long fileSize(char *fname)
{
    FILE *fich;
    long ftam=-1;

    fich=fopen(fname, "r");
    if (fich)
    {
        fseek(fich, 0L, SEEK_END);
        ftam=ftell(fich);
        fclose(fich);
    }
    else
        printf("ERRNO: %d - %sn", errno, strerror(errno));
    return ftam;
}

void procesoArchivo(char *ruta, struct dirent *ent)
{
    long ftam;
    char *nombrecompleto;
    char strtam;
    char strtipo="";
    /* Tiene que ser del mismo tipo de dirent.d_type en nuestro sistema */
    static unsigned char tipoID={DT_BLK, DT_CHR, DT_DIR, DT_FIFO, DT_LNK, DT_REG, DT_SOCK};
    static char* tipoSTRs={"Dispositivo de bloques", "Dispositivo de caracteres", "Directorio", "FIFO",
"Enlace", "Archivo regular", "Socket Unix"};

    int i;
    int tmp;
    unsigned char tipo;

    /* Sacamos el nombre completo con la ruta del archivo */
    tmp=strlen(ruta);
    nombrecompleto=malloc(tmp+strlen(ent->d_name)+2); /* Sumamos 2, por el \ y la barra de directorios
(/) no sabemos si falta */
}
```

```
if (ruta=='/')
    sprintf(nombrecompleto,"%s%s", ruta, ent->d_name);
else
    sprintf(nombrecompleto,"%s/%s", ruta, ent->d_name);

/* Calcula el tamaño */
ftam=fileSize(nombrecompleto);
if (ftam>=0)
    sprintf(strtam, "%ld bytes", ftam);
else
    strcpy(strtam, "No info");

/* A veces ent->d_type no nos dice nada, eso depende del sistema de archivos que estemos */
/* mirando, por ejemplo ext*, brtfs, sí nos dan esta información. Por el contrario, nfs */
/* no nos la da (directamente, una vez que hacemos stat sí lo hace), y es en estos casos donde
probamos con stat() */
tipo=ent->d_type;
if (tipo==DT_UNKNOWN)
    tipo=statFileType(nombrecompleto);

if (tipo!=DT_UNKNOWN)
{
    /* Podíamos haber hecho un switch con los tipos y devolver la cadena,
    pero me da la impresión de que así es menos costoso de escribir. */
    i=0;
    while ( (i
```

Re: Listar los archivos dentro de un directorio

Escrito por adrianvaca - 10/12/2014 14:07

En el siguiente links tienes un código que te puede servir de referencia:

http://programacionenc.net/index.php?option=com_content&view=article&id=128:mostrar-los-archivos-y-directorios-contenidos-en-un-directorio-determinado&catid=31:general&Itemid=41