

ARCHIVOS Y SISTEMAS DE ARCHIVOS.

Manejo de archivos en GNU/Linux

Como se explicó en las clases teóricas, el manejo de archivos se lleva a cabo mediante llamadas POSIX de manejo de archivos.

Las funciones que invocan estas llamadas (en Freepascal) se encuentran en la unit Linux:

fdOpen	Abre/crea un descriptor de archivo.
fdRead	Lee datos desde un descriptor de archivo.
fdWrite	Escribe en un descriptor de archivo.
fdSeek	Se posiciona en un archivo.
fdClose	Cierra un descriptor de archivo.

Para ver otras ver otras funciones de manejo de archivos vea la documentación de Freepascal.

Ingrese al sistema como **usuario no privilegiado**. Cree un directorio donde almacenar los programas de este práctico dentro de su directorio home. Luego sitúese en el directorio recién creado.

Escriba el siguiente programa y nómbrelo "prog_0901.pp" y compílelo.

```
Program prog_0901;

uses Linux;

Var fd : Longint;
    datos: String;

begin
  fd := FDOpen(ParamStr(1),Open_WrOnly or Open_Creat);
  if fd > 0 then
    Writeln('Escriba lo que desea en el archivo:');
    Readln(datos);
    begin
      if (Length(datos)+1) <> (FDWrite(fd,datos,Length(datos)+1))
        then Writeln ('Error al escribir en el archivo!!!');
      FDClose(fd);
    end;
  end.
end.
```

Cree un archivo vacío. Ejecute el programa pasándole como parámetro el archivo recién creado e ingrese algo de texto (al menos 20 caracteres) para escribir dentro del archivo. Verifique que la información se grabó en el archivo.

(1) Ejecute el programa nuevamente pasándole como parámetro un archivo inexistente. Se ejecutó igual? Grabó la información dentro del archivo? Por qué?

(2) Ejecute el programa nuevamente pasándole como parámetro el primer archivo e ingrese otra información (al menos 20 caracteres) para grabar en el archivo. Verifique que la información se grabó en el archivo. Qué ocurrió con la información que habíamos grabado en la primera ocasión? Por qué?

La documentación de Freepascal pueden ser de gran ayuda para responder a las preguntas anteriores.

Escriba el siguiente programa y nómbrelo "prog_0902.pp" y compílelo.

```
Program prog_0902;

uses linux;

var fd : Longint;
    datos: String;

begin
  fd := fdOpen(ParamStr(1), Open_RdOnly);
  if fd > 0 then
    begin
      if fdRead(fd, datos, 10) < 0 then
        begin
          Writeln('Error leyendo archivo!!!');
          Halt(2);
        end;
      Writeln(datos);
    end;
  fdClose(FD);
end.
```

(3) Ejecute el programa pasándole como parámetro el archivo de texto del ejercicio anterior. Observe los resultados. Por qué no mostró la totalidad del archivo?

Manejo de archivos y directorios en GNU/Linux

En un nivel más bajo se puede acceder mediante funciones POSIX, a operaciones generales de manejo de archivos, directorios y sistemas de archivos.

Las funciones que invocan estas llamadas (en Freepascal) se encuentran en la unit Linux:

Access	Chequea los permisos de acceso a un archivo
Chown	Cambia el dueño de un archivo.
Chmod	Cambia los permisos de acceso de un archivo.
Flock	Bloquea un archivo.
FStat	Brinda información sobre un archivo.
FRename	Renombra un archivo.
LStat	Brinda información sobre un enlace (link).
Link	Crea un enlace (link).
ReadLink	Lee el contenido de un enlace simbólico.
SymLink	Crea un enlace simbólico.
UnLink	Elimina un archivo.
Utime	Cambia la fecha de un archivo.
OpenDir	Abre un directorio para lectura.
CloseDir	Cierra un directorio,
ReadDir	Lee una entrada de directorio.
TellDir	Indica la entrada de directorio actual.
SeekDir	Se ubica en una entrada determinada del directorio.
FStat	Brinda información sobre el filesystem.

Para ver otras ver otras funciones de manejo de archivos, directorios y sistemas de archivos vea la documentación de Freepascal.

Escriba el siguiente programa y nómbrelo “**prog_0903.pp**” y compílelo.

```
program prog_0903;

uses Linux;

var info : Stat;

begin
  if not FStat(ParamStr(1),info) then
    begin
      Writeln('Fallo la llamada a FStat!!!');
      halt (1);
    end;
  writeln;
  writeln ('Resultados del fstat del archivo:');
  writeln ('I-nodo                : ',info.ino);
  writeln ('Modo (Tipo y Permisos)   : ',info.mode);
  writeln ('Numero de links              : ',info.nlink);
  writeln ('User ID del dueño            : ',info.uid);
  writeln ('Group ID del dueño           : ',info.gid);
  writeln ('Tipo de dispositivo inodo    : ',info.rdev);
  writeln ('Tamaño Bytes                 : ',info.size);
  writeln ('Tamaño Bloques               : ',info.blksze);
  writeln ('Nro. de Bloques              : ',info.blocks);
  writeln ('Ultimo acceso                : ',info.atime);
  writeln ('Ultima modificacion          : ',info.mtime);
  writeln ('Ultimo cambio                : ',info.ctime);

end.
```

(4) Pruebe el programa pasándole como parámetro distintos archivos. ¿Dónde está almacenada esta información de cada archivo?

Puede obtener más información sobre el registro Stat mediante la página man del mismo Stat(2).

Escriba el siguiente programa y nómbrelo “**prog_0904.pp**” y compílelo.

```
program prog_0904;

uses Linux;

var archivo: Stat;

begin
  if LStat(ParamStr(1),archivo) then
    begin
      if S_ISLNK(archivo.mode) then
        Writeln ('El archivo es un enlace...');
      if S_ISREG(archivo.mode) then
        Writeln ('El archivo es un archivo regular...');
      if S_ISDIR(archivo.mode) then
        Writeln ('El archivo es un directorio...');
      if S_ISCHR(archivo.mode) then
        Writeln ('El archivo es un dispositivo de caract...');
      if S_ISBLK(archivo.mode) then
        Writeln ('El archivo es un dispositivo de bloques...');
      if S_ISFIFO(archivo.mode) then
        Writeln ('El archivo es una canieria (Pipe)...');
      if S_ISSOCK(archivo.mode) then
        Writeln ('El archivo es un socket...');
    end;
  end.
end.
```

(5) Pruebe el programa pasándole como parámetro archivos de distintos tipos. Dónde está almacenada esta información de cada archivo?

Escriba el siguiente programa y nómbrelo "prog_0905.pp" y compílelo.

```
program prog_0905;

uses Linux;

var directorio : PDir;
    entrada : PDirent;

begin
  directorio := OpenDir(ParamStr(1));

  repeat
    entrada := ReadDir(directorio);
    if entrada <> nil then
      begin
        Writeln ('-----');
        Writeln ('Entrada No      : ',TellDir(directorio));
        Writeln ('Nombre          : ',pchar(@entrada^.name[0]));
        Writeln ('I-nodo          : ',entrada^.ino);
        Writeln ('Offset          : ',entrada^.off);
        Writeln ('Long. Nombre    : ',entrada^.reclen);

        end;
    until entrada = nil;

    CloseDir (directorio);
  end.
```

Pruebe el programa pasándole como parámetro distintos directorios.

Escriba el siguiente programa y nómbrelo “**prog_0906.pp**” y compílelo.

```
program prog_0906;

uses Linux;

var ubicacion : String;
    fsinfo : StatFS;

begin
    ubicacion:= '.';
    if not FSStat(ubicacion,fsinfo) then
        begin
            Writeln('Fallo el FSStat. Error No : ',linuxerror);
            Halt(1);
        end;
        Writeln ('Tipo de FS           : ',fsinfo.fstype);
        Writeln ('Tamaño en bloques      : ',fsinfo.bsize);
        Writeln ('Bloques libres          : ',fsinfo.bfree);
        Writeln ('Bloques disponibles     : ',fsinfo.bavail);
        Writeln ('Archivos                 : ',fsinfo.files);
        Writeln ('Descriptores libres      : ',fsinfo.ffree);
        Writeln ('Identificacion de FS    : ',fsinfo.fsid);
        Writeln ('Long. del Nombre        : ',fsinfo.namelen);
    end.
```

(6) Sobre que filesystem nos está mostrando información. Por qué?

(7) Realice su propia versión del comando **ls** empleando las llamadas POSIX de gestión de archivos y directorio. Nómbrelo “**prog_0907**”, compílelo y pruebelo listando distintos directorios. Para mantener la sencillez, el comando deberá cumplir con las siguientes características:

- Deberá mostrar (en este orden): i-nodo, tamaño en bloques y nombre de cada archivo del directorio en cuestión. (Uno por línea)
- Deberá enumerar al final el número de archivos del directorio.

```
program mi_ls;

uses Linux;

var directorio : PDir;
    entrada : PDirent;
    archivo : Stat;
    num_archivos: LongInt;

begin
    num_archivos := 0;

    directorio := OpenDir(ParamStr(1));
    Writeln ('Listando directorio ',ParamStr(1), '...');

    repeat
        entrada := ReadDir(directorio);
        if entrada <> nil then
            begin
                num_archivos := num_archivos + 1;
                FStat(ParamStr(1)+'/'+pchar(@entrada^.name[0]),archivo);
                Write (archivo.ino,' ');
                Write (archivo.size,' ');
                Writeln (pchar(@entrada^.name[0]));
            end;
        until entrada = nil;
        Writeln (num_archivos,' elemento(s) listado(s)');

        CloseDir (directorio);
    end.
```