

ENTRADA/SALIDA EN GNU/LINUX.

Primitivas de E/S en GNU/Linux

Como se explicó en las clases teóricas, las llamadas al sistema referidas a dispositivos de entrada/salida se separaban en tres categorías:

- Servicios de Fecha y Hora.
- Temporizadores.
- Servicios de entrada/salida sobre dispositivos.

Para ver todas las funciones de manejo de entrada/salida vea la documentación de la RTL de FreePascal ([rtl.pdf](#)).

Servicios de Fecha y Hora

Ingresa 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, nómbrelo “**prog_0701.pp**” y compílelo.

```
Program prog_0701;  
  
Uses oldlinux;  
  
begin  
  Write('Segundos transcurridos desde Epoch (00:00 1/1/1970): ');  
  Writeln(GetEpochTime);  
end.
```

Este programa hace uso de la función **GetEpochTime()** que nos retorna el tiempo del sistema en formato *UNIX timestamp*, es decir, los segundos transcurridos desde el 1/1/1970.

(1) Tomando como base el timestamp calcule aproximadamente cuál es la fecha que le devuelve el sistema (convierta el timestamp a años, meses, y súmelos a la fecha base).

Escriba el siguiente programa y nómbrelo “**prog_0702.pp**” y compílelo.

```
Program prog_0702;  
  
Uses oldlinux;  
  
Var epoch: longint;  
    anio, mes, dia, hora, min, seg : word;  
  
begin  
  epoch := GetEpochTime;  
  Writeln('Segundos transcurridos desde Epoch: ', epoch);  
  EpochToLocal(epoch, anio, mes, dia, hora, min, seg);  
  Writeln('Fecha: ', dia:2, '/', mes:2, '/', anio:4);  
  Writeln('Hora: ', hora:2, ':', min:2, ':', seg:2);  
end.
```

(2) Ejecute el programa y compare los resultados con los de (1). A qué se deben las discrepancias?

Escriba el siguiente programa y nómbrelo “**prog_0703.pp**”, compílelo y pruebelo.

```
Program prog_0703;

Uses oldlinux;

Var anio, mes, dia, hora, min, seg, mseg, useg : word;

begin
  GetDate(anio, mes, dia);
  GetTime(hora, min, seg, mseg, useg);
  Writeln('Fecha: ', dia:2, '/', mes:2, '/', anio:4);
  Writeln('Hora: ', hora:2, ':', min:2, ':', seg:2,
        ':', mseg:3, ':', useg:3);
end.
```

(3) En qué casos resultaría más útil emplear las funciones **GetDate()** y **GetTime()** en lugar de **GetEpochTime()**? En qué casos no?

Servicios de Temporizadores

Para el manejo de temporizadores podemos emplear el servicio **Alarm()** que solicita el envío de una señal (de tipo **SIGALRM**) al proceso un determinado número de segundos luego de invocar al servicio.

Escriba el siguiente programa y nómbrelo “**prog_0704.pp**” y compílelo.

```
Program prog_0704;

Uses oldlinux;

Procedure MSenialAlarma(Senial : longint);cdecl;
begin
  Writeln ('Recibida senial de alarma!');
end;

begin
  Writeln('Armando senial de alarma...');
  Signal(SIGALRM,@MSenialAlarma);
  Writeln ('Solicitando una alarma en 10"...');
  Alarm(10);
  Writeln ('Pausa hasta recibir alguna senial...');
  Pause;
  Writeln ('Terminando');
end.
```

(4) Pruebe el programa. Para qué situaciones sería mejor emplear **Alarm()**/ **Pause()** en lugar de la función **delay()** de FreePascal? Vea la documentación de las funciones *Alarm*, *Pause*, *delay* y de la unit *Crt* en [rtl.pdf](#).

Servicios de E/S sobre dispositivos

La entrada salida sobre dispositivos se implementa en GNU/Linux a través de varios servicios:

- `ioctl()`
- `tcgetattr()`
- `tcsetattr()`
- `tcsendbreak()`, `tcdrain()`, `tcflow()`, `tcflush()`,.

(5) Obtenga más información sobre la finalidad de las funciones mencionadas en la documentación de la unit *OldLinux* ([rtl.pdf](#)) y en la página `man` de *tcgetattr*.

Los atributos de los dispositivos se definen en el registro `termios` que tiene la siguiente estructura:

```
Termios = record
    c_iflag : Cardinal;
    c_oflag : Cardinal;
    c_cflag : Cardinal;
    c_lflag : Cardinal;
    c_line : Char;
    c_cc : Array[0..NCCS-1] of Byte;
    c_ispeed : LongInt;
    c_ospeed : LongInt;
end
```

(6) Explique los elementos del *registro termios* (vea “`man 3 termios`”).

Escriba el siguiente programa nombrandolo “`prog_0705.pp`” y compílelo.

(7) Pruebe el programa y analice su funcionamiento y los resultados mostrados. Por qué los “atributos actuales” aparecen de manera “desprolija”?

```
Program prog_0705;

uses oldlinux;

procedure MuestraTermios(var tios:Termios);
begin
    WriteLn('Flags Entrada: $',hexstr(tios.c_iflag,8)+#13);
    WriteLn('Flags Salida: $',hexstr(tios.c_oflag,8));
    WriteLn('Flags Linea: $',hexstr(tios.c_lflag,8));
    WriteLn('Flags Control: $',hexstr(tios.c_cflag,8));
end;

var oldios, tios: Termios;

begin
    WriteLn('Atributos originales:');
    TCGetAttr(1,tios);
    MuestraTermios(tios);
    oldios:=tios;
    WriteLn('Estableciendo modo raw');
    CFMakeRaw(tios);
    TCSetAttr(1,TCSANOW,tios);
    WriteLn('Atributos actuales:');
    TCGetAttr(1,tios);
    MuestraTermios(tios);
    TCSetAttr(1,TCSANOW,oldios);
    WriteLn('Valores originales reestablecidos.');
```

Escriba el siguiente programa y nómbrelo "prog_0706.pp":

```
Program prog_0706;

uses oldlinux;

var oldios, tios: Termios;
    fd : Longint;

begin
  fd := FDOpen('/dev/tty1',Open_RdWr);
  if IsATTY(fd) then
  begin
    WriteLn('Escriba algo de texto y enter para continuar...');
    ReadLn;
    WriteLn('Abierta terminal: ', TTYname(fd) );
    TCGetAttr(fd,tios);
    WriteLn('Salvando atributos de ', TTYname(fd), ':');
    oldios:=tios;
    WriteLn('Desactivando ECHO en ', TTYname(fd), '...');
    tios.c_lflag := tios.c_lflag XOR ECHO; // *
    TCSetAttr(fd,TCSANOW,tios);
    WriteLn('Escriba algo de texto y enter para continuar...');
    ReadLn;
    TCSetAttr(fd,TCSANOW,oldios);
    WriteLn('Valores originales reestablecidos.');
```

(8) Compílelo y ejecútelo. Observe el comportamiento de la consola antes y después de cambiar sus atributos. Qué cambios observa? Qué hace la línea comentada con el asterisco? (Vea la documentación de las constantes de la unit OldLinux para saber más de ECHO). Por qué se guardan los atributos en la variable oldios?

Performance de las operaciones de E/S

Escriba el siguiente programa y nómbrelo "prog_0707.pp":

```
program prog_0707;

uses oldlinux;

var i: integer; f: longint; p: pointer;

procedure MuestraHora();
var hora, min, seg, mseg, useg : word;
begin
  GetTime(hora, min, seg, mseg, useg);
  WriteLn(hora:2,':',min:2,':',seg:2,':',mseg:3,':',useg:3);
end;

begin
  WriteLn('Operaciones en memoria: ');
  Write('Inicio: '); MuestraHora();
  for i:=1 to 1000 do begin
    GetMem(p,sizeOf(i));
    FillChar(p,sizeOf(i),i);
  end;
  Write('Fin: '); MuestraHora();
  Write('Presione enter para continuar...'); ReadLn;

  WriteLn('Operaciones en pantalla: ');
  Write('Inicio: '); MuestraHora();
  for i:=1 to 1000 do Write(i); //incremento y lo muestro...
  Write('Fin: '); MuestraHora();
  Write('Presione enter para continuar...'); ReadLn;

  WriteLn('Operaciones en disco (Sincronas): ');
  f := fdOpen('test.txt',Open_WrOnly OR Open_Creat OR Open_Sync);
  Write('Inicio: '); MuestraHora();
  for i:=1 to 1000 do fdWrite(f,i,1); //incremento y grabo...
  Write('Fin: '); MuestraHora();
  fdClose(f);
  Write('Presione enter para continuar...'); ReadLn;

  WriteLn('Operaciones en disco (Asincronas): ');
  f := fdOpen('test.txt',Open_WrOnly OR Open_Creat);
  Write('Inicio: '); MuestraHora();
  for i:=1 to 1000 do fdWrite(f,i,1); //incremento y grabo...
  Write('Fin: '); MuestraHora();
  fdClose(f);
end.
```

(9) Compile y verifique el funcionamiento del programa. Ejecútelo al menos 5 veces y anote los resultados en cada ocasión. Calcule la demora promedio de cada tipo de operación y compárelas. Explique (asistido por la teoría) el por qué de las diferencias.

Confeccione un informe **original, conciso y completo** donde se dé respuesta a las preguntas y consignas precedidas por un número encerrado entre paréntesis. Éste informe deberá ser confeccionado y entregado por cada grupo que llevó a cabo las actividades. La longitud máxima del informe es de tres páginas (sin contar las líneas correspondientes a código fuente).