

MIGRANDO DE CLIPPER A CLIP

SACL a882sacl@yahoo.com.ar

Grupo Clip clip-castellano@gruposyadoo.com.ar

Versión 0.1 - 06/06/2005

Revisiones:

1) POSIBLES PROBLEMAS E INCOMPATIBILIDADES

La primera gran pregunta que se hace un Clippero, es si el programa que tiene desarrollado es 100% compatible con Clip. O sea, si será tan fácil como tomar un *.prg y compilarlo siguiendo las instrucciones de nuestro primer manual (Introducción a Clip) y obtener automáticamente un nuevo ejecutable. Pues bien, a continuación intentaremos dilucidar este cuestionamiento.

Uno de los más importantes objetivos de Clip, es la compatibilidad con Clipper. Pero cuando la compatibilidad entra en contradicción con la funcionalidad, se privilegia esta última opción. Esta es la causas más importante que genera las incompatibilidades

Debemos tener presente, que estamos trabajando en otro sistema operativo, por lo que a la compatibilidad del lenguaje se agregan factores inherentes al mundo GNU/Linux.

2) ELEMENTOS DE LINUX

A) GCC 3.2.x tiene errores, Clip no lo soporta.

B) Como regla en GNU/Linux, el usuario debe presionar la tecla ESC dos veces. Clip hace posible evitar ese inconveniente emulando un segundo ESC con un intervalo de teclado. La longitud del intervalo es controlado por `SET(_SET_ESC_DELAY,<nMilisegundos>)`.

C) Es imposible abrir una tabla en modo SOLOLECTURA EXCLUSIVO en Clip bajo GNU/Linux. En tal caso, la tabla es abierta en modo SOLOLECTURA COMPARTIDO. La causa es la diferencia de los mecanismos de bloqueo, entre los archivos DOS y GNU/Linux.

D) Clip puede trabajar de dos formas: En modo **Pantalla Completa** (*full screen*) y en modo **Flujo de Salida Estándar** (*stream*). Las aplicaciones Clip se inician por defecto en modo "stream", es decir las salidas de las aplicaciones son colocadas en el "stdout stream". Una aplicación trabaja en este modo hasta que alguna operación de pantalla completa sea ejecutada (p.ej.: mostrar una ventana, mostrar datos en filas y columnas, recepción desde teclado, etc.). Este modo es útil para aplicaciones CGI (Common Gateway Interface; como una interface entre HTML y programas). Cuando se ejecuta alguna operación de pantalla completa, la aplicación inicializa el terminal, cambia a modo pantalla completa y detiene la salida "stdout stream".

Se aconseja seguir estas dos simples reglas. Para una aplicación tipo Clipper, deberíamos comenzar con el comando **CLEAR SCREEN**. En cambio para una tipo CGI, se debería redefinir el bloque de error estándar para evitar activar las salidas a pantalla (usar la función **errorblock()**).

E) En el sistema de archivos GNU/linux, la máscara *.* significa todos los archivos con un punto en sus nombres (“.”). Use la constante macro FILE_MASK, es dependiente de la plataforma (“*” en Unix y “*.*” en Windows).

F) - El sistema GNU/linux, diferencia el tamaño de las letras en los nombres de archivo (**USE table.dbf** y **USE TABLE.DBF** no son comandos equivalentes). Para ayudar a portar aplicaciones DOS a Unix, se recomienda usar el comando **SET TRANSLATE PATH ON**. Cuando lo activamos (ON), todos los nombres de archivos y directorios son traducidos a minúsculas. En este caso, la vía y el archivo mismo deben estar en minúsculas en el disco.

Los nombres de disco (C:, D:, etc.) pueden ser asignados a directorios de la siguiente forma:

```
SET ("C:", "/home/rust/diskC")
SET ("D:", "/home/rust/diskD")
```

Por omisión, el disco C: es asignado al directorio raíz (/).

También te puedes ubicar en el directorio donde están todos los archivos *.prg y ejecutar el utilitario “lowname” de la siguiente forma:

```
~/mis_prg$ $CLIPROOT/bin/lowname *
```

Te dejará todos los nombres de archivos en minúsculas.

G) Normalmente, usamos símbolos pseudográficos para construir nuestros bordes de pantalla, ya sea simples o dobles. En linux podemos tener algún problema al visualizar los caracteres dobles, en ciertos tipos de terminales. Para hacerlo posible, hay que realizar algunos cambios en la configuración usando “termcap” en vez de “terminfo”. Se debe proceder de la siguiente forma: export TERMCAP=/etc/termcap y luego verificar la entrada “gl” para gráfico simple y “g2” para caracteres de gráfico doble, en el archivo /etc/termcap.

H) Visualizar códigos de caracteres ascii menores de 32, es imposible para los terminales GNU/linux. Estos caracteres son mostrados como una cadena resaltada con el código *<nSymbol> + 32*.

3) ELEMENTOS DE CLIP

1) La función **SETBLINK()** no trabaja con Clip bajo GNU/linux (este siempre retorna .T.). Cuando usas el atributo BLINK (en la definición de color de Clipper), algunos terminales muestran texto parpadeante y otros el fondo de pantalla intensificado (color más iluminado). La consola linux en modo texto VGA parpadea, en cambio en modo FB intensifica el fondo. Si estás usando el modo consola, puedes usar el programa “SVGATextMode” para cambiar el adaptador VGA a modo BG-HI (mira la opción “16color” en /etc/TextConfig). Si estás usando otro terminal, tienes que encontrar alguna forma de cambiarlo al modo apropiado.

Debido a que esto no es controlado desde el lado del servidor, encontrarás interesante esta función:

blinkattr([IHabilita]) -> IEstadoPrevio

Esta permite, bloquear el envío del atributo BLINK desde una aplicación Clip al terminal. De manera que si tu no deseas setear el terminal en modo BG-HI, puedes deshabilitarlo usando el atributo BLINK y el texto no parpadeará (pero tampoco el fondo podrá ser intensificado):

blinkattr(.f.)

2) Algunas funciones RDD (por ejemplo **dbskip()**), no generan un error EG_NOTABLE cuando no hay áreas de trabajo en uso con Clipper 5.01, pero si con Clipper 5.2 y 5.3. En tal caso, Clip trabaja similar a Clipper 5.3.

3) En Clip, AS y REF son palabras reservadas. Ver descripción de la opción **-q** del compilador.

4) En Clipper, **CHR(<nConstante>)** es constante, traducida en tiempo de compilación. En Clip **CHR()** es similar a las funciones usuales, es decir, es evaluada en tiempo de ejecución.

5) No uses las palabras claves; **FIELD, WHILE, FOR, CASE, END**, etc., como nombres de variables.

6) En algunas circunstancias, la función if(,) es interpretada como la instrucción **IF ... ELSE ... ENDIF**. En tales casos el compilador genera una advertencia.

7) Con Clipper la función **TYPE()** invoca algunas funciones (p.e. SUBSTR()) y no otras (p.e. **ALERT()**). Clip no invoca ninguna función y retorna "UI" para expresiones conteniendo cualquier función de llamada.

8) Clipper tiene una extraña característica:

```
if NIL==0 // genera un error en tiempo de ejecución.  
if (NIL==0) // OK
```

En Clip esto no es soportado, por tanto ambas estarían OK.

9) La función **INDEXEXT()** retorna el sufijo del índice (la extensión) en minúsculas.

10) Se debería revisar la utilización de las funciones **DISPBEGIN()** y **DISPEND()**. En aquellas funciones y comandos (**PROMPT/MENU, GET/READ**, etc) donde el procesamiento de visualizar y teclear es ejecutado en forma independiente, Clip siempre usa un modo de salida optimizado, de manera que emplearlos es innecesario. Sólo es deseable su utilización, pero no necesaria, en aquellos lugares donde visualizas y procesas simultáneamente la entrada desde el teclado.

11) La restricción del tamaño del nombre de una función o procedimiento a 10 caracteres, ha sido removida. Esto se aplica igualmente a los nombres de las variables. Para simplificar la transición a una longitud de más de 10 símbolos, debes usar la directiva **#xdefine** ó **#define** (ésta última insensible al tamaño de las letras).

12) La restricción del tamaño de las cadenas a 64 Kb ha sido removida.

13) La restricción del tamaño de un arreglo a 4.096 Kb y/o 8.192 elementos, ha sido removida.

14) La función **MEMOEDIT()** está basada en la clase **TEXTEDIT** y pueden haber algunos problemas con las funciones de usuario para **MEMOEDIT()**.

15) Ahora `get:baddate()` es un método y no un atributo.

16) El objeto **TBrowse** es visualizado con efecto distinto al estándar, lo cual elimina problemas con el número de campos mostrados y posicionamiento del cursor cuando se está navegando sobre la tabla con funciones de usuario.

17) El comando **RUN** estropea los contenidos de salida de la pantalla del terminal en uso. Este problema, no puede ser completamente resuelto sin recurrir a algunos trucos, como el impedir que varios programas salgan independientemente a un dispositivo, de la misma forma como si varios programas independientes saquen la información hacia un puerto COM. Recomendamos usar algo parecido a esto:

```
run &comando
wait "Presione cualquier tecla"
clear screen // o restaure la pantalla anterior.
```

Otra peculiaridad más del comando **RUN**, en GNU/linux, es que corre el “shell” (similar al command del DOS) y luego ejecuta el programa. Si bien “**run cd mydir**” funcionará, después de salir del “shell”, el corriente directorio será el mismo que antes del comando.

18) En el objeto Error, existe alguna incompatibilidad.

19) Es preferible usar **inkey(0.1)** en vez de **inkey()**, de otra forma la aplicación usará demasiado tiempo del procesador (CPU). Particularmente, debemos hacer algunos cambios en el DBU concernientes al **inkey()** y su estabilización en el **TBrowse**. Estos cambios están abarcados por **#ifdef __CLIP__**.

20) **RESTORE FROM** siempre opera como con **ADDITIVE**.

21) El comando **SET PROCEDURE** no es y probablemente no será soportado. Enlace el programa completo desde los archivos objetos usando el compilador Clip (muy práctico usar un archivo “Makefile”) o use las nuevas características relacionadas con archivos PO. Lea acerca de la función **LOAD()**.

22) La función **SETCOLOR()** tiene 10 pares de colores, los 5 pares nuevos no tienen ningún significado especial.

Ya con estas consideraciones, están listos para traspasar sus aplicaciones de Clipper a Clip. Unas últimas observaciones:

- Traten de usar la sintaxis del Clipper 5.x y vayan abandonando la del Summer 87.

- Clip trae incorporadas sus propias y poderosas librerías, traten de migrar a su utilización. Si tuviesen la necesidad de utilizar una librería de terceros (contando con las fuentes en Clipper), pueden portarla fácilmente adaptándola y recompilándola con Clip.