INTRODUCCIÓN A CLIP, CLIPPER EN LINUX

SACL a882sacl@yahoo.com.ar
Grupo Clip clip-castellano@gruposyahoo.com.ar
Versión 0.1 - 21/05/2005

Revisiones:
0.2 - 29/05/2005
0.3 - 04/06/2005

INDICE

1) INTRODUCCIÓN

Este manual en español, nació producto de la ambigüa información que existía, para nosotros los novatos hispanoparlantes sobre CLIP. Lo primero que se nos ocurrió, fue visitar la página de los desarrolladores donde encontramos documentación en ruso e inglés, la cual intentamos traducir.

Al cabo de un tiempo, descubrimos que si lo hacíamos siguiendo la estructura que ellos preconizaban, la introducción a este mundo de CLIP no sería muy sencilla, así que optamos por ir desarrollando una instalación y en forma paralela, ir describiendo todos las vicisitudes enfrentadas. Para esto se recurrió a la desinteresada participación de todos los integrantes de la lista Clip en español.

Como nada es perfecto, las descripciones aquí vertidas corresponden a un sistema Debian Sarge. Esta distribución, es una de las más estables y de más fácil instalación de paquetes. En cuanto a la interfaz gráfica de usuario GUI, pueden usar cualquiera compatible con GTK, pues suponemos en un futuro cercano, hacer uso de esta librería. En esta guía se usó Icemaker por su compatibilidad con GTK y Qt , además de su escasa demanda de recursos.

Ahora bien, dentro de linux podríamos habernos decantado por (x)Harbour o Clip. Lo que nos ha decidido por Clip, fue su soporte para Mysql, Postgresql., Oracle, DBF/CDX. Todo sin limitaciones. Por ejemplo, en (x)Harbour los CDX vienen sin soporte rushmore y Mysql viene en una librería adicional básica. Si uno quisiera toda la potencia, tendría que optar por la distribución comercial de xHarbour.

Debemos dejar en claro para los puristas de linux, que existen diversas formas de obtener o realizar las tareas aquí comentadas, pero no olvidar que no se trata de un manual de linux.

Este manual pretende introducirlos en este mundo de Clip, hasta el momento de poder llegar a compilar tu primer programa "Hola Mundo".

2) POR QUE USAR CLIP

1) Lenguaje totalmente compatible con Clipper, al cual se le han agregado un set extendido de instrucciones que lo hacen más poderoso y actualizado. Existe una leve incompatibilidad con algunas instrucciones, que pueden ser fácilmente resueltas usando el método de buscar y reemplazar.

2) Modelo Programación a Objetos muy rápido y eficiente.

3) API de C tiene muchas más posibilidades que CA-Clipper.

4) Completo soporte internacional. Incluye ajuste de cualquier código de página o lenguaje al teclado.

5) Interfaz Gráfica de Usuario (GUI) basada en la librería GTK.

6) Uso de librerías dinámicas. Cargado y ejecución desde archivos externos y también en modo de tiempo de ejecución.

7) Compatibilidad con :

   - Clipper hasta la versión 5.3
   - Implementa casi todas las funciones de las Ca-Tools.
   - Soporta todas las funciones y características SIX.
     - MEM,DBF,DBT,FPT,NTX,CTX,CDX,NSX...
     - Tipos de datos VFP: horafecha, dinero.
     - RDD permite usar tus propios "drivers", o combinarlos.
     - RDD también permite ser usado al estilo OO.

8) Soporte Multitarea.

9) Bases de Datos Objetos CODB – CLIP

10) SQL :

   - Librerías y clases para acceso directo a servidores SQL (PostGreSql,MySql,Oracle,Interbase)
   - ODBC y puente ODBC para manejadores Windows.
   - Interpretador y comandos SQL, compatibles con FoxPro

11) Otras Posibilidades :

   - Operadores para objetos de sobrecarga ("Overloading").
   - Soporte de expresiones regulares.
   - Funciones para conexiones TCP/IP

- Funciones COM_()
- Tecnología similar a Rushmore, pero mucho más rápida y eficiente.
- Soporte para números de muy gran escala con ilimitada precisión.
- Soporte para archivos gráficos PNG, GD,JPEG y similares para líneas, rectángulos, sectores cuadrados, circunferencias, etc.
- Varias clases comunes para procesadores de palabras, programas html/cgi.
- Varias utilidades para interpretar patrones de documentos, *www_sql,clip_bl, clip_blank, clip_sql, clip_hindex, clip_hseek,* etc.
- Depurador interactivo multiventanas.

12) Sistemas operativos soportados: *linux, freebsd, openBsd, SPARC & x86 solaris, IBM mainframe con TurboLinux, Win32 (con herramientas de desarrollo CYGWIN).*

3) DOCUMENTANDONOS CON LINUX Y ENTRANDO EN CLIP

El principal obstáculo que tienen los programadores de ambientes Windows para querer incursionar en el mundo linux, es que éste difiere practicamente en todo. Es como entrar a otro mundo.

Existe información en internet acerca del tema pero, es tan abundante y variada que esto mismo provoca un cierto desconcierto en los novatos. Comenzando con las distintas distribuciones, aunque existe una base común, poseen ciertas peculiaridades específicas para cada una.

A continuación daremos una breve lista de páginas de ayuda, orientadas siempre a nuestra utilización de Clip :

(a) Página oficial de las traducciones al español de linux, incluye manuales, tutoriales y otros:
http://es.tldp.org/

(b) Página argentina de manuales:
http://www.linux-cd.com.ar/manuales/

(c) Página oficial Debian del Dpto. de Física de la Universidad de Chile:
http://ftp.cl.debian.org/man-es/

(d) Tutorial básico de MySql:
http://www.programacion.net/tutorial/mysql_basico/2/

(e) Manual de Referencia Debian :
http://qref.sourceforge.net/index.es.php

(f) Página de Clip en español:
http://www.lugli.org.ar/wiki/bin/view/Main/ClipDebian

(g) Pagina rusa, Oficial de Clip:
http://www.itk.ru/english/index.shtml

(h) Página de Clip en “Sourceforge”:
http://sourceforge.net/projects/x-clip

(i) Página del grupo CLIP, para unirse hacer click en el botón “¡Unite a este grupo!”:
http://ar.groups.yahoo.com/group/clip-castellano/

(j) Página de Paco Aldarias, Debian-Knoppix (excelente) :
http://www.ceedcv.com/paginas/pacodebian/linux.html

(k) Recopilación de un montón de tutoriales y cursos:
http://www.inforsist.net/todos_tut.php

(l) Sistema Gescom, realizado en Clip:
http://stockyfact.sourceforge.net/

(m) Hispafuentes (ver manuales en columna izquierda):
http://www.hispafuentes.com/index.php

(n) Página oficial de Cygwin, librería para la ejecución de programas linux (Clip) en Windows:
http://www.cygwin.com/

(o) Guía de instalación de Clip (en portugués):
http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=2584

(p) Traducción de los manuales “man”:
http://es.tldp.org/PAMELI/%257Epiernas/manpages-es/man-pages-es-1.28.tar.gz
http://es.tldp.org/PAMELI/%257Epiernas/manpages-es/man-pages-es-extra-0.8a.tar.gz

(q) Los “Cómos” en español :
http://es.tldp.org/htmls/comos.html

Bueno, como siempre, a lo mejor no están todas las que deberían estar pero, es un buen comienzo apoyarse en las páginas mencionadas arriba. De todas formas para los “impulsivos”, no es necesario leer toda esa documentación (aunque es recomendable) previa a la instalación de Clip, puesto que esta guía tratará de ser lo más clara posible en este proceso.

La distribución más fácil de instalar es Knoppix (basada en Debian), pero eres libre de instalar la que más te guste o esté a tu alcance. Suse, Mandriva (ex Mandrake y Conectiva) y Fedora (versión libre de Redhat), son excelentes alternativas.

4) VARIABLES DE AMBIENTE Y SETEO BASICO DE CLIP

La administración de los procesos de compilación y ensamble, es posible con la ayuda de las variables de ambiente. A continuación una breve discripción por tipo :

(A) Estándar, **no es necesario** modificarlas:

CC	Nombre del compilador C (omisión=='gcc')
CFLAGS	Opciones para el compilador C (omisión=='')
COMPILE_FLAG	Sólo compile, argumento para el compilador C (omisión=='-c')
COPT	Argumento de optimización para el compilador C (omisión=='-O2')

CDBG	Argumento depuración para el compilador C (omisión=='-g')
OUT_FLAG	Argumento de salida para el compilador C (omisión=='-o')
INCLUDE_FLAG	Argumento “include” para el compilador C (omisión=='-I')
OBJSUF	Sufijo para la salida de los archivos objetos (omisión=='.o')
SOBJSUF	Sufijo para la salida de los archivos compartidos (omisión=='.so')
LIBSUF	Sufijo para los archivos tipo librería (omisión=='.a')
SLIBSUF	Sufijo para los archivos tipo librerías compartidas (omisión=='.so')
SFLAGS	Opciones para el compilador C - versión compartida (omisión=='-shared')

(B) Nombres de librerías de definición básica:

CLIPLIB	Librería de soporte para tiempo de ejecución (omisión=='libclip.a')
CLIPSLIB	Librería compartida de soporte para tiempo de ejecución (omisión=='libclip.so')
CLIPLIBS	Librerías adicionales (omisión=='')

(C) Usadas para definir donde los archivos CLIP son colocados:

CLIPROOT	Raíz para todos los archivos CLIP relacionados (*)
CLIP_LOCALE_ROOT	Raíz para los mensajes locales (omisión == CLIPROOT)
CLIP_MODULE	Corriente módulo local (omisión=='environ') (**)

(*) Viene por omisión para una instalación local == '$HOME/cliproot')
(**) En tiempo de ejecución, su valor puede ser accesado a través de las macros predefinidas _CLIP_MODULE_.

(D) Utiles para la localización y trabajo con los diferentes códigos de página:

CLIP_LANG
: Usada para setear el código de página de las constantes cadena de los fuentes, durante la compilación(si las opciones del comando de línea -T y -S no son seteadas y el archivo .clipcharset no está presente). Si esta variable no es seteada, el valor es tomado de la variable LANG.

CLIP_HOSTCS
: Especifica una representación interna de la cadena (valor por omisión de CLIP_LANG en tiempo de compilación).

CLIP_CLIENTCS
: Define un codigo de página de salida para el terminal (valor por omisión de CLIP_LANG en tiempo de compilación). Si en tiempo de ejecución, no coincide con CLIP_HOSTCS, luego la salida será recodificada.

CLIP_LOCALE
Variable principal que define un “charset” usado por un programa CLIP. Si no está seteada, entonces serán chequeados CLIP_CLIENTCS, CLIP_CHARSET y CHARSET.

(E) Otras variables de ambiente:

CLIP_CMDSTR
: Lista de opciones del comando de línea, separadas por comas.

CLIP_NAMES
: Genera archivos .exe y .nm (valores posibles yes/no|on/off|0/1). El ensamblador C no puede ver si todas las funciones necesarias estarán o no disponibles en las librerías. Debido a esto los archivos .exe y .nm, los cuales son colocados cerca de sus correspondientes archivos objetos y librerías, dan la posibilidad al compilador CLIP, durante la generación del ejecutable, de comparar el contenido de esos archivos y determinar que funciones estarán extraviadas en tiempo de ejecución.

CLIP_LOGLEVEL
: Define el nivel de "log". Los valores 1,2 y 3 dan la información más detallada para el usuario. 4 - muestra todas las llamadas a funciones. 5 - muestra el máximo de la información detallada para análisis del programa en tiempo de ejecución sin usar el depurador.

CLIP_SCANMODE
: Define el modo de trabajo para el terminal (valores posibles: no, ioctl, terminal).

CLIP_SCANSTART
: Define la secuencia para activar "scancodes passong" (por ejemplo \033[R).

CLIP_SCANSTOP
: Define la secuencia para desactivar "scancodes passong" (por ejemplo \033[S).

CLIP_KEYMAP
: Descripción del Mapa de Teclado para el manejador de teclado de CLIP.

CLIP_LINECHARS
: Símbolos para los pseudográficos simples.

CLIP_DLINECHARS
: Símbolos para los pseudográficos dobles.

CLIP_COLORMAP
: Recodifica la tabla de colores.

A continuación necesitamos setear nuestras variables. Debemos tener presente dos cosas; una es el lugar físico donde realizaremos la declaración y otra, es el momento en que debemos hacerlo (antes o después de la generación de nuestro compilador CLIP).

Lectura **obligada** para los que venimos del ambiente DOS/Windows, es un COMO que explica claramente nuestro nuevo entorno de programación linux. Tómese su tiempo y léalo completa y detenidamente :

http://es.tldp.org/COMO-INSFLUG/COMOs/Dos-a-Linux-Como/

De la página anterior hemos obtenido este cuadro, para tener una idea de los archivos que debemos modificar para setear el ambiente de nuestros programas. Vendrían a equivaler a nuestros Config.sys y Autoexec.bat del mundo DOS.

```
FICHEROS                        NOTAS
/etc/issue                      Establece el mensaje antes del login.
/etc/motd                       Establece el mensaje después del login.
/etc/profile                    Establece el PATH y otras variables, etc.
/etc/bashrc                     Define alias y funciones, etc.
/home/usuario/.bashrc           Define alias y funciones para usuario.
/home/usuario/.bash_profile     Establece entorno y ejecuta programas para usuario.
/home/usuario/.profile          Establece entorno y ejecuta programas para usuario.
```

Las variables de ambiente pueden ser seteadas usando alguno de los archivos descritos arriba o vía interpretador de comandos. Posterior a la creación de CLIP, puede ser usado el archivo

$CLIPROOT/etc/environment y los archivos dependientes para el lenguaje y terminal:

$CLIPROOT/lang/$LANG $CLIPROOT/lang/$LC_ALL
$CLIPROOT/lang/$CHARSET $CLIPROOT/term/$TERM.

Algunas variables pueden tener opciones duplicadas en la línea de comandos. Estas últimas prevalecen sobre los seteos de las variables de ambiente.

5) OBTENIENDO CLIP Y PREPARANDO SU INSTALACION

La distribución de los fuentes presenta dos variantes: una dividida en paquetes por separado y otra compacta (todo en uno, por ejemplo el clip-prg-1.1.14.tg). Esta última es la recomendada y la que abordaremos.

Puedes bajar lo que necesites desde el servidor ftp principal: ftp://ftp.itk.ru/pub/clip

O desde los sitios réplicas:

Muy rápido, es actualizado cada noche ftp://ftp.linux.ru.net/mirrors/clip
Desde el proyecto Sourceforge http://sourceforge.net/projects/x-clip

1) Estando en nuestra raíz, creamos el directorio "clip_fuentes" (puedes darle otro nombre a elección tuya):

~$ *mkdir clip_fuentes*

Nos cambiamos a este directorio y bajamos la última distribución de Clip, en nuestro caso la 1.1.14-1 (al 6 de Marzo del 2005). Pueden utilizar cualquier método para esto, en nuestro caso, elegímos un servidor cercano (Brasil) y el comando wget:

~/clip_fuentes$ *wget http://ufpr.dl.sourceforge.net/sourceforge/x-clip/clip-prg-1.1.14.tgz*

También vamos a necesitar el manual (en inglés), para tener una referencia de los comandos y funciones extendidas (no se incluyen las sentencias de Clipper porque se asumen conocidas):

~/clip_fuentes$ *wget http://ufpr.dl.sourceforge.net/sourceforge/x-clip/clip-doc-en-html.tgz*

2) Desempaquetamos y descomprimimos nuestras fuentes:

~/clip_fuentes/$ *tar -zxvf clip-prg-1.1.14.tgz*

Aprovechamos de crear un directorio "doc" en "~/clip_fuentes". Aquí descomprimimos el manual en inglés (lo necesitamos para ver la sintaxis de comandos extendido), lo cual creará un directorio "html", donde se encontrará el archivo index.html que cargará la página del manual.

~/clip_fuentes/$ *tar -zxvf clip-doc-en-html.tgz*

3) Seteos para instalación Local o Global :

Local, es útil para un usuario en particular. No requiere de permisos especiales (de "root"), pero las librerías deben incluir su "path" en el ejecutable. Tienes que editar el archivo ~/.bash_profile e insertar las siguientes líneas. No olvides relogearte para que los cambios tomen efecto:

export CLIPROOT=$HOME/cliproot
PATH=$CLIPROOT/bin:$PATH

Global, es la instalación sistémica. El programa va a estar accesible para cualquier usuario. Requieres actuar con permisos de superusuario (root) para las compilaciones. En las librerías, no es necesario incluir el "path", éstas son buscadas en /usr/lib. Debes editar tu archivo /etc/profile y añadir las siguientes líneas:

export CLIPROOT=usr/local/clip/
PATH=$CLIPROOT/bin:$PATH

Recordar que los seteos también se pueden hacer desde la línea de comandos, por ejemplo:

./make system CLIPROOT=/usr/local/opt/clip

4) Debemos asegurarnos de tener instalados los siguientes paquetes, antes de generar CLIP. Recordar que las versiones y/o nombres pueden variar según la distribución de linux que se vaya a usar:

gcc : El compilador GNU C (versión 3.3.5, el por defecto de Debian Sarge).
libc6 : La librería GNU de C. Librerías compartidas e información de zonas horarias.
binutils : El ensamblador GNU. Utilidades binarias y de ensamblado.
make : La utilidad make.
flex : Rápido generador y analizador de léxico.
bison : Un generador de análisis de programas.
libmysqlclient14-dev : Para el uso de Mysql (Existe la 10, 12 y 14; usaremos la más actualizada).
libgtk2.0-dev : Para el ambiente gráfico GTK (Existe también la 1.2, usaremos la más actualizada).
libgpmg1-dev : Para el mouse, propósito general.
ncurses-bin : Programas relacionados al terminal y páginas de manual.
libncurses5-dev : Librerías y documentos para el desarrollador de ncurses.
zlib-bin : Librería de compresión.
zlib1g : Librería de compresión en tiempo de ejecución.
zlib1g-dev : Librería de compresión para desarrolladores.

Para poder construir los paquetes debian *.deb, se necesita lo siguiente:

time : El programa GNU de tiempo para medir el uso del recurso CPU.
dpkg-dev : Paquete de herramientas de construcción para Debian.
dpkg-cross : Herramientas para la compilación cruzada de paquetes Debian (Ej. generar *.rpm).
fakeroot : Habilita el uso del ambiente falso administrador ("fake root").
dh-make : Herramienta que convierte archivos fuentes en paquetes fuentes Debian.
dhelp : Sistema de ayuda en linea.
alien : Genera archivos *.deb a partir de paquetes precompilados no nativos de Debian (opcional).

Para poder construir los paquetes RedHat *.rpm, se necesita lo siguiente:
(** POR HACER, se aceptan contribuciones **)

6) GENERANDO NUESTRO COMPILADOR CLIP

Una vez asegurada la instalación de todos aquellos paquetes anexos, necesarios para no obtener mensajes de error al momento de la compilación, procedemos con la generación de nuestro CLIP.

Aquí tenemos dos opciones; generar un paquete de instalación específico para nuestra distribución de linux y luego instalarlo o exportarlo (*.rpm ó *.deb) o ... instalar directamente nuestro compilador Clip.

A ver si se entiende. Cuando uno crea un paquete precompilado (rpm,deb,etc), está generando un binario dependiente de las condiciones ambientales imperantes: versiones específicas de gcc, glib y otras librerías compartidas. Generalmente se usa esta técnica para entregar paquetes a distribuciones de linux específicas, listas para llegar y usar.

En cuanto al tipo de instalación, los creadores de Clip recomiendan la instalación local debido a que es más "limpia", fácil de parchar y de reinstalar para nuevas versiones. Otros miembros de nuestra lista, prefieren la instalación global, debido a que se necesitan permisos de "root" para modificar los archivos esenciales, evitando corrupciones involuntarias del sistema.

Dando por hecho, que hemos seteado nuestras variables de ambiente básicas y que estamos en el directorio de las fuentes de Clip -"clip-prg-1.1.14-1"-, digitamos lo siguiente (sólo una de las siguientes opciones):

TIPO	**COMANDO**	ALTERNATIVO	DESCRIPCION
Local	*./make local*	*./mklocal*	Instalación directa Local.
Global	*./make system*		Instalación directa Global
Debian	*./make deb*	*./mkdeb*	Genera paquetes Debian
RPM	*./make rpm*	*./mkrpm*	Genera paquetes RPM
Tgz	*./make tgz*	*./mktgz*	Sistemas sin administrador de paquetes
Tar.bz2	*./make tbz*		Idem, pero con formato tar.bz2

Ej: ~/clip_fuentes/clip-prg-1.1.14-1$ *./make local*

Dependiendo de la velocidad de nuestra máquina, esto puede tomar un buen tiempo. En mi "tarrito" Pentium I de 200 Mhz, se demoró como 40 minutos. Al final de la compilación debes ingresar una clave para un certificado de privacidad, yo ingresé la misma del usuario activo. Aún no sé que objetivo específico tiene esa consulta.

Te van a salir una serie de mensajes de advertencia o error. Para comprobar que tu Clip se generó correctamente, ejecuta en la consola: $ *clip -V* . Esto te deberia dar un mensaje con la version de Clip recien compilado. Si obtienes un mensaje de "comando no encontrado", es porque algo ha fallado en tu compilación.

La instalación local, creará dos directorios que colgarán de tu "home":
~/cliproot : Directorio base del compilador (bin) y las librerías (lib).
~/bin : Contiene enlaces simbólicos a ~/cliproot/bin.

Dependiendo de los paquetes que hayas tenido previamente instalado (Mysql, Postgresql, Gtk, etc), será la generación de algunas librerías. En mi caso en ~/cliproot/lib encontramos:

libclip
libclip-cti
libclip-gzip
libclip-r2d2
libclip-mysql
libclip-bzip2
libclip-rtf
libclip-fw
libclip-nanfor
libclip-codb
libclip-gtk2
libclip-ui
libclip-netto
libclip-com
libsup

Estas librerías vienen con las extensiones “a”, “ex”, “nm” y “so”.Recuerden lo que se mencionó sobre el significado de las extensiones:

*.a = Librería de soporte en tiempo de ejecución.
*.so = Librería compartida de soporte en tiempo de ejecución.

Instalación Selectiva: Puedes instalar sólo los paquetes necesarios. En este caso, debes bajar e instalar clip_dev.tgz. Este contiene el compilador Clip, algunas utilidades y el depurador estándar. Luego debes colocarlo en el directorio de los fuentes y desempaquetarlo. Cualquier otro paquete clip_* puede ser instalado de igual forma.

Local Selectiva:
- Seteas correctamente CLIPROOT
- $ *./configure* (en el directorio desempaquetado)

Global Selectiva:
- # *./configure -r*

Luego para ambas:
-$ *./make*
-# *./make install* (con los permisos adecuados).

Parchando tu distribución: En el servidor Ftp, puedes encontrar los archivos “patch.tgz”. Estos contienen el último parche acumulativo para el paquete clip_dev. Sólo desempaquétalo y procede como en la instalación selectiva.

La instalación para Windows, la trataremos en un documento aparte. Bueno al fin, estamos listos para compilar nuestro primer programa Clip, a continuación.

7) OPCIONES DEL COMPILADOR

Clip puede ser usado para generar archivos objetos (.o), de pseudocódigo (.po) o ejecutables. La sintaxis general es :

clip [<*CLIP opciones compilador*>] [<*lista_archivos*>] [<*opciones compilador C*>]

Donde:

<*CLIP opciones compilador*> Es una lista de una o más opciones, para controlar el curso de la compilación y/o ensamblado.

<*lista_archivos*> Es una lista separada por espacios de archivos fuentes, objetos o librerías para compilar o ensamblar. Todos los nombre de archivos deben de estar escritos con sufijos (su extensión), no se asumen tipos de archivos.

<*opciones compilador C*> Opciones pasadas al compilador C.

Cualquiera de las opciones descritas abajo, pueden ser colocadas en el archivo **.cliprc** en el directorio en uso, en el archivo **$HOME/.cliprc,** o en cualquier archivo en este último directorio. Por ejemplo, podrías crear un archivo con el siguiente contenido:

-e
-M
-O

A continuación las opciones más comunes para generar un programa Clip:

-e, --exec	Genera un archivo ejecutable.
-O, --optimise[=<val>]	Optimización para el compilador C.
-M, --main	Primer archivo lo define como procedimiento principal.
-n, --nomain	Suprime definición del nombre del prg como principal.
-a, --auto-memvar[=1|0]	Declara cualquier variable no definida como MEMVAR.
-r, --shared-exec	Si es posible,enlaza librerías compartidas,ejecutable más pequeño.
-s, --shared	Crea un objeto compartido, junto a **-e** crea ejecutables pequeños.
-o<name>, --output=<name>	Nombre del archivo ejecutable.
-d<name>, --outdir=<name>	Directorio de los archivos obj.
-I<path>, --include-path=<path>	Añade este directorio de búsqueda para los “Include”.
-L<path>, --library-path=<path>	Añade este directorio de búsqueda para las librerías.

Las restantes opciones son de más infrecuente uso, solo se mencionarán. Si desea una explicación más detallada consulte el manual en inglés (capítulo 6.2):

-h, --help
-H, --help-environment
-V, --version
-v[<level>], --verbose[=<level>]
-w, --namespace-warning[=1|0]
-c, --compile

-g,-b, --debug[=1|0]
-t, --show-syntax-tree
-p, --pcode
-l, --c-pcode

-D<name>, --define=<name>
-D<name=value>, --define=<name=value>
-U[<filename>], --use-std-ch[=<filename>]
-S<charset>, --source-charset=<charset>
-T<charet>, --target-charset=<charset>
-P, --preprocess
-C, --show-command
-R, --noremove-c
-N, --names
--static,--full-static
-E<name>=<value>, --environment=<name>=<value>
-q <word>, --compat=<word>

8) EJEMPLOS CON CLIP

Asumimos para los ejemplos, que no existen configuraciones previas en **.cliprc**.

(A) Construye una simple aplicación desde un archivo fuente sin procedimiento principal. El ejecutable será enlazado estáticamente con la libreria en tiempo de ejecución Clip RTL.

clip -eM test.prg

(B) Crea un módulo P-Code (archivo **.po**). Silencioso.

clip -pv0 funcs.prg

(C) Construye una aplicación desde tres fuentes. El resultado es un pequeño ejecutable, ensamblado dinámicamente con Clip RTL. **app.prg** contiene la declaración de un procedimiento principal.

clip funcs1.prg
clip funcs2funcs.prg
clip -es app.prg funcs1.o funcs2.o

(D) Construye una librería dinámica desde dos fuentes y la enlaza con una aplicación. Ver la descripción de la utilidad **clip_makeslib**. El resultado es un pequeño ejecutable enlazado con Clip RTL, el cual es capaz de trabajar sólo cuando **libmylib.so** esté colocado en el mismo directorio de trabajo.

clip funcs1.prg
clip funcs2.prg
clip_makeslib libmylib funcs1.o funcs2.o
clip -es app.prg ./libmylib.so

(E) Construye una aplicación, capaz de trabajar sólo cuando **ibmylib.so** está situado en /home/rust/lib.

clip funcs1.prg
clip funcs2.prg
clip_makeslib libmylib funcs1.o funcs2.o
cp libmylib.so /home/rust/lib
clip -es app.prg /home/rust/lib/libmylib.so

(F) Para ver la salida del preprocesador.

clip -P test.prg

(G) Compila en modo **C+P-code**. Ensambla dinámicamente con Clip RTL. Entregará más información del proceso ("verbose") y el resultado será un ejecutable del tamaño más pequeño posible.

clip -elsv2 app.prg

(H) Usa la opción **-l** del compilador C para ensamblar con la librería compartida **libclip-postgres.so** (manejador PostgreSQL para CLIP).

clip -els test.prg -lclip-postgres

Les recomiendo que vean los ejemplos básicos que vienen en los paquetes de instalación de los fuentes:

~/clip_fuentes/clip-prg-1.1.14-1/example$

De ahí extraemos la idea de nuestro primer programa en Clip, el famoso "Hola Mundo". Editamos un archivo vacío y le agregamos las siguientes líneas:

```
? '¡ Hola, mundo !'
?
```

Luego lo grabamos con el nombre **hola.prg** y lo compilamos con : $ *clip -erMO hola.prg*

---------- clip CLIP clip ----------