

INTERNACIONALIZACIÓN DE CLIP

SACL a882sacl@yahoo.com.ar
Grupo Clip clip-castellano@gruposyadoo.com.ar
Versión 0.1 - 01/07/2005

Revisiones:

TECLADO

Las aplicaciones CLIP tienen dos tipos de controladores de teclado: el modo ANSI y el modo SCAN.

En el modo ANSI, los códigos de las teclas son aceptados desde el terminal y dos modificadores Ctrl-J y Ctrl-K son usados para extender las posibilidades.

Un modificador cambia el código de la tecla, para las aplicaciones CLIP. Por ejemplo: presionando la tecla "1" se genera el código 49, presionando "Ctrl-J + 1" se genera el código 376, presionando "Ctrl-K + 1" también genera el código 376. El modificador Ctrl-J trabaja de la misma forma que ALT y, el modificador K trabaja de la misma forma que CTRL, pero no con todas las teclas.

El modo de trabajo de los teclados más avanzados es SCAN. En este modo, el terminal pasa al servidor códigos de teclas pulsadas y liberadas. Todas las combinaciones de teclas están disponibles en el caso de poseer correctas descripciones de "keymap" (mapas de teclas). Además hay posibilidad de agregar código de símbolos, usando "Alt + teclas". En estos casos se usan las descripciones de diseño de consola ubicados en "/usr/share/keymaps/i386/qwerty/*". Damos por hecho que te estás moviendo en la plataforma i386 y con teclados cuyo orden físico, debajo de los números, es "qwerty...".

Para trabajar con un terminal con capacidades SCAN, uno puede usar la consola de linux o nuestro emulador de terminal (bajo Windows) Stelnet (bajarlo de <ftp://ftp.itk.ru/pub/telnet>).

Para poder usar las capacidades avanzadas de Stelnet (Windows), hay que ejecutarlo de la siguiente forma:

```
stelnet -s -t linux-stelnet nombre_pc_servidor
```

Y lanzar la aplicación CLIP como sigue:

```
export TERM=linux-stelnet  
mi_programa_clip  
export TERM=linux
```

MOUSE

El mouse es soportado en los siguientes casos:

a) En consola, en modo terminal estándar usando el servicio "gpm".

- b) Sobre terminales X-Window (gráficos) con TERM=xterm ó TERM= terminales rxwt.
- c) Sobre otros terminales X-Window, en el caso que se haya definido la variable XTERM_MOUSE=ON
- d) Desde el emulador “stelnets” (Windows), con la variable de ambiente XTERM_MOUSE=ON.
- e) Desde consola a través del protocolo “ssh” sobre un Servidor (se requiere el paquete trans_1.2).
- f) Sobre un servidor Windows.

GENERANDO NUESTRO CHARSETS

Linux y por consecuencia CLIP, vienen preparados para su uso internacional. Esto significa que podemos adaptarlo, en teoría, a cada uno de nuestros requerimientos nacionales. Para esto se vale principalmente, de dos herramientas: “keymaps” y “charsets”.

Charsets : Traducido generalmente como “juego o tabla de caracteres”, es un archivo donde aparecen los 256 códigos ASCII en una columna y su correspondiente codificación a caracteres en otra. Generalmente en los primeros 128 no hay problemas, en los segundos 128 es donde se marca la diferencia. Este archivo es la base para interpretar, mostrar o imprimir nuestros caracteres. En general para el ámbito latinoamericano se usa la tabla “cp850” (codepage 850).

La nacionalización de una aplicación CLIP, incluye dispositivos de pantalla, teclado, constantes de cadena y funciones embebidas.

Los dispositivos de pantalla son la consola, el terminal y el emulador de terminal. Uno puede configurar el entorno por medios estándar que el S.O. proporciona, por ej. cargado de fonts, terminal dependiente de BIOS, o por controladores de teclado/pantalla.

Algunas funciones Clipper (tales como upper, lower, isalpha, isdigit), pseudográficos, coeficientes de peso para indexación y comparación, etc., tienen características nacionales. Se dispone de la utilidad “**gen_tbl**” para la creación de un archivo con estas especificaciones.

CLIP posee especiales seteos para poder usar datos de DBF para distintos usuarios con diferentes “charsets” simultáneamente:

```
set(“DBF_CHARSET”, “cp866”)
```

O en forma de comando:

```
set dbf charset to cp866
```

Donde cp866 es el nombre del archivo “**tbl**” (tabla) con las descripciones de las características nacionales.

Puede darse que los datos son almacenados en una DBF con un “charset”, un programa los opera con otro y un tercero, los muestra a su vez con un set distinto.

También existe la posibilidad de setear un “charset” para la impresora:

```
set("PRINTER_CHARSET","cp866")
```

O en su forma de comando:

```
set printer charset to cp866
```

¿ Cómo crearlo ?

Los “charsets” (set de caracteres) son usados para recodificar las entradas y salidas de un terminal, para transacciones con terminales pseudográficos, para determinar ordenamientos y, para recodificación de archivos y bases de datos en las finalizaciones de entradas y salidas.

Al igual que los “keymaps”, CLIP usa su propio formato para almacenar “charsets”, pero cuenta con una utilidad para la generación estándar de mapeos “charset-a-unicode”.

Para añadir nuevos “charset”:

1) Copia el apropiado mapeo de “charset” desde /usr/share/consoletrans/ (de cualquier distribución):

```
$ cp cp850.sfm.gz $CLIPROOT/chartsets/
```

2) Cambiarse al directorio y descomprimir cambiandole el nombre:

```
$ cd $CLIPROOT/chartsets/  
$ zcat cp850.sfm.gz > cp850.uni
```

Advertencia: Algunos archivos de mapeo unicode, proporcionan mapeo sólo para parte de los 256 símbolos en el “charset”. Para tales casos probablemente, debes agregar manualmente los caracteres faltantes.

3) Usa la utilidad **gen_tbl**, para la generación del archivo *.tbl:

```
$ zcat UnicodeData-2.1.8.txt.gz | $CLIPROOT/bin/gen_tbl cp850.uni > cp850.tbl
```

4) Setea la variable CLIP_CHARSET, en tu archivo de configuración “.profile” o “.bash_profile” :

```
export CLIP_CHARSET=cp850
```

Este seteo fue probado y me funcionó perfectamente. Falta verlo al imprimir y en la ordenación de claves de índices.

GENERANDO NUESTRO KEYMAPS

Keymaps: Traducido como “mapa de caracteres o teclas”, también se corresponde con un archivo donde se define el comportamiento del controlador del teclado (como se debe interpretar las pulsaciones de las teclas).

En el caso del teclado en modo SCAN, su diseño y codificación son hechos por medio de la creación de un archivo “keymap” (mapa de teclas) específico para CLIP, que viene con descripciones predefinidas para controladores de teclado.

Para que una aplicación CLIP pueda ser capaz de usar el modo SCAN, es necesario generar la descripción del “keymap” deseado usando la utilidad “**genmap.sh**”, para obtener nuestro mapa en el directorio “\$CLIPROOT/keymaps/”.

La configuración del mapa de teclado para ambiente de consola (“terminal”) en GNU/linux, se realiza con uno de los siguientes comandos:

```
# kbdconfig (Otras distribuciones)
# dpkg-reconfigure console-data (Debian)
(ver también “install-keymap” y “loadkey”)
```

Producto de esta acción, se selecciona un mapa desde “/usr/share/keymaps/i386/qwerty/” y se almacena como “/etc/console/*.kmap.gz” .

Los “keymaps” son usados en modo de teclado directo, por ahora posible sobre consola linux y, sobre algunos tipos de terminales que pueden ser switcheados a modo SCAN por secuencias de escape.

CLIP usa el mismo “keymap” del kernel de linux y, el procesamiento de SCAN es tomado prestado desde los fuentes del núcleo. Esto significa, que puedes fácilmente añadir nuevos “keymaps” y modificarlos cuando lo requieras.

¿ Cómo crearlo ?

1) Copiar el “keymap” deseado desde “/usr/share/keymaps/i386/qwerty/” o desde “/etc/console/” (puedes usar cualquier distribución linux):

```
$ cp /etc/console/boottime.kmap.gz $CLIPROOT/keymaps/
```

2) Cambiarse de directorio y descomprimir:

```
$ cd $CLIPROOT/keymaps/
$ gunzip boottime.kmap.gz
```

3) Usar “**genmap.sh**” para la generación de un “keymap” leible para CLIP:

```
$ ./genmap.sh $CLIPROOT/keymaps/boottime.kmap > es-cp850
```

Advertencia: El parámetro para **genmap.sh**, debe incluir el “path” completo.

4) Debo decirle a CLIP que ocupe mi mapa de teclado, en mi archivo de configuración “.profile” o “.bash_profile”, debo setear CLIP_KEYMAP=es-cp850 .

NOTA: No es necesario que lo generes. Si tienes configurado tu linux en español, CLIP tomará esas definiciones por defecto. Yo hice la prueba de cambiar el teclado y al iniciar la aplicación, me envía un mensaje de error sobre su uso en conflicto con la terminal. Debe de haber algún seteo que me faltó por definir (?). NO OLVIDAR que el ambiente gráfico (xterminal), maneja su propia configuración.

SECUENCIA DE INICIO DE UN PROGRAMA CLIP

Cuando un programa se inicia, el sistema de ejecución de CLIP, chequea la existencia de algunos archivos y trata su contenido de acuerdo a las asignaciones de ciertas variables de ambiente.

El primer archivo útil es “\$CLIPROOT/lang/\$LANG”. Este es un script que carga valores predeterminados para ciertas variables de ambiente relacionadas con el lenguaje. Si LANG está vacío, CLIP chequeará también la variable CHARSET. Por ejemplo, veamos el contenido del archivo ru_RU.KOI8-R :

```
#CLIP_KEYMAP se usa en modo de teclado “raw” (directo).  
CLIP_KEYMAP=ru-koi8-r
```

El próximo archivo es \$CLIPROOT/term/\$TERM, donde TERM es la variable que identifica el tipo de terminal usado y lo asocia a un archivo, que contiene un script para el comportamiento del teclado. Por ejemplo, veamos en contenido de \$CLIPROOT/term/linux-console :

```
# switchea sobre la consola linux un teclado “raw” (directo).  
CLIP_SCANMODE=IOCTL
```

TRADUCCION DE MENSAJES

El compilador de CLIP, almacena todas las constantes de cadena definidas como [cadena_de_datos] en los subdirectorios “\$CLIPROOT/locale.po*”.

Por ejemplo para las personas de habla hispana, podríamos usar los siguientes archivos ubicados en “\$CLIPROOT/locale.po/ISO-8859-1/”, para la traducción de los mensajes:

```
clip.po  
cliprt.po  
sys.po
```

De manera de que si existe la posibilidad, de hacer traducciones a otros lenguajes. Al momento de ejecutarse una aplicación, CLIP lee la variable de ambiente LANG y obtiene las apropiadas constantes desde los archivos antes mencionados.

Gettext y utilidades para los archivos de traducción “po”.

Gettext es un paquete común, ampliamente usado para los mensajes de localización. El paquete tiene dos partes principales: las utilidades de extracción y preparación del mensaje y, una librería para el manejo de mensajes en tiempo de ejecución.

CLIP usa algunas de las utilidades de “gettext” para la extracción de mensajes y preparación de archivos de mensajes, pero usa código propio para el manejo de los mensajes en tiempo de ejecución.

CLIP tiene numerosas utilidades para hacer el manejo de mensajes más fácil:

Ruta para obtener mensajes locales:

- Extracción de mensajes y creación del archivo “pot” (PO-planTilla). Para la extracción desde fuentes C, se puede usar el comando “xgettext” desde el paquete “gettext”. CLIP automáticamente extrae los mensajes encerrados entre paréntesis cuadrados “[]” y los almacena en el archivo *\$CLIPROOT/locale/pot/<nombreródulo>/<nombrearchivo>.pot* . El *<nombreródulo>* es determinado por CLIP desde el nombre del directorio de trabajo en uso, o puede ser especificado por la variable de ambiente CLIP_MODULE. El *<nombrearchivo>* se obtiene del nombre del archivo compilado.
- Usar el script “clip_msgmerge” para mezclar los archivos pot desde el directorio *\$CLIPROOT/locale.pot* con los archivos po traducidos en el directorio *\$CLIPROOT/locale.po*. El “clip_msgmerge” mezcla archivos para todos los locales en el directorio *\$CLIPROOT/locale.po/<local>*. Para añadir un nuevo lical, necesitas crear un directorio apropiado, por ejemplo; *\$CLIPROOT/locale.po/ru_RU.KOI8-R* para usar las locales de ru_RU.KOI8-R.
- Editar los archivos creados po, con cualquier editor de texto, o usar un editor especializado en archivos po, similar al kbabel de KDE.

NOTA: Para la definición del “charset” en el archivo po, por favor chequea el campo “Content-Type: test/plain, charset=ASCII” en la primera entrada del archivo. Debes especificar un valor válido de “charset” usado. Por ejemplo “Content-Type: test/plain, charset=KOI8-R”). Esto es muy importante para corregir el manejo de los mensajes.

- Y el último paso es invocar el script “clip_msgfmt”, para formatear mensajes dentro del archivo de mensajes compilado, en el directorio “*\$CLIPROOT/locale.mo*”.

Ahora CLIP, en tiempo de ejecución, obtendrá los mensajes necesarios desde estos archivos y los sustituirá en tu programa.