Linux Keystroke-HOWTO

Zenon Fortuna(*zenon@netcom.com*) Traduzione a cura di Veronese Andrea (*albero@maya.dei.unipd.it*)
(Aprile 1996) Versione 2.0, 4 Aprile 1995

Questo documento descrive come assegnare macro (sequenze di tasti o esecuzione di script) ad alcuni tasti della tastiera. Il metodo proposto è quello di usare **loadkeys(1)**, oppure quello di modificare il file `defkeymap.c` e relinkare il Kernel. Questo testo NON descrive la riconfigurazione dei tasti, come ad esempio [Backspace] o [Delete]. Per informazioni su come rimappare i tasti leggere il BackSpace Mini-HOWTO scritto da Stephen Lee. Il metodo descritto di seguito è stato testato sulla versione 1.2.1 di Linux e incluso nella distribuzione Slackware 2.2.0.

1 Revisioni a questo documento

4 Aprile 1995 - Versione 2.0
Adattato per Linux 1.2.1:
- semplici cambiamenti circa le modifiche proposte al file `my_defkeymap.map`.
- modifiche all'esempio di macro con lo **screen dumping** di /dev/vcs* (riversamento della schermata di una console virtuale in un file o alla stampante).

7 Maggio 1994 - Versione 1.0
Versione iniziale del "Keystroke-HOWTO", scritta per Linux 1.0.

2 Breve Descrizione

I driver di terminale virtuale e di tastiera attribuiscono come valore di default alla mappa dei tasti quello definito nel file `/drivers/char/defkeymap.c` presente nei sorgenti del Kernel. Si possono associare delle stringhe all'azione dei 12 tasti Funzione presenti sulla tastiera: dopo aver premuto uno di quei tasti, anche assieme a [Ctrl] o [Alt] (detti tasti modificatori), il terminale virtuale corrente aggiunge la stringa corrispondente ai suoi buffer d'ingresso e d'uscita, emulando quindi la digitazione di tale stringa dalla tastiera.

Associando una stringa appropriata ad un determinato tasto Funzione possiamo semlificare l'esecuzione di certi comandi, per esempio la chiamata ad uno Shell-script `/usr/local/bin/key_macro` che ci possiamo creare e modificare a piacimento.

3 Strumenti per la modifica del driver di tastiera

Possiamo usare **loadkeys(1)**, **dumpkeys(1)** e **showkey(1)**. L'utility **loadkeys(1)** aiuta a caricare nuove stringhe nei buffer di tastiera del Kernel, o anche a preparare nuovo codice C per modificare il Kernel. **dumpkeys(1)** dovrebbe essere usata per visualizzare la mappa attuale della tastiera se si volesse poi ispezionarla o modificarla. **showkey(1)** può aiutarci per ottenere il codice del tasto Funzione selezionato.

Se il vostro sistema Linux non ha queste utility potete ottenerle tramite ftp anonimo agli indirizzi

sunsite.unc.edu:/pub/Linux/system/Keyboards , oppure
tsx-11.mit.edu:/pub/linux/sources/system

coi nomi `kbd-0.89.tar.gz` oppure `kbd-0.90.tar.gz` . Usare lo GNU tar per scompattare i file desiderati.

4 Modificare il file tabella di conversione (keytable)

Il Kernel di Linux comprende il codice compilato `defkeymap.c`, il quale è generato con l'utility **loadkeys(1)** partendo dal file `defkeymap.map`. Entrambi i file sono inclusi nella directory `src/linux/drivers/char`.

Al fine di modificare il file `defkeymap.map` ce ne facciamo una copia locale con

`# cp defkeymap.map my_keytable.map` ; oppure
`# dumpkeys > my_keytable.map`

C'è anche un'ampia raccolta di file di configurazione della tastiera nella directory `/usr/lib/kbd/keytables`, dalla quale `defkeymap.map` può essere copiato e usato come file `src/linux/drivers/char/defkeymap.map` nel vostro sistema.

Il metodo che usa l'utility **dumpkeys(1)** è raccomandato perché può accadere che il vostro Kernel sia già stato modificato o creato per voi con un differente `defkeymap.map` rispetto a quello che potreste trovare nei sorgenti.

Leggiamo ora il contenuto del file `my_defkeymap.map`. Non ci sono più di 300 linee di codice e vi possiamo trovare 3 gruppi di dichiarazioni: il primo gruppo consiste in linee con la parola "`keycode`", anche preceduta da altre parole come "`alt`", "`control`", ecc.; nel secondo gruppo troviamo la parola "`string`"; il terzo ed ultimo gruppo consiste in linee con la parola "`compose`".

Altre informazioni sulla sintassi di **keytables(5)** possono essere reperite tramite il comando

```
% man keytables
```

4.1 Esempio di modifica del file tabella di conversione

Come esempio dell'assegnazione di una stringa-macro alla pressione di un tasto Funzione facciamo in modo che la combinazione [Ctrl+F1] chiami l'esecuzione del nostro Shell-script "`/usr/local/bin/key_macro`".

Prima di tutto dovremmo vedere qual è il codice del tasto Funzione [F1]. Possiamo usare l'utility **showkey(1)** per trovare tale codice, premendo [F1]. Altrimenti possiamo cercare la stringa "`F1`" nel file `my_keytable.map`, col risultato di ottenere la seguente linea:

```
keycode 59 = F1
```

Questa suggerisce che il codice per il tasto Funzione [F1] è 59. Inoltre definisce il fatto che, dopo aver premuto [F1], il driver di tastiera manda in uscita la stringa denotata dal codice stringa "`F1`". Per vederne il contenuto si può cercare sempre all'interno di questo file "`string F1`" e si trova:

```
string F1 = "\033[[A"
```

Cioè, dopo aver premuto [F1], il driver di tastiera manda "`Esc[[A`" (senza spazi).

Non bisognerebbe combiare questa stringa perché alcune applicazioni basano su di essa l'azione di [F1] come valore di default.

Comunque possiamo definire una nuova macro per la combinazione [Ctrl+F1], la quale non è riservata dal Kernel per altri usi speciali. Per vedere i codici del tasto [F1] premuto assieme a [Ctrl], [Shift] o con altri modificatori, possiamo cercare nel file `my_keytable.map` il codice 59 tramite

```
# grep 59 my_keytable.map
```

Nel caso in cui non vi sia alcuna linea con "`control keycode 59`" allora possiamo usare [Ctrl+F1] senza problemi (si può procedere anche se è presente una linea con "`shift control keycode 59`").

Aggiungiamo la seguente linea al file `my_keytable.map`:

```
control keycode 59 = key_symbol
```

dove "`key_symbol`" definirebbe l'azione della combinazione [Ctrl+F1]. Linux 1.2.* ammette un'allocazione dinamica delle stringhe, ma la stringa "`key_symbol`" può essere scelta solo tra un insieme ben definito di nomi. Tra gli altri sono permessi i simboli-tasto F1-F246. Nel mio sistema F21 non era usato nel file `my_keytable.map` ma ognuno dovrebbe ispezionare il proprio e scegliere il simbolo-tasto opportuno. Nel nostro esempio si può quindi concludere con la linea

```
string F21 = "/usr/local/bin/key_macro\n"
```

Riassumendo, abbiamo apportato due cambiamenti al file di partenza `my_keytable.map`: abbiamo dichiarato la nuova stringa F21 e abbiamo deciso che la combinazione [Ctrl+F1] chiamerà l'esecuzione del contenuto della stringa 21.

4.2 Modifica temporanea della configurazione della tastiera

Dopo aver opportunamente modificato il file `my_keytable.map` possiamo copiarne i cambiamenti al driver di tastiera del Kernel, usando l'utility **loadkeys(1)**:

```
% loadkeys my_keytable.map
```

La modifica al driver di tastiera del Kernel è concessa a chiunque abbia l'accesso in lettura al dispositivo "`/dev/console`".
Per verificare che i cambiamenti voluti siano stati effettivamente installati possiamo usare l'utility **dumpkeys(1)** e controllare il valore di F21, ad esempio

```
% dumpkeys |grep F21
```

Si ottiene:

```
keycode 59 = F1          F11       Console_13 F21
string F21 = "/usr/local/bin/key_macro\012"
```

```
la quale va bene perche "012", o LF, e equivalente a "\n".
```

Ora, premendo [Ctrl+F1] dovrebbe avvenire la chiamata allo Shell script "`/usr/local/bin/key_macro`", come desiderato.

4.3 Modifica permanente

I cambiamenti imposti da **loadkeys(1)** al driver di tastiera del Kernel rimangono effettivi solo fino al successivo reboot (o fino alla successiva chiamata a `loadkeys`).

(N.d.T.: Per una modifica permanente) possiamo aggiungere a `/etc/rc.d/rc.local` la chiamata a `loadkeys` con argomento il nostro `my_keytable.map`. Altrimenti, è possibile modificare `src/linux/drivers/char/defkeymap.c` e relinkare il Kernel con i nuovi valori di default.

Non si dovrebbe modificare manualmente il file `defkeymap.c` ma, piuttosto, generarlo tramite l'utility **loadkeys(1)**:

```
# mv defkeymap.c defkeymap.c.ORIG
# loadkeys -mktable my_keytable.map > defkeymap.c
```

Poi si dovrebbe creare il nuovo Kernel essenzialmente ponendosi alla radice dei sorgenti di Linux e usando **make(1)**.
Infine, per mezzo di **lilo(1)** dovremmo installare e riavviare il nuovo Kernel.

5 Esempio di script assegnato a un tasto

Uno script particolarmente utile come esempio di macro eseguibile premendo un tasto, può essere quello che riversa in un file o alla stampante la schermata del video (screen dumping).

Questo esempio è stato cambiato rispetto a quello della versione 1.0 di Linux a causa delle modifiche al Kernel, il quale non fornisce più la chiamata di sistema `ioctl(0,TIOCLINUX)`.

Al fine di leggere le schermate delle console virtuali si dovrebbero prima preparare alcuni file di dispositivo. Come utente "root" possiamo creare i seguenti file:

`# mknod` dev/vcs1 c 7 1

`# mknod` dev/vcs2 c 7 2

`...`

`# mknod` dev/vcs63 c 7 63

Naturalmente è sufficiente avere tanti file /dev/vcs* quante sono le console virtuali che si stanno usando.

Il codice sotto riportato dovrebbe essere considerato come esempio di possibile file `/usr/local/bin/key_macro`:

```
#!/bin/sh
#
# Questo e un esempio utile di script
# eseguibile premendo un tasto
#

VT_NUMBER=`tty|cut -c9-`
FILE=/tmp/vt$VT_NUMBER.dump
cp /dev/vcs$VT_NUMBER $FILE
echo SCREEN DUMP saved in $FILE
#
# Uncommenta la linea sotto se vuoi
# stampare il file "schermata"
#
# lpr $FILE
```

6 Considerazioni

Non c'è limite pratico alla somma delle lunghezze di tutte le stringhe che vorremmo caricare nel driver di tastiera. Il precedente buffer della lunghezza di FUNC_BUFSIZE, costante (posto a 512 byte), è stato rimpiazzato in Linux 1.2.* da una strategia di allocazione del buffer dinamica ... in pezzi di 512 byte ciascuno.

La copia più recente del Keystroke-HOWTO può essere trovata all'indirizzo:

ftp.netcom.com: /pub/ze/zenon/linux/howto

7 Altre idee?

Nel caso troviate qualsiasi cosa che valga la pena di aggiungere a questo documento, spedite i vostri suggerimenti a *zenon@netcom.com* - grazie (zf)