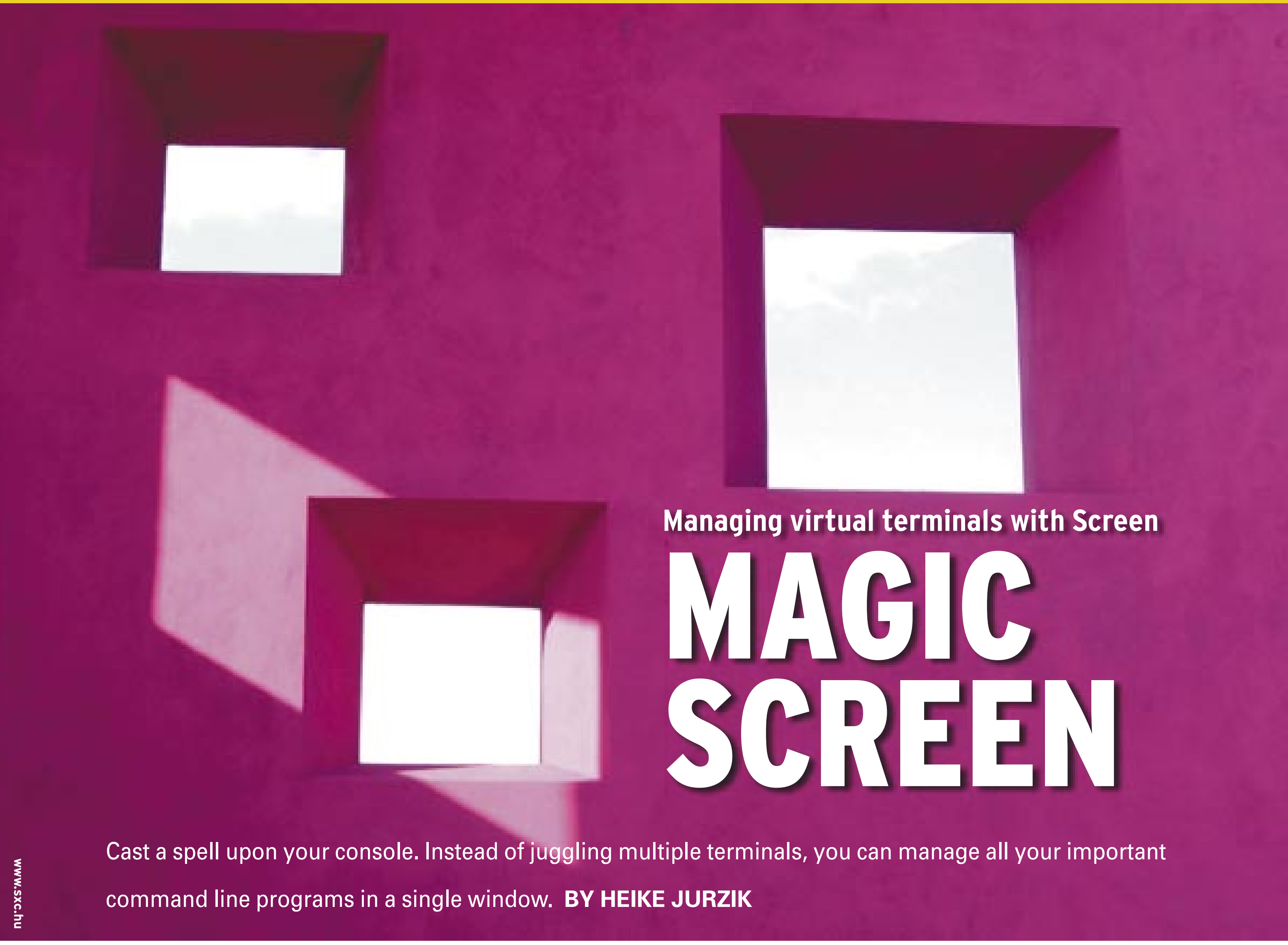


Managing virtual terminals with Screen

MAGIC SCREEN

Cast a spell upon your console. Instead of juggling multiple terminals, you can manage all your important command line programs in a single window. **BY HEIKE JURZIK**

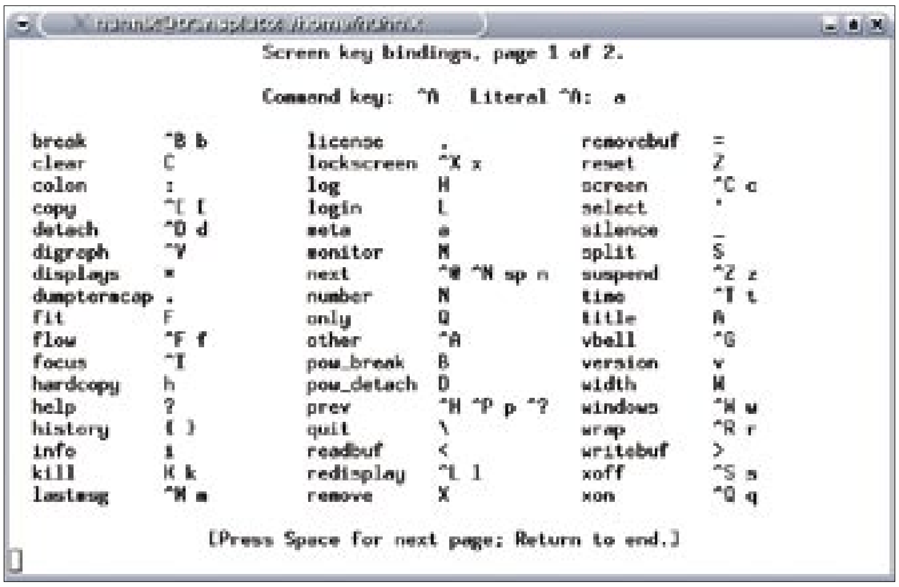
Screen lets you launch multiple virtual consoles in a single terminal. This gives users the ability to keep processes running on remote computers after logging off. And you can open up the screen to other users, letting them share your session.

The terminal applications on KDE and Gnome point the way: you can launch multiple tabs, just like in a browser, thus combining multiple console sessions in a single program window. If you do not use a GUI, or if you use SSH to work on a remote computer, there is no need to do without this convenience. The Screen tool will help you manage multiple virtual terminals, giving you the ability to keep processes running if you log off the remote machine. Screen even provides tools that let you share a console session with other users.

Setting the Screen

To launch the program, just enter *screen* at the command line. This brings up the splash screen that tells you the version number, and prompts you to press the space bar or Enter to continue. If you prefer to do without the welcome message, you can disable it using an entry in the Screen configuration file (see the "Setting Up Screen" box).

After launching your window juggler, things may not appear to have changed at first glance – the terminal will look like it always did. But your fingertips will tell you a different story, at least they will if you press [Ctrl] + [A] in Bash



```
                    Screen key bindings, page 1 of 2.

                    Command key:  ^A   Literal ^A:  a

 break        ^B b          license      ,              removebuf   =
 clear        C             lockscreen   ^X x           reset       Z
 colon        :             log          H              screen      ^C c
 copy         ^[ [          login        L              select      '
 detach       ^D d          meta         a              silence     _
 digraph      ^V            monitor      M              split       S
 displays     *             next         ^@ ^N sp n     suspend     ^Z z
 dumptermcap  .             number       N              time        ^T t
 fit          F             only         Q              title       A
 flow         ^F f          other        ^A             vbell       ^G
 focus        ^I            pow_break    B              version     v
 hardcopy     h             pow_detach   D              width       W
 help         ?             prev         ^H ^P p ^?     windows     ^W w
 history      { }           quit         \              wrap        ^R r
 info         i             readbuf      <              writebuf    >
 kill         K k           redisplay    ^L l           xoff        ^S s
 lastmsg      ^M m          remove       X              xon         ^Q q

               [Press Space for next page; Return to end.]
```

Figure 1: Pressing [Ctrl]+[A] and "?" displays a command reference.

to move to the start of the input line. Screen uses this shortcut to take you a collection of commands that control the program.

The "Screen Commands" table gives you an overview. Pressing [Ctrl] + [A] and "?" displays the online help (Figure 1). This command reference also tells you that you can restore [Ctrl] + [A]'s "normal" behavior by pressing [Ctrl] + [A], [A]. Pressing the space bar takes you to a second page of online help.

Opening and Closing Windows

As I mentioned earlier, you can quit Screen without stopping the processes running in Screen. The programs that are running inside Screen keep on doing whatever they are supposed to do, and if you revive Screen at some point later, you can check the resulting output. The keyboard shortcut for this is [Ctrl] + [A],[D]. To reestablish contact with Screen, just enter *screen -r* on the console.

You can also "detach" Screen remotely; you can move Screen to the background without having to interact. To do this, enter *screen -d*.

You can combine these parameters – to detach a session, and revitalize the session in the current console, just enter *screen -dr*.

If you have launched multiple Screen sessions, you have to be more specific about what you want to do. The program will let you know if you are not clear about what you tell it to do. As Listing 1 shows, Screen also shows what is going on in the background, and it gives you instructions on moving a session to the foreground. To revitalize the second entry in Listing 1, for example, you would do this:

```
screen -dr 9865.pts-11.xena
```

The output is similar if you enter *screen -ls*. The list shows you the process number, the terminal name, and the computer name, besides letting you know if Screen is running in the foreground (*attached*) or background (*detached*). From time to time you see screens with a *dead* status, but you can easily clean up the mess by entering *screen -wipe*.

Behind the Scenes

Screen lets you scroll back 100 lines by default to check out program output. If you need to increase the buffer, launch Screen with the *-h* parameter, specifying the new buffer size. The *screen -h 1000* command gives you 1000 lines to play with in copy/scroll mode ([Ctrl] + [A], [Esc]). To make this behavior the default, just add a matching entry to the program's configuration file (see the "Setting Up Screen" box).

Window Cleaning

If you have a large number of sessions running in Screen, you can leverage a practical feature to avoid losing track: as the "Screen Commands" box tells you, a shortcut lets you display a status line where you can assign names to individual sessions. To make this view permanent, enter the following command:

```
screen -X caption always
```

Various escape sequences add the ability to configure the status line to tell you things like the current user and hostnames, the date and time – in color if you like. To do so, add a combination of colors and details to the command line shown above (not forgetting to escape this in double quotes):

```
screen -X caption always "%{rw}➥
* %H * | $LOGNAME | %{bw}%c %D |➥
%{-}%-Lw%{rw}%50>%{rW}%n%f* %t %➥
{-}%+Lw%<"
```

Listing 1: Screen Output

```
$ screen -dr
There are several suitable
   screens on:
      9559.pts-11.xena
   (Detached)
      9865.pts-11.xena
   (Attached)
Type "screen [-d] -r
   [pid.]tty.host" to resume one
   of them.
```

Table 1: Screen Commands

[Ctrl]+[A],[C]	Opens another virtual terminal.
[Ctrl]+[D]	Closes a virtual terminal; if this is the only session running in Screen, this shortcut quits the program.
[Ctrl]+[A],[N] (or [Ctrl]+[A],space)	Switches to the next virtual terminal.
[Ctrl]+[A],[0]-[9]	Jumps to the first through tenth virtual terminal.
[Ctrl]+[A],[W]	Displays a status line at the bottom of the window for a couple of seconds; the asterisk tells you where you are right now.
[Ctrl]+[A], [Shift]+[A]	Lets the user assign a name to the virtual terminal (all sessions are titled "bash" by default).
[Ctrl]+[A], [Shift]+[2]	Displays a list of all active sessions; you can use the arrow keys to navigate to the required virtual terminal, and press [Enter] to open the selected window.
[Ctrl]+[A], [X]	Closes the window to prevent spying; you need to enter a password to unlock the window.
[Ctrl]+[A], [Esc]	Toggles to copy/scroll mode. You can use the arrow keys for navigation. To copy something to the clipboard, place the cursor at the start of the area you wish to copy, and press the space bar; the move the cursor to the end of the area, and press the space bar again. Pressing [Esc] cancels this action.
[Ctrl]+[A], "]"	Inserts the copied text.
[Ctrl]+[A], [D]	Sends Screen to the background; this only quits the program itself, that is, the processes running in Screen keep on running.
[Ctrl]+[A], [K]	Forcibly quits the current virtual terminal; you are prompted to confirm that you really want to kill the session.
[Ctrl]+[A], "\"	Forcibly quits Screen and all sessions running in Screen; a prompt is displayed (*Really quit and kill all your windows [y/n]*).

GLOSSARY

s bit: Besides read, write, and execute, Linux supports a couple of special permissions. The setuid/setgid bit replaces the standard execute permission (x) for the owner or group. This means that the program will always run with the permissions of the file owner or group, no matter who launches it.

The details in the curly brackets describe the colors, such as red on a gray background (*%{rw}*), blue on gray (*%{bw}*), or red on white (*%{rW}*). *%{-}* lets you enable the previous color scheme – the *STRING ESCAPES* section of the manpage gives you more detailed listings. The other variables represent the following:

- *$LOGNAME*: username
- *%H*: hostname
- *%c*: time – 24 hour format
- *%D*: day of week
- *%n*: number of virtual terminal
- *%f*: flag; for example, an asterisk denotes the current virtual terminal, and a minus sign the last session you visited
- *%t*: the session title, set using [Ctrl] + [A],[Shift] + [A]

Additionally, if the list of windows is too long, entering *%50>* tells Screen to curtail the list to display the following section in the middle of the line. Figure 2 shows the result of this window dressing.

Leaning Out the Window

With a few tweaks, you can use Screen as a teaching aid. To use Screen as a teaching tool, first assume the root user role and modify the access privileges for the program. Then one user launches Screen and allows other users access to the program.

Working as root, start by setting the **s bit** for the screen binary:

```
sudo chmod +s /usr/bin/screen
```

Then adjust the permissions for the */var/run/screen* directory, removing the group's write permission:

```
chmod g-w /var/run/screen
```

The user who will be giving visitors access to their screens, should now launch Screen, and type [Ctrl] + [A],[Shift] + [.] (":"), enable multi-user mode at the prompt, and enter one or more usernames as welcome guests:

```
: multiuser on
: addacl petrosilie
```

If you intend to invite more than one visitor to share your windows, just enter multiple comma-separated usernames of the visitors.

To visit the host user, a second user should now launch Screen, specifying the *-x* option, and entering and the host user's name, and the process number of the screen: *screen -x petronella/4552*.

Setting up Screen

Settings for the program are stored in a *~/.screenrc* file below your own home directory. You can use any text editor to modify the file. The file is not created by default, so you will need to create it. Most distributions store the global setup file in */etc/screenrc*; this file has a number of useful comments and examples to help you get started.

To use the file as a template, just copy it to your own home directory, as follows:

```
cp /etc/screenc ~/.screenrc
```

Listing 2 gives you a sample Screen configuration file with comments. ■

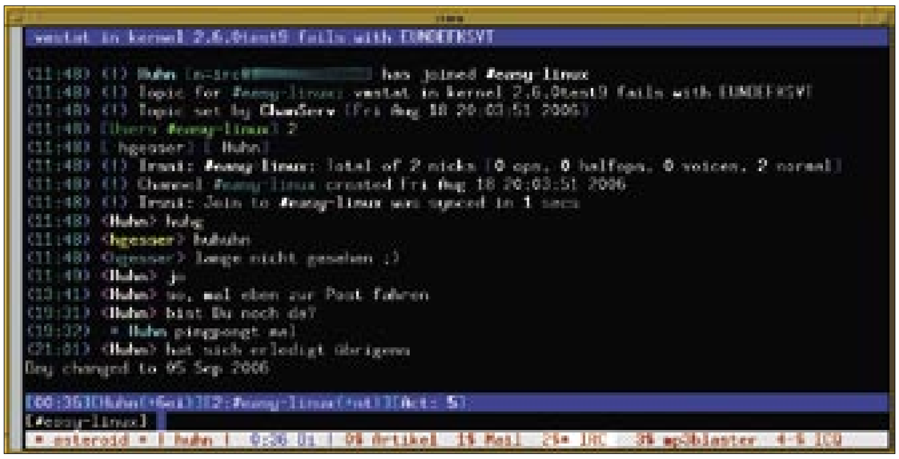
Listing 2: Screen Configuration File

```
# /etc/screenc ~/.screenrc
# ------------------------
# switch off splash screens on program launch:
startup_message off

# Set buffer, e.g., to 1000 lines:
defscrollback 1000

# Displays a colorful status line with the
# session name, date, time, etc.:
caption always "%{rw} * %H * | $LOGNAME | %{bw}%c %D |
   %{-}%-Lw%{rw}%50>%{rW}%n%f* %t %{-}%+Lw%<"

# [Ctrl]+[F] launches a new virtual terminal
# called "MAIL" and automatically establishes a
# SSH connection to "another.computer.com":
bindkey ^f screen -t MAIL ssh another.computer.com
```



```
vmstat in kernel 2.6.0test9 fails with EUNDEFKSYM
(11:48) (!) Huhn [n=irc[redacted]] has joined #easy-linux
(11:48) (!) Topic for #easy-linux: vmstat in kernel 2.6.0test9 fails with EUNDEFKSYM
(11:48) (!) Topic set by ChanServ (Fri Aug 18 20:03:51 2006)
(11:48) [Users #easy-linux] 2
(11:48) [ hgesser] [ Huhn]
(11:48) (!) Irssi: #easy-linux: Total of 2 nicks [0 ops, 0 halfops, 0 voices, 2 normal]
(11:48) (!) Channel #easy-linux created Fri Aug 18 20:03:51 2006
(11:48) (!) Irssi: Join to #easy-linux was synced in 1 secs
(11:48) <Huhn> huhu
(11:48) <hgesser> huhuhn
(11:48) <hgesser> lange nicht gesehen ;)
(11:49) <Huhn> ja
(13:41) <Huhn> so, mal eben zur Post fahren
(19:31) <Huhn> bist Du noch da?
(19:32)  * Huhn pinggongt mal
(21:01) <Huhn> hat sich erledigt übrigens
Day changed to 05 Sep 2006
[00:36][Huhn(+Gei)][2:#easy-linux(+nt)][Act: 5]
[#easy-linux]
 * asteroid * | huhn |  0:36 01 | 0$ Artikel  1$ Mail  2$* IRC  3$ mp3blaster  4-$ ICQ
```

Figure 2: Some window dressing makes working with Screen really convenient.

THE AUTHOR

Heike Jurzik studied German, Computer Science and English at the University of Cologne, Germany. She discovered Linux in 1996 and has been fascinated with the scope of the Linux command line ever since. In her leisure time, you might find Heike hanging out at Irish folk sessions or visiting Ireland.