10 Screen Command Examples to Manage Linux Terminals

By Pungki Arianto[1] Under: Linux Commands[2] On: October 9, 2013

Download Your Free eBooks NOW - 10 Free Linux eBooks for Administrators[3]

Screen is a full-screen software program that can be used to multiplexes a physical console between several processes (typically interactive shells). It offers a user to open several separate terminal instances inside a one single terminal window manager.

The screen application is very useful, if you are dealing with multiple programs from a command line interface and for separating programs from the terminal shell. It also allows you to share your sessions with others users and detach/attach terminal sessions.

[4]

Screen Command Examples

On my Ubuntu 10.04 Server Edition, **Screen** has been installed by default. But, in Linux Mint does not have screen installed by default, I need to install it first using **apt-get command** before using it. Please follow your distribution installation procedure to install screen.

```
# apt-get install screen (On Debian based Systems)
# yum install screen (On RedHat based Systems)
```

Actually, Screen is a very good command in Linux which is hidden inside hundreds of Linux commands. Let's start to see the function of Screen.

Start screen for the first time

Just type screen at the command prompt. Then the screen will show with interface exactly as the command prompt.

```
pungki@mint ~ $ screen
```

Show screen parameter

When you enter the screen, you can do all your work as you are in the normal CLI environment. But since the screen is an application, so it have command or parameters.

Type “**Ctrl-A**” and “**?**” without quotes. Then you will see all commands or parameters on screen.

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

                                    Command key:  ^A   Literal ^A:  a

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

^]  paste .
"   windowlist -b
-   select -
0   select 0
1   select 1
2   select 2
3   select 3
4   select 4
5   select 5
6   select 6
7   select 7
8   select 8
9   select 9
I   login on
O   login off
]   paste .
```

To get out of the help screen, you can press “space-bar” button or “**Enter**“. (Please note that all shortcuts which use “**Ctrl-A**” is done without quotes).

Detach the screen

One of the advantages of screen that is you can detach it. Then, you can restore it without losing anything you have done on the screen. Here’s the sample scenario:

You are in the middle of **SSH-on** your server. Let’s say that you are downloading **400MB** patch for your system using wget command[5].

The download process is estimated to take **2 hours** long. If you disconnect the **SSH** session, or suddenly the connection lost by accident, then the download process will stop. You have to start from the beginning again. To avoid that, we can use screen and detach it.

Take a look at this command. First, you have to enter the screen.

```
pungki@mint ~ $ screen
```

Then you can do the download process. For examples on my Linux Mint, I am upgrading my **dpkg** package using **apt-get** command.

```
pungki@mint ~ $ sudo apt-get install dpkg
```

Sample Output

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  dpkg
1 upgraded, 0 newly installed, 0 to remove and 1146 not upgraded.
Need to get 2,583 kB of archives.
After this operation, 127 kB of additional disk space will be used.
Get:1 http://debian.linuxmint.com/latest/ testing/main dpkg i386 1.16.10 [2,583 kB]
47% [1 dpkg 1,625 kB/2,583 kB 47%]                                    14,7 kB/s
```

While downloading in progress, you can press "**Ctrl-A**" and "**d**". You will not see anything when you press those buttons. The output will be like this:

```
[detached from 5561.pts-0.mint]
pungki@mint ~ $
```

Re-attach the screen

After you detach the screen, let say you are disconnecting your **SSH** session and going home. In your home, you start to **SSH** again to your server and you want to see the progress of your download process. To do that, you need to restore the screen. You can run this command:

```
pungki@mint ~ $ screen -r
```

And you will see that the process you left is still running.

When you have more than **1 screen** session, you need to type the screen session **ID**. Use screen **-ls** to see how many screen are available.

```
pungki@mint ~ $ screen -ls
```

Sample Output

```
pungki@mint ~ $ screen -ls
There are screens on:
        7849.pts-0.mint (10/06/2013 01:50:45 PM)        (Detached)
        5561.pts-0.mint (10/06/2013 11:12:05 AM)        (Detached)
2 Sockets in /var/run/screen/S-pungki
```

If you want to restore screen **7849.pts-0**.mint, then type this command.

```
pungki@mint ~ $ screen -r 7849
```

Using Multiple Screen

When you need more than **1 screen** to do your job, is it possible? Yes it is. You can run multiple screen window at the same time. There are 2 (two) ways to do it.

First, you can detach the first screen and the run another screen on the real terminal. Second, you do nested screen.

Switching between screens

When you do nested screen, you can switch between screen using command "**Ctrl-A**" and "**n**". It will be move to the next screen. When you need to go to the previous screen, just press "**Ctrl-A**" and "**p**".

To create a new screen window, just press "**Ctrl-A**" and "**c**".

Logging whatever you do

Sometimes it is important to **record** what you have done while you are in the console. Let say you are a **Linux Administrator** who manage a lot of Linux servers.

With this screen logging, you don't need to write down every single command that you have done. To activate screen logging function, just press "**Ctrl-A**" and "**H**". (Please be careful, we use capital '**H**' letter. Using non capital '**h**', will only create a screenshot of screen in another file named hardcopy).

At the bottom left of the screen, there will be a notification that tells you like: Creating logfile "**screenlog.0**". You will find **screenlog.0** file in your home directory.

This feature will append everything you do while you are in the screen window. To close screen to log running activity, press "**Ctrl-A**" and "**H**" again.

Another way to activate logging feature, you can add the parameter "**-L**" when the first time running screen. The command will be like this.

```
pungki@mint ~ $ screen -L
```

Lock screen

Screen also have shortcut to **lock** the screen. You can press "**Ctrl-A**" and "**x**" shortcut to lock the screen. This is handy if you want to lock your screen quickly. Here's a sample output of lock screen after you press the shortcut.

```
Screen used by Pungki Arianto  on mint.
Password:
```

You can use your Linux password to unlock it.

Add password to lock screen

For security reason, you may want to put the **password** to your screen session. A Password will be asked whenever you want to **re-attach** the screen. This password is different with **Lock Screen** mechanism above.

To make your screen password protected, you can edit "**$HOME/.screenrc**" file. If the file doesn't exist, you can create it manually. The syntax will be like this.

```
password crypt_password
```

To create "**crypt_password**" above, you can use "**mkpasswd**" command on Linux. Here's the command with password "**pungki123**".

```
pungki@mint ~ $ mkpasswd pungki123
l2BIBzvIeQNOs
```

mkpasswd will generate a hash password as shown above. Once you get the hash password, you can copy it into your "**.screenrc**" file and save it. So the "**.screenrc**" file will be like this.

```
password l2BIBzvIeQNOs
```

Next time you run screen and detach it, password will be asked when you try to **re-attach** it, as shown below:

```
pungki@mint ~ $ screen -r 5741
Screen password:
```

Type your password, which is "**pungki123**" and the screen will **re-attach** again.

After you implement this screen password and you press "**Ctrl-A**" and "**x**" , then the output will be like this.

```
Screen used by Pungki Arianto  on mint.
Password:
Screen password:
```

A Password will be asked to you **twice**. First password is your **Linux password**, and the second password is the password that you put in your **.screenrc** file.

Leaving Screen

There are **2** (two) ways to leaving the screen. First, we are using "**Ctrl-A**" and "**d**" to detach the screen. Second, we can use the exit command to terminating screen. You also can use "**Ctrl-A**" and "**K**" to kill the screen.

That's some of screen usage on daily basis. There are still a lot of features inside the **screen command**. You may see **screen man page** for more detail.

Pungki Arianto

Currently I am a Linux/Unix administrator. But I also play Windows both in server and desktop area. Interested in information technology, information security and writing.

Linux Services & Free WordPress Setup

Our post is simply '**DIY**' aka '**Do It Yourself**, still you may find difficulties and want us to help you out. We offer wide range of **Linux** and **Web Hosting Solutions** at fair minimum rates. Please submit your orders by Clicking Here[6].

1. http://www.tecmint.com/author/pungkiarianto/
2. http://www.tecmint.com/category/linux-commands/
3. http://www.tecmint.com/10-useful-free-linux-ebooks-for-newbies-and-administrators/
4. http://www.tecmint.com/wp-content/uploads/2013/10/Screen-Commands.png
5. http://www.tecmint.com/10-wget-command-examples-in-linux/
6. http://www.tecmint.com/hire-us/