

# Using Screen on UNIX for Terminal Sharing

Weiping He

## 1 Document Summary

This document describes how to screen to share terminals between users.

## 2 Introduction

Screen is a terminal multiplexer: it takes many different running processes and manages which of them gets displayed to the user. Think of it as a window manager for your console or terminal emulator. With screen, you can have interactive processes running on your home computer and connect to them from anywhere else to check on their progress.

What sorts of programs are good for running in screen?

- Build source code/giant compiles;
- Terminal sharing;
- Log watching;
- Command line programs;
- Education/demo.

## 3 Requirements

The screen package is installed. The path is at:

`/usr/bin/screen`, `/bin/screen`, or `/usr/local/bin/screen`

The Set-user-ID-on-execution permission is set by this command:

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

The permission of `/var/run/screen` is set to 755 by this command:

```
chmod 755 /var/run/screen
```

The above two commands need to be run as root.

## 4 Steps

- A) The terminal owner starts *screen* in a local xterm, via this command

```
screen -S SessionName
```

The `-S` switch gives the session a name, which makes multiple screen sessions easier to manage.

For example:

```
rh-prague:/home/weiping> screen -S mytest
```

- B) The owner allows multiuser access in the screen session via the command ***Ctrl-a*** : *multiuser on* (all screen commands start with the screen escape sequence, ***Ctrl-a***). To avoid typine this command every time, you can put the folling line into the `.screenrc` file

```
multiuser on
```

- C) Next the owner grants permission to the guest user to access the screen session with ***Ctrl-a*** : *acladd guest* where *guest* is the guest user login name.
- D) The guest user can see the list of screen session owned by user owner by specifying the user whose session list is requested:

```
screen -list username/
```

For example

```
rh-prague:/home/tester> screen -list weiping/
```

There is a suitable screen on:

```
23194.mytest      (Multi, attached)
```

```
1 Socket in /var/run/screen/S-weiping.
```

- E) The guest can now connect to the owner's screen session. The syntax to connect to another user's screen session is

```
screen -x username/SessionName
```

For example

```
rh-prague:/home/tester> screen -x weiping/mytest
```

- F) The owner can also connect to this screen session from another terminal by using this command:

```
screen -x username/SessionName
```

- G) The screen logging can be turned on or off at any time with ***Ctrl-a H***, or you can use the ***-L*** switch when starting screen to enable it by default. The log file is written to the current directory under the name *screenlog.n*, where *n* is incremented by one for each new log.

## 5 Common problems

- If you get a “*chmod /dev/pts/xx: Operation not permitted*” error, it may be because you have *su* on a tty you do not own. This does not work because you have to own the tty for *screen* to work.
- If you have a */tmp/uscreen* file instead of a */tmp/screen/S-xxxxxxx* file, put *multiuser on* in your *.screenrc* file, and try again.

## 6 References

Multiple users of a terminal

<http://www.pixelbeat.org/docs/screen/>

Using screen for remote interaction

<http://www.linux.com/archive/feed/56443>

GNU screen

<http://aperiodic.net/screen/multiuser>

## 7 Appendix

### 7.1 screen command tips

- ***CTRL-a*** gets into command mode. like ESC in vi
- ***CTRL-a + CTRL-a*** cycles through the screens
- ***CTRL-a + "*** shows list of screens. use arrow or j/k keys to select
- ***CTRL-a + : + quit*** closes session
- ***CTRL-a + d*** - exits screen but does not kill session. It is still running in the background waiting to be reattached with "screen -r"

### 7.2 screen command under Ctrl-a for multiuser sharing

- **acladd** - Adds users with full permission to all windows.
- **aclchg** - Adds users with more flexible permissions or changes the permissions on an existing user.
- **acldel** - Removes a user from screen's knowledge.
- **aclgrp** - Adds a user to a group or just describes user's group membership.
- **aclumask** - Sets default permissions for windows not yet created.
- **defescape** - Like escape, but sets the command character for all users.
- **defwritelock** - Sets the default writelock setting for new windows.
- **multiuser** - Enables or disables multiuser mode.
- **su** - Operate as a different user.
- **writelock** - Sets writelock mode for current window.