

time-saving tricks

on the command line

gimme a new window

\$ cd /usual/path/to/work/on/projectx

\$ screen -R projectx

\$ tail -f my.log

Hm... what's in config.sh?

Gimme a new window!

ctrl-a c

\$ less config.sh

gimme another window

\$ tail -f /some/log

ctrl-a c

\$ less /some/file

ctrl-a c

\$ man some_doc

ctrl-a c

gimme a new window

gimme a new window

gimme a new window

gimme a new window

gimme a new window

* ctrl-a is the *command key*

* c stands for *create window*

next prev window

Next window

Previous window

Window list

ctrl-a n

ctrl-a p

ctrl-a w

Screen is a full-screen window manager between several processes. Each virtual terminal provides the user with a full-screen environment. In addition, several control functions are available (see X3.64) and ISO 2022 standards (for multiple character sets). The user can move between each virtual terminal and a control region between windows.

When screen is called, it creates a window (or the specified command) and you can use the program as you normally would.

```
0$ tail 1-$ less 2*$ man 1
```

```
h2 {
    background-color: #A0C0F0;
    width: 98%;
    padding: 3px 6px;
}
table {
    text-align: right;
    margin-left: 30px;
    background-color: #D0E0F0;
    border-collapse: collapse;
}
```

```
0-$ tail 1*$ less 2$ man
```

```
173.44.37.234 - - [12/May/2013:11:
ws7-host-and-debian-guest-with-vi
gyerik.com/sharing-files-between-v
" "Mozilla/5.0 (Windows; U; Window
refox/3.5.3 GTB5"
77.226.248.32 - - [12/May/2013:11:
530 "-" "Mozilla/3.0 (co
77.226.248.32 - - [12/May/2013:11:
3805 "-" "Mozilla/3.0 (compatibl
208.115.111.75 - - [12/May/2013:11:
/ HTTP/1.1" 200 32169 "-" "Mozilla
.com)"
```

detach/reattach

I want to get out!

detach

ctrl-a d

I want to go back!

reattach

\$ screen -R label

Terminal crashed!

detach and reattach

\$ screen -R label -D

My sessions?

\$ screen -ls

An iceberg floating in a blue ocean under a blue sky with light clouds. The visible tip of the iceberg is small and jagged, while the submerged portion is much larger and more complex in shape. The text 'ctrl-a ?' is written in a bold, red, sans-serif font across the middle of the submerged part of the iceberg.

ctrl-a ?

search history with **ctrl-r**

*...I want that **for** loop I
used a while ago...*

```
$ history | grep for
362  for i in *; do mv "$i"
-t us-ascii//translit)" -v; do
364  for i in *; do mv "$i"
us-ascii//translit)" -v; done
```

ctrl-r and type "for"

```
(reverse-i-search)
`for': history |grep
for
```

ctrl-r again...

```
(reverse-i-search)`for': for i in *; do
mv "$i" "$(echo $i | iconv -f utf-8 -t
us-ascii//translit)" -v; done
```

ctrl-r again...

```
(reverse-i-search)`for': for i in *; do
mv "$i" "$(echo $i | iconv -f iso8859-1
-t us-ascii//translit)" -v; done
```

editing quickly

Delete last word **ctrl-w**

Delete until line end **ctrl-k**

Paste deleted stuff **ctrl-y**

Cancel entire line **ctrl-c**

moving quickly

Jump to line start

ctrl-a

ctrl-a a inside screen

Jump to line end

ctrl-e

An iceberg floating in a blue ocean under a blue sky with light clouds. The tip of the iceberg is above the water, while the much larger base is submerged. The text is overlaid on the submerged part of the iceberg.

man bash

Search for: "READLINE"

"Commands for Moving"

"Commands for Changing Text"

moving quickly in less

Search for "error"

/error

Next match

n

Previous match

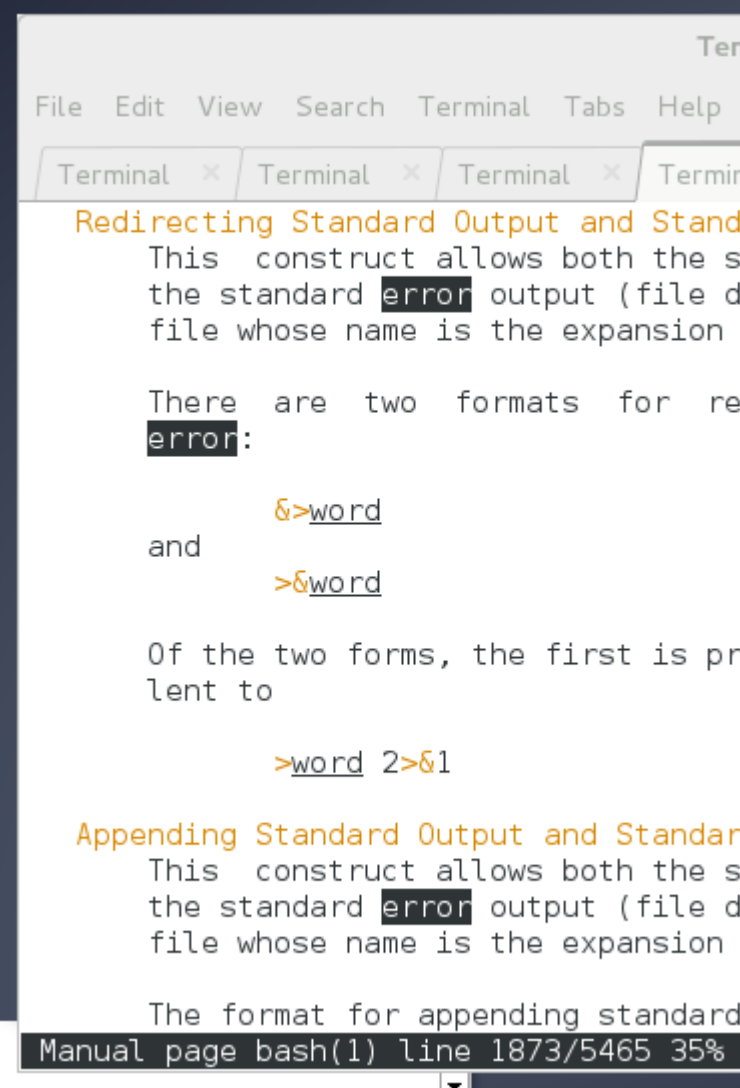
N

Jump to first line

g

Jump to last line

G



The screenshot shows a terminal window with the 'less' command-line pager. The title bar indicates 'Terminal' and 'Terminal x'. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', 'Tabs', and 'Help'. The content area displays a manual page for 'Redirecting Standard Output and Standard Error'. The text is as follows:

```
Redirecting Standard Output and Standard Error
This construct allows both the standard
the standard error output (file descriptor
file whose name is the expansion of the
file name) to be redirected to a file or
device.

There are two formats for redirecting
error:

    &>word
and
    >&word

Of the two forms, the first is preferred
to the second.

    >word 2>&1

Appending Standard Output and Standard Error
This construct allows both the standard
the standard error output (file descriptor
file whose name is the expansion of the
file name) to be appended to a file or
device.

The format for appending standard output
and standard error is:

    >&word &>&word
```

The status bar at the bottom shows 'Manual page bash(1) line 1873/5465 35%'.

moving quickly in less

Mark line in register "a", "b"

ma mb

Jump to register "a", "b"

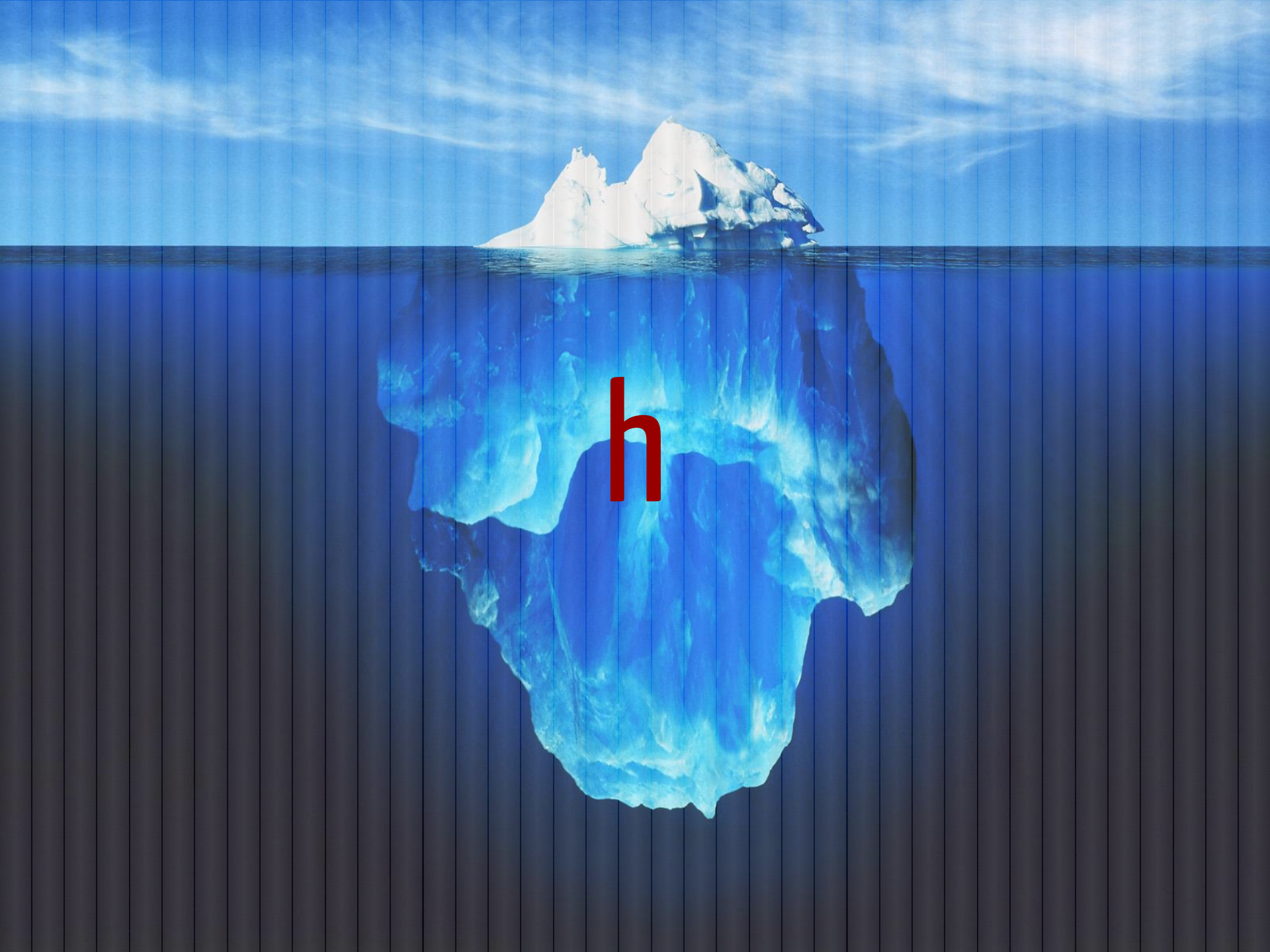
'a 'b

Toggle case sensitive search

-i

Go to next file

:n



mail it to me

mail me the relevant stuff

```
$ grep stuff /var/log/messages | mailx -s  
"stuff from logs" me@example.com
```

mail me the whole file

```
$ uuencode /var/log/messages messages.  
txt | mailx -s "system logs on $HOST"  
me@example.com
```

mail me when finished

after the loooong task, mail me

```
$ rsync --progress /disk1/iso/*  
/disk2/iso/; ls -lha /disk2/iso/ |  
mailx -s "copy isos done"  
me@example.com
```

mail me when file appears

infinite loop until file appears and then mail me

```
$ while :; do date; test -f file.txt &&  
break; sleep 300; done; ls -lh file.  
txt | mailx -s "file appeared"  
me@example.com
```


mail me when log matches

infinite loop until matching line appears and then mail me

```
$ while :; do date; grep pattern  
/path/to/log && break; sleep 300;  
done; { date; grep -B50 -A50  
pattern /path/to/log; } | mailx -s  
"log snippet" me@example.com
```

encrypt and decrypt simply

```
$ openssl des3 < secret.bin > secret.bin.des3
```

```
enter des-ede3-cbc encryption password:
```

```
Verifying - enter des-ede3-cbc encryption password:
```

```
$ openssl des3 -d < secret.bin.des3 > decrypted.bin
```

```
enter des-ede3-cbc decryption password:
```

```
$ cmp secret.bin decrypted.bin
```

```
$
```

```
# empty output means the two files are identical
```