

Using the Shell Effectively

with Readline and History

TLUG
2009-09-12
Zev Blut

I am not an expert!

Recently humbled

Let's learn a few tricks

Characters to know

C- : Control

Have you have remapped Capslock to Control?

M- : Meta

Typically your ALT & ESC keys

Basic movement

* Emacs mode

C-a

Go to beginning of line

```
> echo cat dog|
```

C-a

Go to beginning of line

```
> | echo cat dog
```

C-e

Goto to end of line

```
> | echo cat dog
```

C-e

Goto to end of line

```
> echo cat dog|
```

C-f

Forward character

```
> echo | cat dog
```

C-f

Forward character

```
> echo c|at dog
```

M-f

Forward word

```
> echo | cat dog
```

M-f

Forward word

```
> echo cat | dog
```


C-b

Backward character

```
> echo c|at dog
```

C-b

Backward character

```
> echo | cat dog
```

M-b

Backward word

```
> echo cat | dog
```

M-b

Backward word

```
> echo | cat dog
```

C- and M-
are often similar :
one for characters
one for words

When the same,
I will just talk about
word commands

M-

M-del / C-w

Delete backward word

```
> echo cat | dog
```

M-del / C-w

Delete backward word

```
> echo | dog
```


* `/var/log/foo/bar.txt`

What is the command to just
delete up to / ?

C-u

Delete line behind prompt

```
> echo cat | dog
```

C-u

Delete line behind prompt

```
> | dog
```

C-k

Cut line after prompt

```
> echo c | at dog
```

C-k

Cut line after prompt

```
> echo c |
```

C-y

Yank / paste last cut

```
> echo c |
```

C-y

Yank / paste last cut

```
> echo cat dog |
```

M-y

Cycle through kill-ring (i.e. cuts)

```
> echo cat dog |
```

* You must first use C-y to then use M-y

M-y

Cycle through kill-ring (i.e. cuts)

```
> echo car bike|
```

M-t

Transpose word

```
> echo cat dog
```

M-t

Transpose word

```
> echo dog cat |
```

M-u

Upcase word

```
> echo cat | dog
```

M-u

Upcase word

```
> echo cat DOG |
```

M-I

Lowercase word

```
> echo cat | DOG
```

M-I

Lowercase word

```
> echo cat dog |
```

M-c

Capitalize word

```
> echo cat | dog
```


M-c

Capitalize word

```
> echo cat Dog |
```

C-I

Clear screen and re-centers at top

```
> echo foo bar  
> echo cat | dog
```

C-I

Clear screen and re-centers at top

```
> echo cat | dog
```

M-r

Revert line

```
> echo fo|o bar
```

M-r

Revert line

>

|

C-_, C-x C-u

Undo

```
> echo | bar
```

C-_, C-x C-u

Undo

```
> echo foo | bar
```

M-#

Comment out line

```
> echo foo | bar
```


M-#

Comment out line

```
> #echo foo bar
```

```
> |
```

C-] char

Move to first forward instance of char

```
> echo | foo bar
```

C-] a

Move to first forward instance of char

```
> echo foo b|ar
```

M-C-[char

Move to first backward instance of char

```
> echo foo b|ar
```

M-C-[f

Move to first backward instance of char

```
> echo | foo bar
```

Navigating History

C-p , up arrow

Up history

```
> ls /var/log/nginx/  
> echo |
```

C-p , up arrow

Up history

```
> ls /var/log/nginx/  
> ls /var/log/nginx/ |
```


C-n , down arrow

Down history

```
> echo |
```

C-n , down arrow

Down history

Better to try this one out

The one command you
must learn!

C-r

Reverse search

```
> git status  
> ls /var/log/nginx/  
> echo |
```

C-r git

Reverse search

```
> git status  
> ls /var/log/nginx/  
> git status |
```

C-s

Forward search

```
> git status  
> ls /var/log/nginx/  
> echo |
```

C-s

Forward search

Tricky to show

C-g

Abort search return to original line

```
> git status  
> ls /var/log/nginx/  
> echo |  
> git status |
```


C-g

Abort search return to original line

```
> git status  
> ls /var/log/nginx/  
> echo |
```

M-<

Start of history

```
> touch hoge.txt  
> git status  
> ls /var/log/nginx/  
> |
```

M-<

Start of history

```
> touch hoge.txt  
> git status  
> ls /var/log/nginx/  
> touch hoge.txt|
```

M->

End of history

```
> touch hoge.txt  
> git status  
> ls /var/log/nginx/  
> |
```

M->

End of history

```
> touch hoge.txt  
> git status  
> ls /var/log/nginx/  
> ls /var/log/nginx/ |
```

M-., M-

Yank last arg

```
> ls /var/log/nginx/  
> echo |
```

* Future slide alert (same as !\$)

M-., M-

Yank last arg

```
> ls /var/log/nginx/  
> echo /var/log/nginx|
```

* Future slide alert (same as !\$)

M-num M-C-y

Yank nth arg of last command
negative n goes backwards

```
> tail foo bar baz  
> echo |
```

* Future slide alert (same as **!!:n**)

M-2 M-C-y

Yank nth arg of last command
negative n goes backwards

```
> tail foo bar baz  
> echo bar|
```

* Future slide alert (same as **!!:n**)

!!History

Event Designators

!!

execute last command

```
> tail foo.txt  
> !! |
```



execute last command

```
> tail foo.txt  
> !!  
tail foo.txt  
> |
```

!\$

Insert last arg

```
> tail foo.txt  
> head !$ |
```

!\$

Insert last arg

```
> tail foo.txt  
> head !$  
head foo.txt  
> |
```

!num

Execute line num from history

```
>24 touch hoge.txt  
>25 git status  
>26 ls /var/log/nginx/  
> !24 |
```

* **!-num** will execute num commands ago

!24

Execute line num from history

```
>24 touch hoge.txt  
>..  
> !24  
touch hoge.txt  
> |
```

* **!-num** will execute num commands ago

!string

Execute first command that starts with string from history

```
> touch hoge.txt  
> git status  
> ls /var/log/nginx/  
> !g |
```

!g

Execute first command that starts with string from history

```
> touch hoge.txt  
> git status  
> ls /var/log/nginx/  
> !g  
git status  
> |
```

!string[?]

Execute first command from history that has string in it

```
> touch hoge.txt  
> git status  
> ls /var/log/nginx/  
> !?ge? |
```

* Like C-r

!ge?

Execute first command from history that has string in it

```
> touch hoge.txt  
> git status  
> ls /var/log/nginx/  
> !ge?  
touch hoge.txt  
> |
```

* Like C-r

`^abc^xyz^`

Run last command but replace abc for xyz

```
> echo foo bar  
> ^foo^hoge^|
```

* Same as `!!:s/abc/xyz/`

^foo^hoge^

Run last command but replace abc for xyz

```
> echo foo bar  
> ^foo^hoge^  
echo hoge bar  
> |
```

* Same as **!!:s/abc/xyz/**

!#

Repeat current line

```
> echo hoge !# |
```


!#

Repeat current line

```
> echo hoge !#  
echo hoge echo hoge  
> |
```

Word Designators

•^
•

Use first word

```
> ls /a/b.txt /c/d/e.log  
> tail !!:^|
```

!!:^

Use first word

```
> ls /a/b.txt /c/d/e.log  
> tail !!!:^  
tail /a/b.txt  
> |
```

:\$

Use last word

```
> ls /a/b.txt /c/d/e.log  
> tail !!:$|
```

!!: \$

Use last word

```
> ls /a/b.txt /c/d/e.log  
> tail !!: $  
tail /c/d/e.log  
> |
```

:n

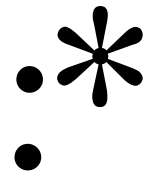
Use nth word

```
> ls /a/b.txt /c/d.log e.c  
> grep 404 !!:2|
```

!!:2

Use nth word

```
> ls /a/b.txt /c/d.log e.c  
> grep 404 !!:2  
grep 404 /c/d.log  
> |
```

Use all args minus the command

```
> ls /a/b.txt /c/d.log e.c  
> rm !!:* |
```

!!.*

Use all args minus the command

```
> ls /a/b.txt /c/d.log e.c  
> rm !!.*  
rm /a/b.txt /c/d.log e.c  
> |
```

Modifiers

:p

Don't execute just print

```
> ls /a/b.txt /c/d.log e.c  
> rm !!:*:p|
```

!!:*:p

Don't execute just print

```
> ls /a/b.txt /c/d.log e.c  
> rm !!:*:p  
# rm /a/b.txt /c/d.log e.c  
> |
```

:s/abc/xyz/

Substitute abc for xyz

```
> ls /a/b.txt /c/d.log e.c  
> tail !!:2:s/d/q/ |
```

!!:2:s/d/q/

Substitute abc for xyz

```
> ls /a/b.txt /c/d.log e.c  
> tail !!:2:s/d/q/  
tail /c/q.log  
> |
```

Can be chained

!-2:^:p:s/f/m/

just print 2nd to last command first arg
substituting f for m

```
> echo foo bar baz  
> ls /var/log/nginx/  
> !-2:^:p:s/f/m/ |
```

!-2:^:p:s/f/m/

just print 2nd to last command first arg
substituting f for m

```
> echo foo bar baz
> ls /var/log/nginx/
> rm !-2:^:p:s/f/m/
# rm moo
> |
```

Extras!

Completion

**detailed completion is
another talk**

Tab

Complete at point

```
> echo f |
```

Tab

Complete at point

```
> echo foo|
```

M-?

Show all possible matching completions

```
> ls b|
```


M-?

Show all possible matching completions

```
> ls b|  
bar  
baz
```

M_*

Add all possible matching completions

```
> ls b|
```

M-*

Add all possible matching completions

```
> ls bar baz |
```

C-x e

Edit current line in EDITOR

```
> for i in *.txt; do \  
>> cp ${i} backups/ \  
>> done |
```

Create a macro

C-x (

Start macro

>



C-x)

Stop macro

```
> cd /var/log/  
> tail -n 50 messages  
> pwd |
```

C-x e

Execute last macro

>



C-x e

Execute last macro

```
> cd /var/log/  
> tail -n 50 messages  
> pwd |
```

Customizing keys

Bind

- bind -p
- bind -v
- bind -l

.inputrc

ZSH

M-h

Call for help on current command

```
> grep -v 200 |
```

M-h

Call for help on current command

```
> grep -v 200 |  
man grep
```

bindkey "^%" replace-string

Docs for Readline

- <http://cnswww.cns.cwru.edu/php/chet/readline/rltop.html>
- First hit if you search for readline

Extra notes from members at talk

- rlwrap
 - to wrap readline on apps that don't use it
- kill command from history.
 - i.e. accidental password not using
HISTSIZE=0