screen

demo

Theory

Context-switches = Costly

Text = Cheap & Portable

Attention = Expensive

Avoid Hardware Context Switches

Huffman code your keybindings

```
bindkey "\351" next
bindkey "\357" prev
```

Updating titles automatically

```
shell /bin/bash
shelltitle “$ |sh”
```

```
hardstatus on
hardstatus alwayslastline
hardstatus string "%{kw}%w"
```

SSH Integration

```
screen_show_in_shelltitle() {
        printf "\ek$*\e\134"
}

screen_exec() {
if [[ ! -z "$STY" ]]; then
 display=$(echo $1 | sed -e 's/\..*//' )
 screen_show_in_shelltitle ${display}
fi
shift
command $* }
```

SSH Integration

```
ssh()    { screen_run $1 ssh    $@ ; }
telnet() { screen_run $1 telnet $@ ; }
exec()   { screen_run $1 exec   $@ ; }
sudo()   { screen_run $1 sudo   $@ ; }
```

SSH Integration

setenv SSH_AUTH_SOCK
$HOME/.ssh/agent

ssh-agent -a $HOME/.ssh/agent screen

Text wants to be Free

defscrollback=2048

CTRL-a ESC = copy mode
CTRL-a] = paste

Vim Integration

```
bash$  screen -p ${title} -X readbuf ${file}
```

```
bash$  screen -p ${title} -X paste .
```

Vim Integration

```
function! ScreenExec(window) range
  exe writefile( getline(a:firstline, a:lastline),
       “/tmp/buffer”)
  let cmd = “screen -p “ . a:window
  system(cmd . “ -X readbuf /tmp/buffer”)
  system(cmd . “ -X paste . “)
endfun

command! -range -nargs=* FeedToScreen
<line1>,<line2> call ScreenExec(<q-args>)
```

Mutt Integration

```
#!/bin/sh
# set editor=”this script.sh”
screen -X select vim
vim –servername local –remote-wait $1
screen -X select mutt
```

Monitoring

CTRL-a m

Macros

CTRL-a :exec !! macro.pl

```
setenv PATH
        $PATH:$HOME/etc/screen/macros
```

Torrents

```
screen -S bt -md transmissioncli
        -f 'kill $PPID' -u 10 ${torrent} &
```

the detach demo