Random Tips/Tricks for Linux Users

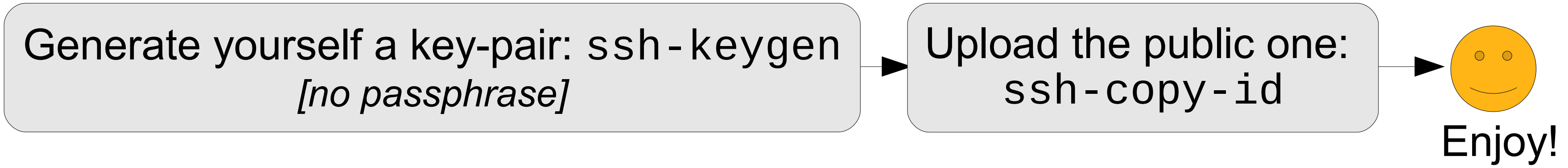
Mesut Ali Ergin
ergin@winlab.rutgers.edu
June 30, 2009

ssh without passwords!

or as password-less as it should get...

Purpose: `Minimize the hassle of typing your password every-time you login to a remote server`

RISKY:

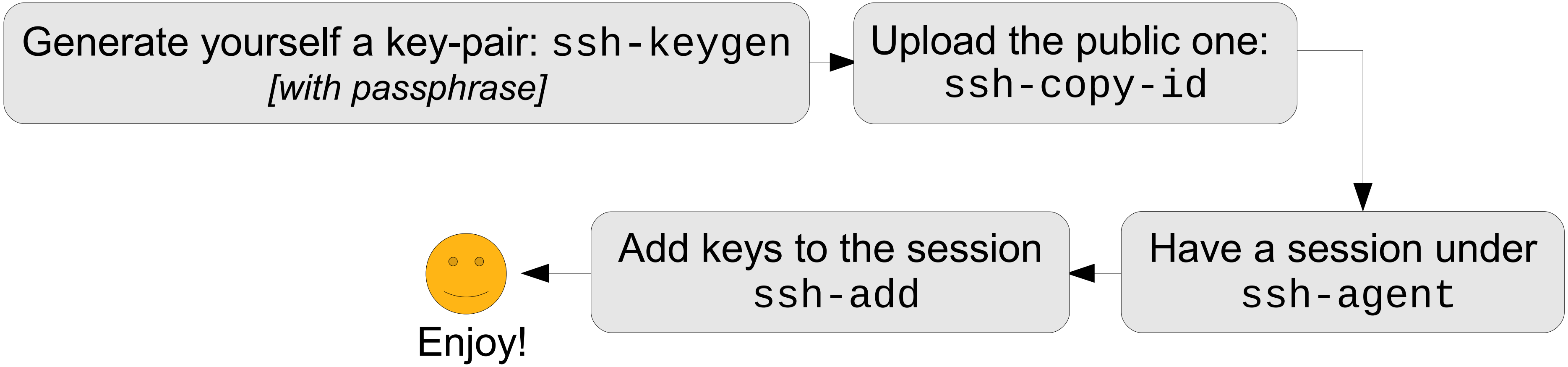


Generate yourself a key-pair: `ssh-keygen`
[no passphrase]

Upload the public one:
`ssh-copy-id`

Enjoy!

BETTER:



Generate yourself a key-pair: `ssh-keygen`
[with passphrase]

Upload the public one:
`ssh-copy-id`

Have a session under
`ssh-agent`

Add keys to the session
`ssh-add`

Enjoy!

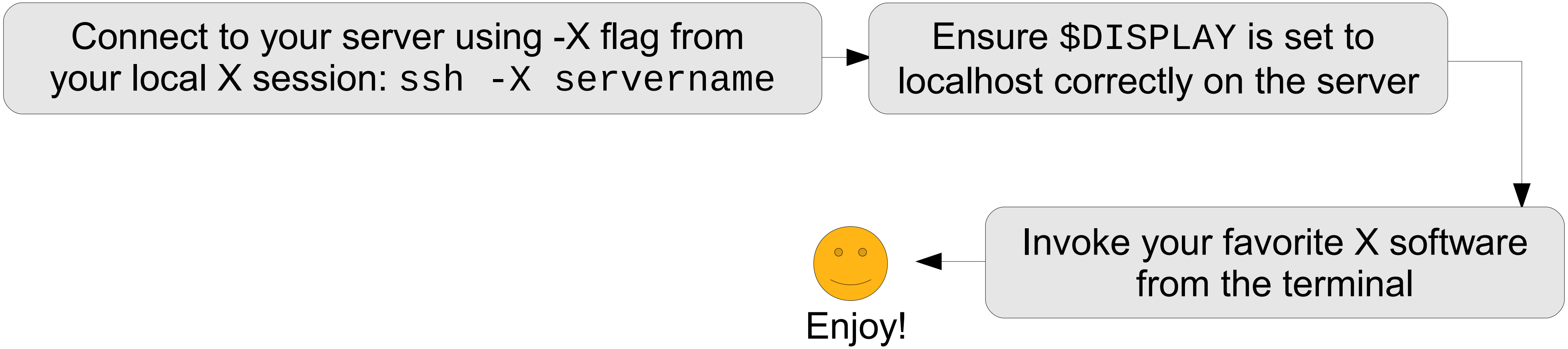
Beware: `Cycle your keys frequently (like your passwords). Protect private-key files, including your backups. Use -t, -X/x with ssh-add`

ssh and GUI

forwarding X windows thru ssh

Purpose: Use programs with GUI remotely

Prerequisites: A remote server with X client libraries (and some useful software :) installed and ssh daemon configured with X11Forwarding. Also, a locally installed X server.



Connect to your server using -X flag from your local X session: `ssh -X servername`

Ensure `$DISPLAY` is set to localhost correctly on the server

Invoke your favorite X software from the terminal

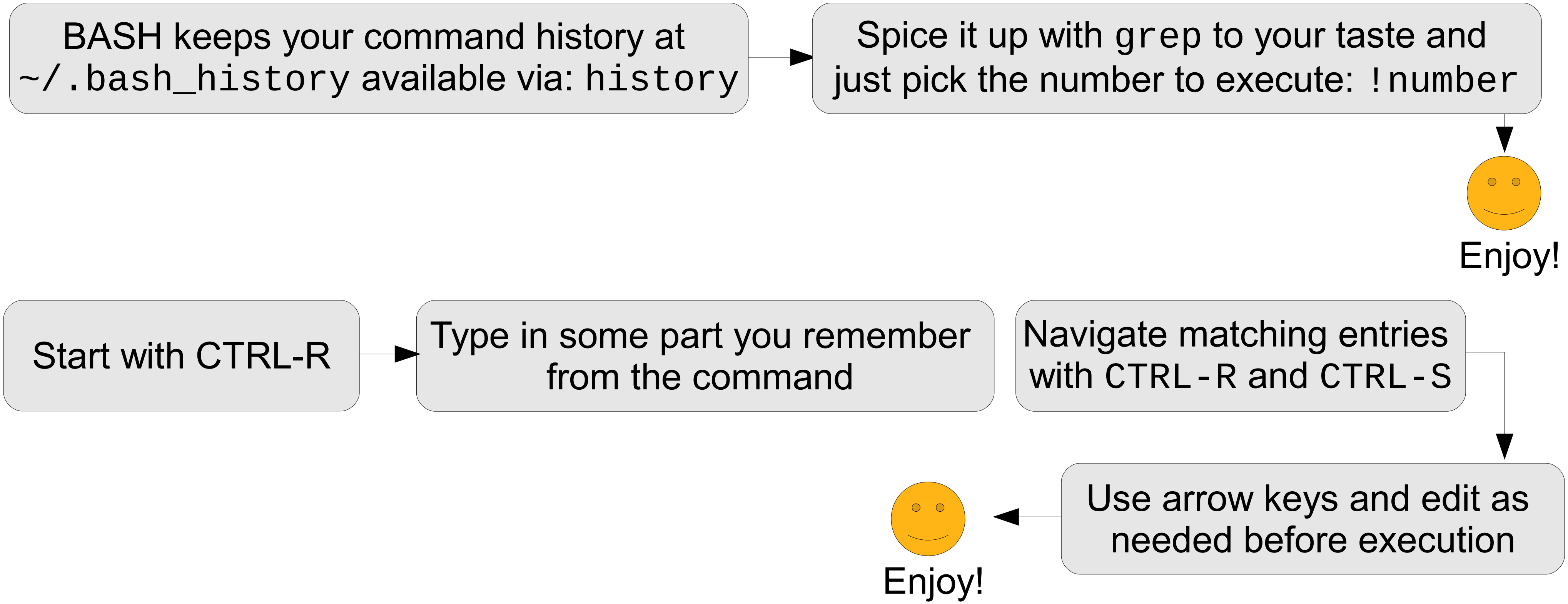
Enjoy!

Beware: High volumes of network traffic are involved! Be considerate of others, use sparingly. Will drive you nuts over a low bandwidth link.

What did I type again?

laziness with style

Purpose: Avoid typing the same things over and over again and fight carpal tunnel syndrome back



BASH keeps your command history at `~/.bash_history` available via: `history`

Spice it up with `grep` to your taste and just pick the number to execute: `!number`

Enjoy!

Start with CTRL-R

Type in some part you remember from the command

Navigate matching entries with `CTRL-R` and `CTRL-S`

Use arrow keys and edit as needed before execution

Enjoy!

Beware: <Arrows> and <TAB> are your friends. $HISTSIZE controls the history window width into past. CTRL-S might be assigned to t-stop in your terminal: stty -a / -ixon

Behind the scenes

Purpose: Have processes run when you are not there

`&` works most of the time but do not forget STDIN (`0`), STDOUT (`1`), and STDERR (`2`)

```
command </dev/null >&0 2>&0 &
```

Alternate 1:

`nohup` runs a command detached from its terminal, outputting to a file

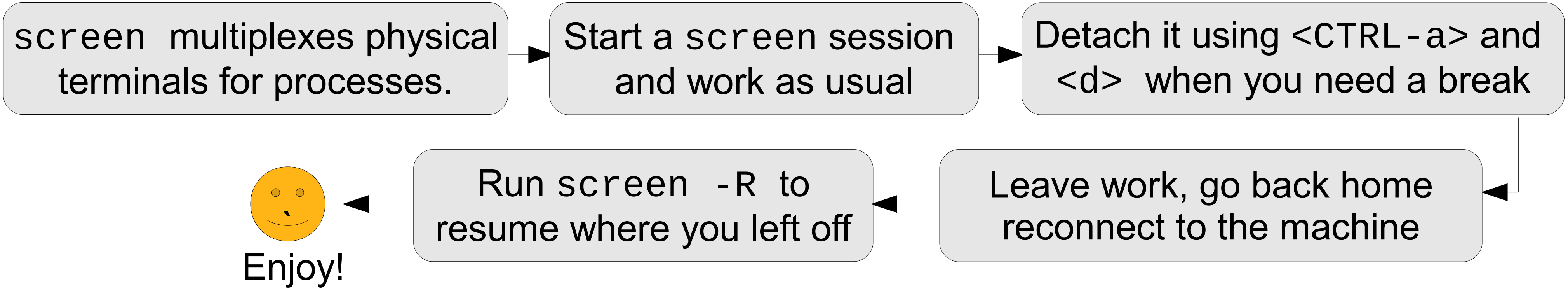
```
nohup command output_file &
```

Alternate 2:

`at` runs a script (or command) at a specified time, e-mailing the STDOUT and STDERR

```
at -f script.sh NOW
```

Interactive Sessions:



`screen` multiplexes physical terminals for processes.

Start a `screen` session and work as usual

Detach it using <`CTRL-a`> and <d> when you need a break

Leave work, go back home reconnect to the machine

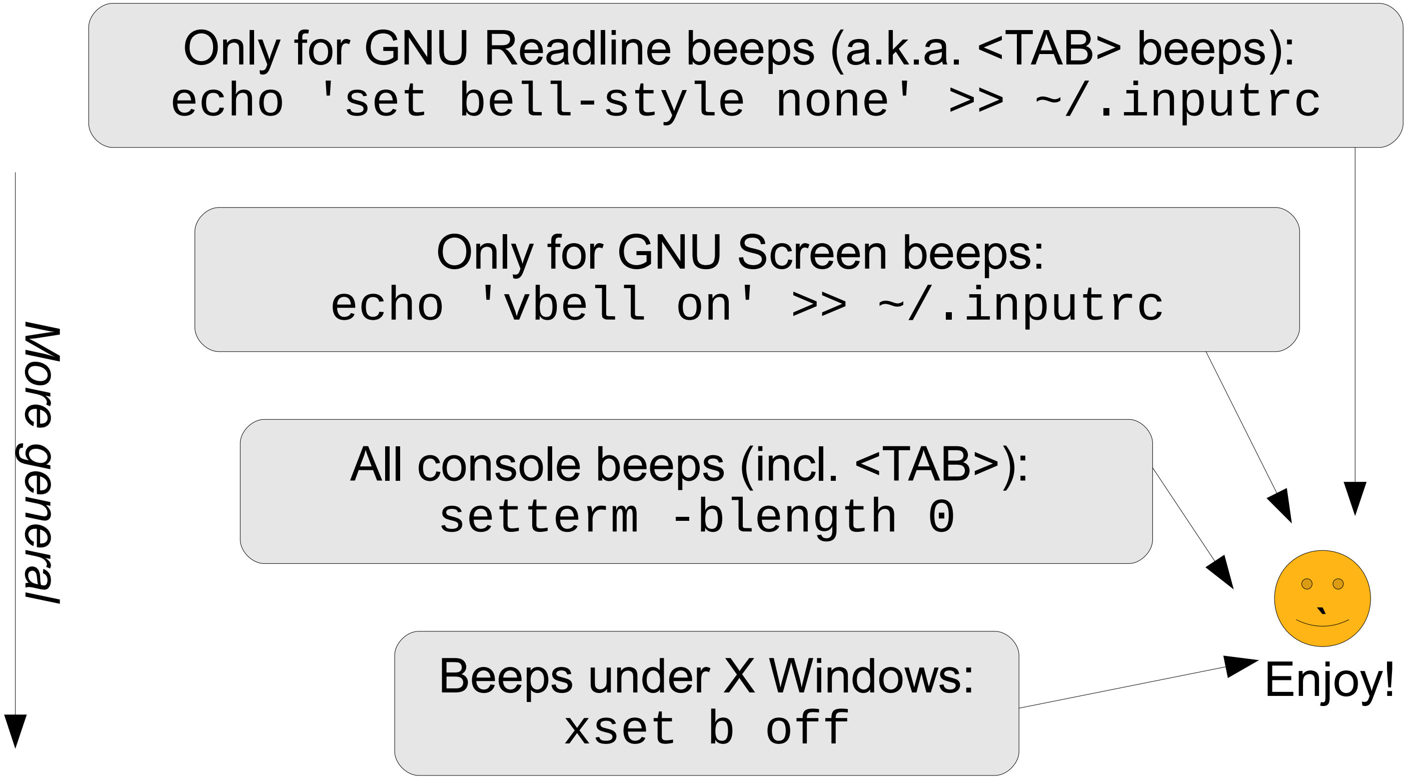
Run `screen -R` to resume where you left off

Enjoy!

Beeping console woes

look ma! no beeps!

Purpose: Silence disturbing beeps for good.



Only for GNU Readline beeps (a.k.a. <TAB> beeps):

```
echo 'set bell-style none' >> ~/.inputrc
```

Only for GNU Screen beeps:

```
echo 'vbell on' >> ~/.inputrc
```

All console beeps (incl. <TAB>):

```
setterm -blength 0
```

Beeps under X Windows:

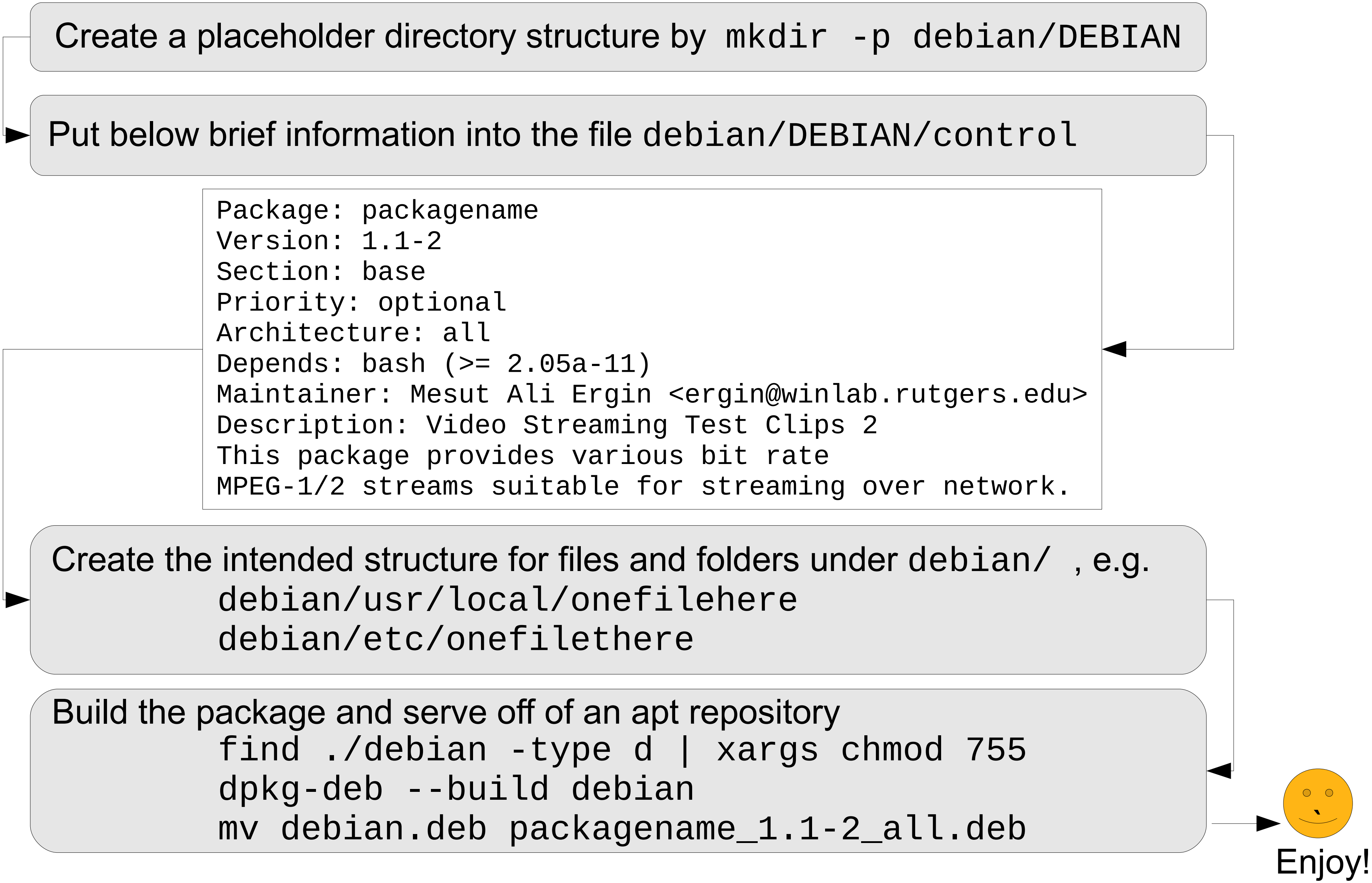
```
xset b off
```

Beware: It is good to disable only annoying ones, and keep the important ones. These are for BASH and have duals for TCSH and ZSH.

Making of a simple .deb package

for dummies

Purpose: A simple installable Debian package for your deliverable



Create a placeholder directory structure by `mkdir -p debian/DEBIAN`

Put below brief information into the file `debian/DEBIAN/control`

```
Package: packagename
Version: 1.1-2
Section: base
Priority: optional
Architecture: all
Depends: bash (>= 2.05a-11)
Maintainer: Mesut Ali Ergin <ergin@winlab.rutgers.edu>
Description: Video Streaming Test Clips 2
This package provides various bit rate
MPEG-1/2 streams suitable for streaming over network.
```

Create the intended structure for files and folders under `debian/` , e.g.

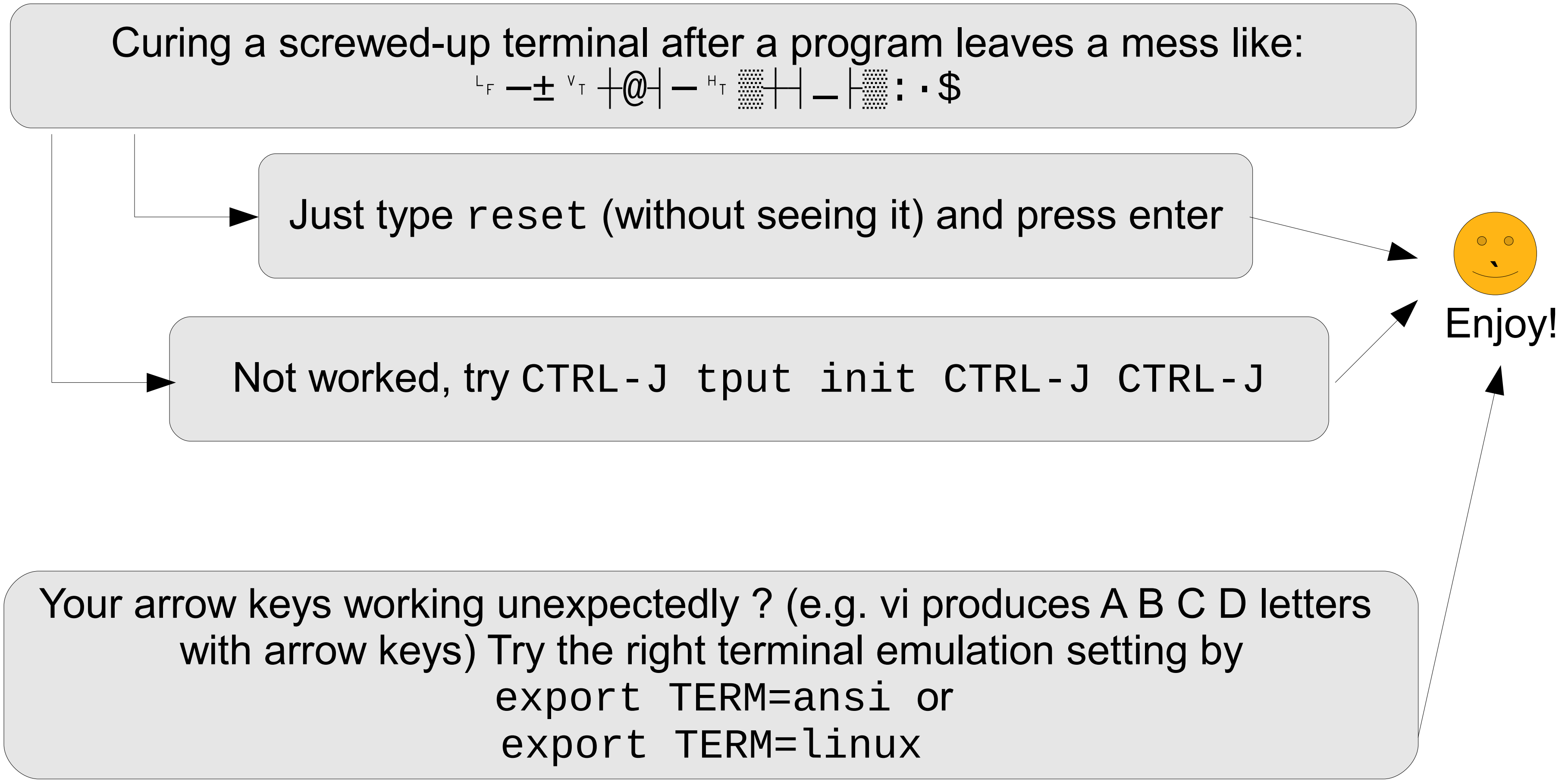
```
debian/usr/local/onefilehere
debian/etc/onefilethere
```

Build the package and serve off of an apt repository

```
find ./debian -type d | xargs chmod 755
dpkg-deb --build debian
mv debian.deb packagename_1.1-2_all.deb
```

Abnormal terminals

Purpose: `Fixing wonky terminal behavior`



Curing a screwed-up terminal after a program leaves a mess like:
␊─±␋┼@┤─␉▒┼┤_├▒:·$

Just type `reset` (without seeing it) and press enter

Not worked, try `CTRL-J tput init CTRL-J CTRL-J`

Your arrow keys working unexpectedly ? (e.g. vi produces A B C D letters with arrow keys) Try the right terminal emulation setting by
`export TERM=ansi` or
`export TERM=linux`

Where is that file?

Far, far away if you really need it

Purpose: Finding the file(s) you know something about

`find / -iname *somepart*` will find files and folders that have `somepart` in their names

`find ~ -iname *.mp3 -size +5M` all mp3 files under my home greater than 5MBytes

`find / -mtime 0 -name *.txt -exec cp '{}' /tmp \;`
find all txt files modified in the last 24hrs and copy them to /tmp

Not-so-universal but fast ones: (*See notes below)*

`locate -i somepart` will find files and folders that have `somepart` in their names

`apt-file search -i somepart` will find files and folders
that came from installed packages with `somepart` in their names

`which fooexec` will find the command `fooexec` anywhere in $PATH

Beware: locate and apt-file will be fast, however they use periodically updated databases, which might have gone stale. So, don't give up if locate or apt-file fails. Use find or run <updatedb> or <apt-file update> before using locate or apt-file.

Who is using that port/file?

Purpose: `Figure out resource usage for ports and files/devices`

a.k.a. couldn't open socket. port might already be in use

`netstat -lnp` will give you the list of open sockets (Unix and Inet) with assoc. processes

a.k.a. umount: /mnt/usbdisk: device is busy

`lsof +D /mnt/usbdisk` will give you the list of processes accessing `/mnt/usbdisk`

Try terminating processes gracefully (exiting programs) to reclaim these used resources.

Use `kill -9 <pid>` to force termination of the processes

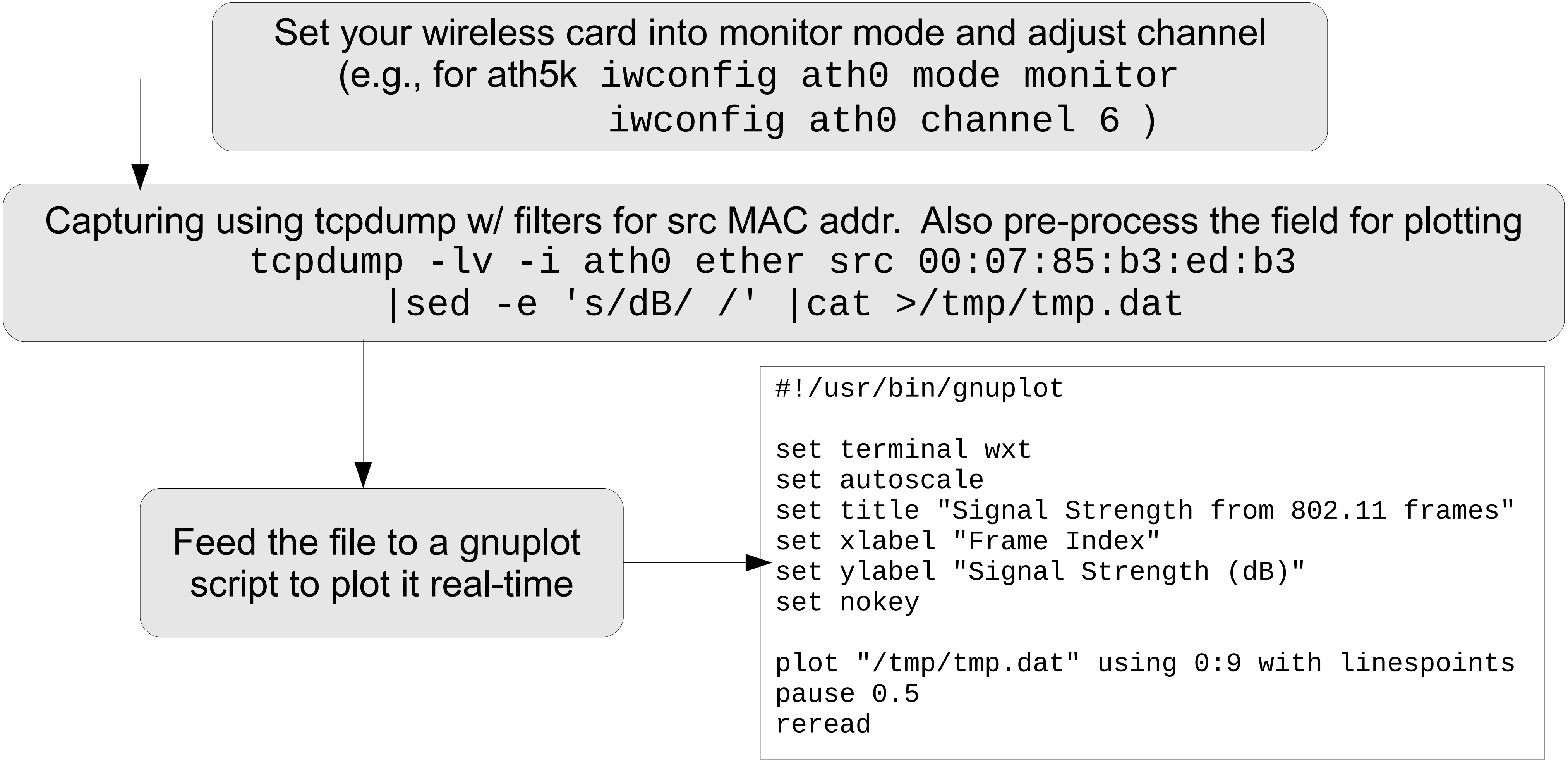
Use `sync` and then `umount -l /mnt/usbdisk` to resort to lazy unmount

Capture 802.11 Signal Strength

quick and dirty way for easy tests

Purpose: Capture 802.11 frames and plot signal strength of them

Prerequisites: Supported WiFi card (e.g. intel, atheros, broadcom, prism1/2/GT, etc.) tcmpdump and gnuplot packages installed.



Set your wireless card into monitor mode and adjust channel (e.g., for ath5k

```
iwconfig ath0 mode monitor
iwconfig ath0 channel 6 )
```

Capturing using tcpdump w/ filters for src MAC addr. Also pre-process the field for plotting

```
tcpdump -lv -i ath0 ether src 00:07:85:b3:ed:b3
|sed -e 's/dB/ /' |cat >/tmp/tmp.dat
```

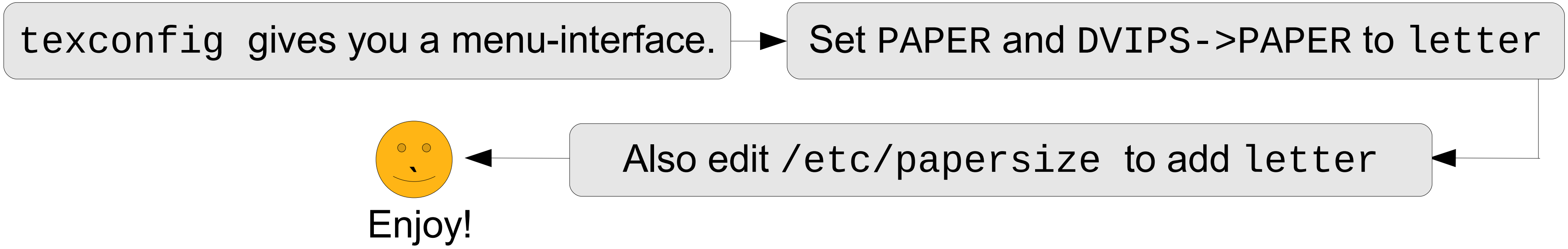
Feed the file to a gnuplot script to plot it real-time

```
#!/usr/bin/gnuplot

set terminal wxt
set autoscale
set title "Signal Strength from 802.11 frames"
set xlabel "Frame Index"
set ylabel "Signal Strength (dB)"
set nokey

plot "/tmp/tmp.dat" using 0:9 with linespoints
pause 0.5
reread
```

LaTeX, please!

Next bugchecks by Don Knuth will be in 2013, 2020, 2028, and 2037

Purpose: `Control default page sizes and font types for latex`



`texconfig` gives you a menu-interface.

Set PAPER and DVIPS->PAPER to `letter`

Also edit `/etc/papersize` to add `letter`

Enjoy!

Everyone wants press-ready PDFs with embedded Type-I fonts:

Make sure your linux box has `tetex-extra` and `cm-super` packages

If possible, use `\usepackage{ae,aecompl}` for T1 fontenc package (*style file permitting*)

Use `dvips` with `-P www` option (or `-P pdf`)

Use `ps2pdf` with `-dPDFSETTINGS=/prepress -dEmbedAllFonts=true -dSubsetFonts=true -dEPSCrop=true` options

Check the final PDF file for the used font list with `pdffonts`

Beware: `Getting fonts right might become tricky. If you don't convert fonts to shapes in your EPS figures, non-Type I font issues may persist`

Editing .eps files

a.k.a. changes requested one hour before the submission

Purpose: Making reasonable changes on an EPS image file without rasterizing it

Your text editor may help for very small text changes (if text is not-converted).

Try `inkscape` open->eps option first (experimental feature)

Use ps2pdf on the file and use `inkscape` open->pdf option

Use pdf2svg from PDFtron Inc. on the file and use `inkscape` open->svg option

Import eps into `scribus` and export an svg to be used with `inkscape` open->svg option

Beware: Vector images are unlike raster ones. Editing them will require some getting used to. It is always best to do edits on the source file, before eps conversion.

Doing Backups

is like shaving/ironing. You know you have to do it but...

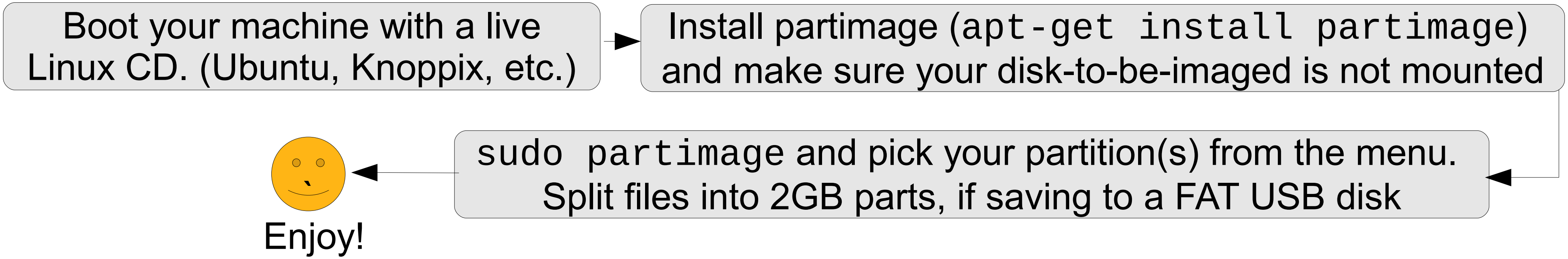
Purpose: Recover from the disaster day with minimal damage
Prerequisite: USB memory/hard drive for storage

Backup a folder to another disk/folder with rsync over ssh. *a.k.a. How do I backup my website ?*

```
rsync --progress -av ergin@server.com:/home/ergin/public_html /tmp/backup
```

Next time, only the changed files will be transferred, keeping two copies in sync.

Backup your hard drive (a complete image) by



Boot your machine with a live Linux CD. (Ubuntu, Knoppix, etc.)

Install partimage (`apt-get install partimage`) and make sure your disk-to-be-imaged is not mounted

`sudo partimage` and pick your partition(s) from the menu. Split files into 2GB parts, if saving to a FAT USB disk

Enjoy!

Beware: There are many ways to do backups and there might be more efficient ones depending on your needs. Backup onto a media which will not be in use and will be stored for only restoration purposes. partimage won't work on disks with bad/unreadable blocks. Use ddrescue instead.

The End