

# Setting up a decent dev box for OSS work

mark.gross@intel.com  
oscon 2008



# I'm a huge OSS community wannabe

- Python
- django
- Gnome
- Freedesktop
- Open Embedded
- Linux kernel / linux power management



# Why are you here?

- Using bleeding edge OSS stuff?
- To be part of an OSS community?

Slides and additional content on :  
[www.thegnar.org/devbox](http://www.thegnar.org/devbox)



# Overview / outline

- getting most of the tools needed to work with upstream bits
- Setting up email for busy OSS mailing lists
- Random things as long as we have time...
  - quilt
  - git
  - Linux kernel development
  - IRC
  - xrandr
  - minicom
  - Python

# Some resources

- [www.mutt.org](http://www.mutt.org)
- [www.procmail.org](http://www.procmail.org)
- [msmtp.sourceforge.net](http://msmtp.sourceforge.net)
- [fetchmail.berlios.de](http://fetchmail.berlios.de)
- [git.or.cz](http://git.or.cz)
- [savannah.nongnu.org/projects/quilt](http://savannah.nongnu.org/projects/quilt)
- [www.thegnar.org/devbox](http://www.thegnar.org/devbox)

# Base Install

- Option 1. Basic 8.04.1 desktop install
- Uses up 2200MB.
- Option 2. Alternate 8.04.1 command line only install
- storage challenged systems
- without desktop clutter
- From the CD boot loader select F3 and choose command line only install
- Uses up 645MB.



# Adding packages needed for OSS work

- start off with the Distro versions
- Apt-get build-dep is **magic**
- Apt-get install is cool too.
- <http://www.thegnar.org/devbox>

# Apt-get examples from my website

- #kernel development

```
sudo apt-get build-dep linux-generic linux-image-generic  
linux-doc
```

```
sudo apt-get install linux-kernel-devel kernel-package  
libncurses-dev linux-source docbook-utils xmlto
```

## #xorg development

```
sudo apt-get build-dep xorg x11-apps xserver-xorg
```

```
sudo apt-get install xorg-dev libssl-dev
```

# When build-dep misses

- Google
- Use Synaptic Package manager to search for the missing package.

# Additional apt-get friends

- Use `dpkg -l`
- `dpkg` and `apt-get` man pages should be read.
- `apt-get clean`
- `apt-get purge <name>` will remove name (d) package

# Email

- Based on [www.andrews-corner.org/mutt.html](http://www.andrews-corner.org/mutt.html)



# Strange names

- Mail Transfer Agent (MTA)
  - Mail Retrieval Agent (MRA) -- fetchmail
  - Mail Sending Agent (MSA) -- msmtmp/ssmpt
- Mail Delivery Agent (MDA) -- procmail
- Mail User Agent (MUA) -- mutt

# Tools I use

- Mutt : [www.mutt.org](http://www.mutt.org)
- Procmail : [www.procmail.org](http://www.procmail.org)
- Msmtplib : [msmtplib.sourceforge.net](http://msmtplib.sourceforge.net)
- Fetchmail : [fetchmail.berlios.de](http://fetchmail.berlios.de)
- “working” config file examples on my web site  
[www.thegnar.org/devbox/email.html](http://www.thegnar.org/devbox/email.html)

# fetchmail configuration highlights

poll pop.gmail.com

protocol POP3

user 'dec0cb5a'

password 'oscon(123)'

mda "/usr/bin/procmail -d %T"

options no keep ssl sslcertck sslcertpath /etc/ssl/certs

- `chmod 600 ~/.fetchmailrc`

# msmtp config highlights

host smtp.gmail.com

port 587 # gmail specific

tls on **#Transport Layer Security**

tls\_starttls on

tls\_trust\_file /usr/share/ca-  
certificates/mozilla/Thawte\_Premium\_Server\_CA.crt

- `chmod 600 ~/.msmtprc`
- <http://msmtp.sourceforge.net/doc/msmtp.html>

# gmail account settings

- Forwarding and POP/IMAP
- POP Download
- 1 : enable POP download
- 2 : delete Gmail's copy when accessed
- Save Changes.

# procmail config highlights

```
# Recipes see "man procmailex"
```

```
:0:
```

```
* ^TOdec0cb5a
```

```
IN.dec0cb5a
```

```
#add more rules here:
```

```
# Last rule: Put mail into mbox #
```

```
:0:
```

```
mbox
```

# procmail examples

:0:

```
* ^X-Mailing-List:. *linux-rt-users.vger.kernel.org
```

```
IN.linux-rt-users
```

:0:

```
* ^X-Mailing-List:. *linux-kernel.vger.kernel.org
```

```
IN.lkml
```

# Procmail help

- Man procmailrc
- Man procmailex
- Google for example .procmailrc files

# mutt

- mutt is old school
  - [www.mutt.org](http://www.mutt.org)
- “alias” file set up in the .muttrc file.
  - alias dorkbotpdx  
dorkbotpdx-blabber@dorkbot.org
  - alias AHRS ahrs user1@isp1.com,  
user2@isp2.org, ...

# mutt configuration highlights

```
set realname = "oscon 2008"
```

```
set from = "dec0cb5a@gmail.com"
```

```
set use_from = "yes"
```

```
set envelope_from = "yes"
```

```
set sendmail=/usr/bin/msmtp
```

```
set alias_file=~/.mutt/aliases
```

```
source ~/.mutt/aliases
```

```
macro index,pager I '<shell-escape> fetchmail  
-v<enter>'
```

# finishing touches

- `chmod 600 .fetchmailrc`
- `chmod 600 .msmtprc`
- `mkdir ~/.mutt`
- `touch ~/.mutt/aliases`

# SCM related



# Using quilt

- Quilt is awesome!
- Many folks use quilt within a git tree.
  - use git to track upstream kernels
  - use quilt to keep track of their work
- Use:
  - `quilt new my_cool_feature.diff`
  - `quilt add <existing file>`
  - `quilt refresh`
- `Quilt edit <filename>` is a good way to add file before editing.

# Using quilt

- quilt -h
- quilt pop
- quilt push
- quilt series
- quilt refresh
- Gotchas
  - parent directories with a directory named “patches”
  - forgetting to add files

# GIT

- Git is a distributed version control system.
- hard to wrap my head around
- none of the tutorials cover : **git-remote**
- See my git page  
<http://www.thegnar.org/devbox/GitTipsThatHe>  
for more detail

# common git commands

- git-remote :

- `git-remote add linus git://git.kernel.org/.../torvalds/linux-2.6.git`
- `git-remote add fboot git://git.kernel.org/.../arjan/linux-2.6-fastboot.git`
- `git-remote add internal mgross@linux-repo.jf.intel.com:/internal/kernel`

- git-status

- git-reset -hard

- git-clean -d

- git-merge

# more common git commands

- git-rebase
- git-branch -a
  - git-branch fastboot fboot/master
- git-fetch
  - **git-fetch fboot**
- git-checkout
  - git-checkout -b fastboot fboot/master

# git with quilt

- checkout your code base with git
- create your new working patch with quilt
  - add files and work work work
- quilt refresh then quilt pop all your patches
- git-fetch - git-merge
- quilt push your work and fix up any issues you may have.
- When ready or just post the patch to LKML.

# Kernel related

- Read `./Documentation/{SubmitChecklist,  
SubmittingDrivers, SumbittingPatches}`



# tools

- Make help
- Ctags : make tags
  - `gvim` works well with the TAGS file
- CSCOPE : old school code browser
  - `make cscope`
  - `cscope -kd`
  - To exit cscope you hit `^d`
- Make {htmldocs, mandocs,...} output are interesting.

# ./scripts/checkpatch

- It's a format and style checking tool that codifies the Linux kernel coding standards
- “./scripts/checkpatch <pathname of patch>” from within the kernel root tree.

# Web and communications



# IRC

- clients:
  - xchat seems to be the easiest for me to use
  - irssi
- Don't post large traces, like a failed compile, to IRC.
  - <http://rafb.net>
- naming a person in your text will “beep” on their remote.

# More IRC

- type-oh correction is normally expressed:
- `s/<your-type-oh>/<what you wanted it to say>/`
- it won't redact your screw up
- `/` commands are how IRC works.
  - `/msg <nick> <message text>`
  - `/kick`
  - `/me`

# Simple web site updates

- ncftp is the nicest FTP client I know
- ncftp -u <loginname> IP/site URL
- Seamonkey (mozilla) Composer.
- its dumb, but works
- It has some warts (save work often)

# Utilities



# xrandr

- kitchen sink full of options
- Options that I've used:
  - `xrandr -auto`
  - `xrandr -output LVDS -off`
  - `xrandr -output VGA -mode 800x600`
  - `xrandr -output VGA -mode 1024x768 -output LVDS -mode 1024x768`
  - `xrandr -output VGA -mode 1280x1024`
- `xrandr` with no options shows available modes per output device

# minicom

- (the?) serial terminal program.
- very old school
- ^a^z gets a help menu of sorts
- some distros give it permission issues with the /dev/ devices
- minicom -s
- minicom

# Python stuff



# Python must haves (for me)

- code checkers : pylint, pyflakes
- python shell : ipython
- python docs
- graphing tools : matplotlib
- scientific stuff : scipy, numpy
- serial port tools : python-serial.
- image processing tools : python-imaging

# iPython is cool

- ? will provide basic help
- help() will give access to all of the python help on your system.
- See Jeff Rush's showmedo videos on iPython
- <http://showmedo.com/videos/series?name=C>

# ipython dynamic features

- running someone else's code with ipython can help debug issues found within that code.
- set magic “%pdb on” to enable the python debugger to trigger on exception.
- %prun / %run -p profiling
- %time / %timeit
- %xmode
- %colors, %cpaste

# THANK YOU



Software and Solutions Group

