Your First Linux Kernel Patch

Kernel Programming Is Like SEX

- You’re **certain** other people are doing it more than you are.

Kernel Programming Is Like SEX

- When you ask people with more experience than you, they **avoid** the conversation or they quickly try to end the conversation (“go away kid, I’m busy”).

Kernel Programming Is Like SEX

- You’re left to “**just figure it out on your own**” and you’re not sure what you’ve figured out is right.

Choose a host

- We have 12 virtual machines
- Username: sfd2013
- Password: softwarefreedomday

- suse{1,2,3}.mailicorn.com
- rhel{1,2,3).mailicorn.com
- ubuntu{1,2,3}.mailicorn.com
- amazon{1,2,3}.mailicorn.com

Installing git

- See if git is already installed

```
user@linux:~$ which git
```

Installing git

- As root, install git if needed.

```
# Fedora/CentOS/RHEL
user@linux:~# yum install git

# Debian/Ubuntu
user@linux:~# apt-get install git
```

Configure yourself in git

- Create your identity (goes in ~/.gitconfig)

```
user@linux:~# git config --global \
  user.email “myemail@gmail.com”

user@linux:~# git config --global \
  user.name “My Name”
```

Download net-next with git

- Takes about 12 minutes at RIT
- Installs into new subdirectory net-next

```
user@linux:~$ git clone \
git://git.kernel.org/pub/scm/linux/kerne
l/git/davem/net-next.git
```

- Try ‘git pull’ to ensure up to date

```
user@linux:~$ git pull
```

Create a branch of your code

- Declare a code branch your changes will be recorded under

```
user@linux:~$ git branch devel
```

- Then actually change to it

```
user@linux:~$ git checkout devel
```

- You’re ready to make changes!

Edit code and commit it

- Do a ‘cd net-next/drivers/net/ethernet’
- Edit code
- When happy, do a ‘git commit -a’. Note: the first line becomes your Subject line in the email, so choose it well.

One of the possible patches

- How a device registers for interrupts can be confusing if there are more than two of them.

```
root@host:~# egrep "NE2000" /proc/interrupts
  5:          1          0   IO-APIC-edge      NE2000
  7:          0          0   IO-APIC-edge      NE2000
 root@host:~#
```

- Which one is which?

The convention

- The accepted style now is to register the device name (ie eth0) when requesting an interrupt with request_irq().
- It would look like this:

```
root@host:~# egrep "NE2000" /proc/interrupts
  5:          1          0   IO-APIC-edge      eth0
  7:          0          0   IO-APIC-edge      eth1
 root@host:~#
```

Go forth and code

- You gotta do some work, I’m not gonna do this for you!
- Look in the following files under drivers/net/ethernet:
- 3c515.c
- ne.c
- ni65.c
- lance.c
- hp100.c
- de4x5.c

For the netdev mailing list, check your progress

- The netdev kernel mailing list is one of a few that use the ‘patchwork’ software to track patches semi-automatically. It is run by Dave Miller

  http://patchwork.ozlabs.org/project/netdev/list/

Generate the patch

- Do a ‘cd ~/net-next’
- Generate patch with

  ```
  git format-patch --subject-prefix \
  net-next  --signoff master
  ```
- A file is created starting with ‘0001-’

Installing git-email

- Try ‘git send-email’ and if it fails, install the package as root.

```
# Fedora/CentOS/RHEL
user@linux:~# yum install git-email

# Debian/Ubuntu
user@linux:~# apt-get install git-email
```

Configure git-email in ~/.gitconfig

- Edit ~/.gitconfig with your email information

```
[sendemail]
smtpencryption = tls
smtpserver = smtp.gmail.com
smtpuser = tedheadster@gmail.com
smtpserverport = 587
smtppass = PASSWORD
```

Send a test email of the patch

- Email the patch to an alternate personal email as a test, because it automatically CC's your address in ~/.gitconfig.

```
user@linux:~$ git send-email -to
whiteheadm@acm.org 0001-patchfile
```

Submit the patch!

- Consult the source tree MAINTAINERS file to find where you should submit the patch.

```
user@linux:~$ git send-email -to
netdev@vger.kernel.org 0001-patchfile
```

Questions on what we did?