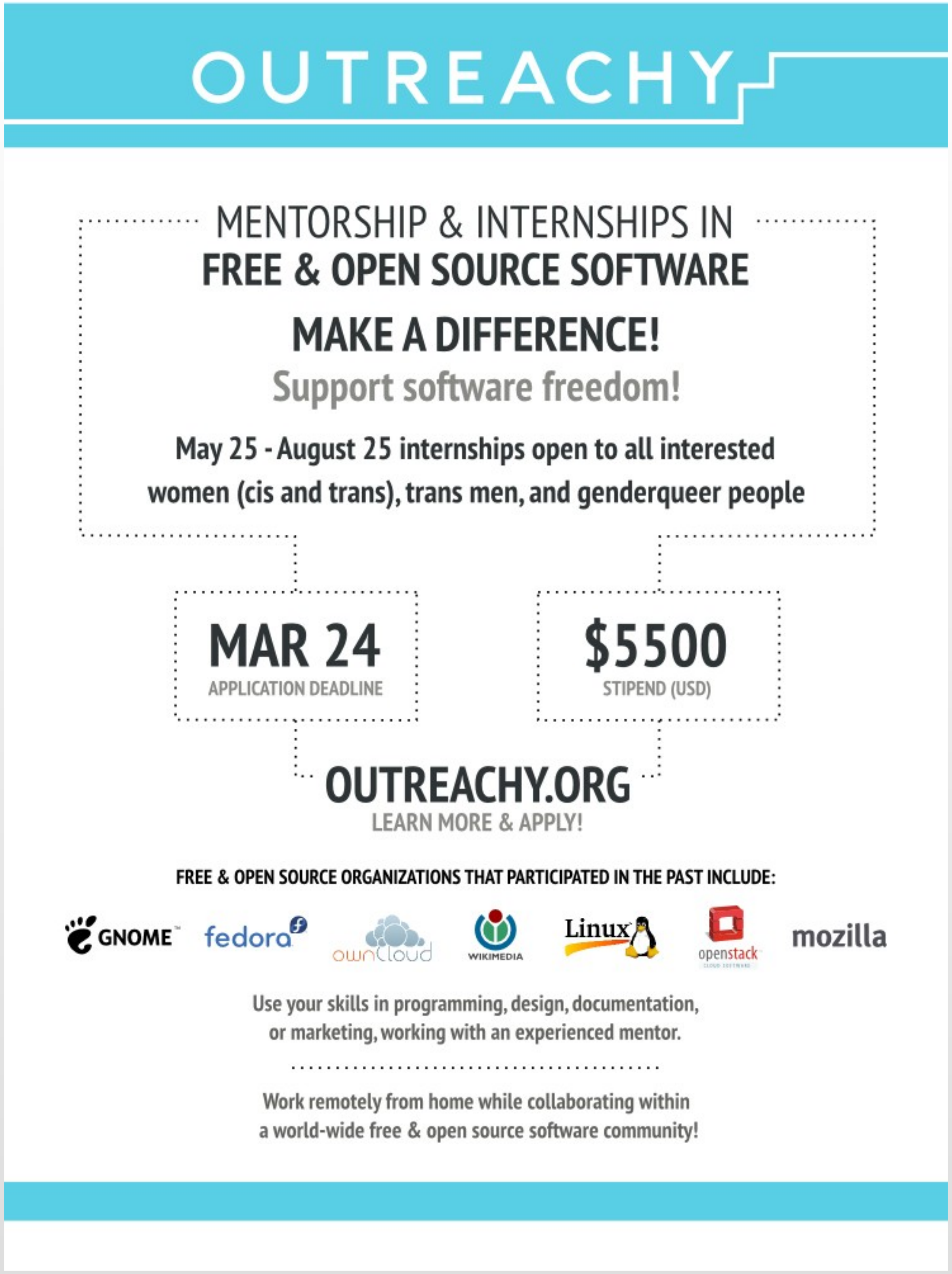
Linux Kernel Hacking 101

Kelley Nielsen
salticidoftheearth.com

github.com/shegeek/kernel-hacking-101



OUTREACHY

MENTORSHIP & INTERNSHIPS IN
FREE & OPEN SOURCE SOFTWARE

MAKE A DIFFERENCE!

Support software freedom!

May 25 - August 25 internships open to all interested women (cis and trans), trans men, and genderqueer people

MAR 24
APPLICATION DEADLINE

$5500
STIPEND (USD)

OUTREACHY.ORG
LEARN MORE & APPLY!

FREE & OPEN SOURCE ORGANIZATIONS THAT PARTICIPATED IN THE PAST INCLUDE:

GNOME fedora ownCloud WIKIMEDIA Linux openstack mozilla

Use your skills in programming, design, documentation, or marketing, working with an experienced mentor.

Work remotely from home while collaborating within a world-wide free & open source software community!

The process of kernel hacking is a

CYCLE

The Creative Cycle

- Code your changes
- Send in your patch
- Gather feedback
- Repeat

The Creative Cycle

- Find a contribution to make
  - Read stored communications
  - Gain experience
  -

The Creative Cycle

- Find a contribution to make
  - Read stored communications
  -
  -

The Creative Cycle

- Find a contribution to make
  - Read stored communications
  - Gain experience
  -

The Creative Cycle

- Find a contribution to make
  - Read stored communications
  - Gain experience
  - Ask

The Creative Cycle

- Code your contribution
- Prepare and send the patchset

The Creative Cycle

- Gather feedback
  -
  -
  -
  -

The Creative Cycle

- Gather feedback
  - Testing results
  -
  -
  -

The Creative Cycle

- Gather feedback
  - Testing results
  - Mentoring and guidance
  -
  -

The Creative Cycle

- Gather feedback
  - Testing results
  - Mentoring and guidance
  - Discussion of strategies
  -

The Creative Cycle

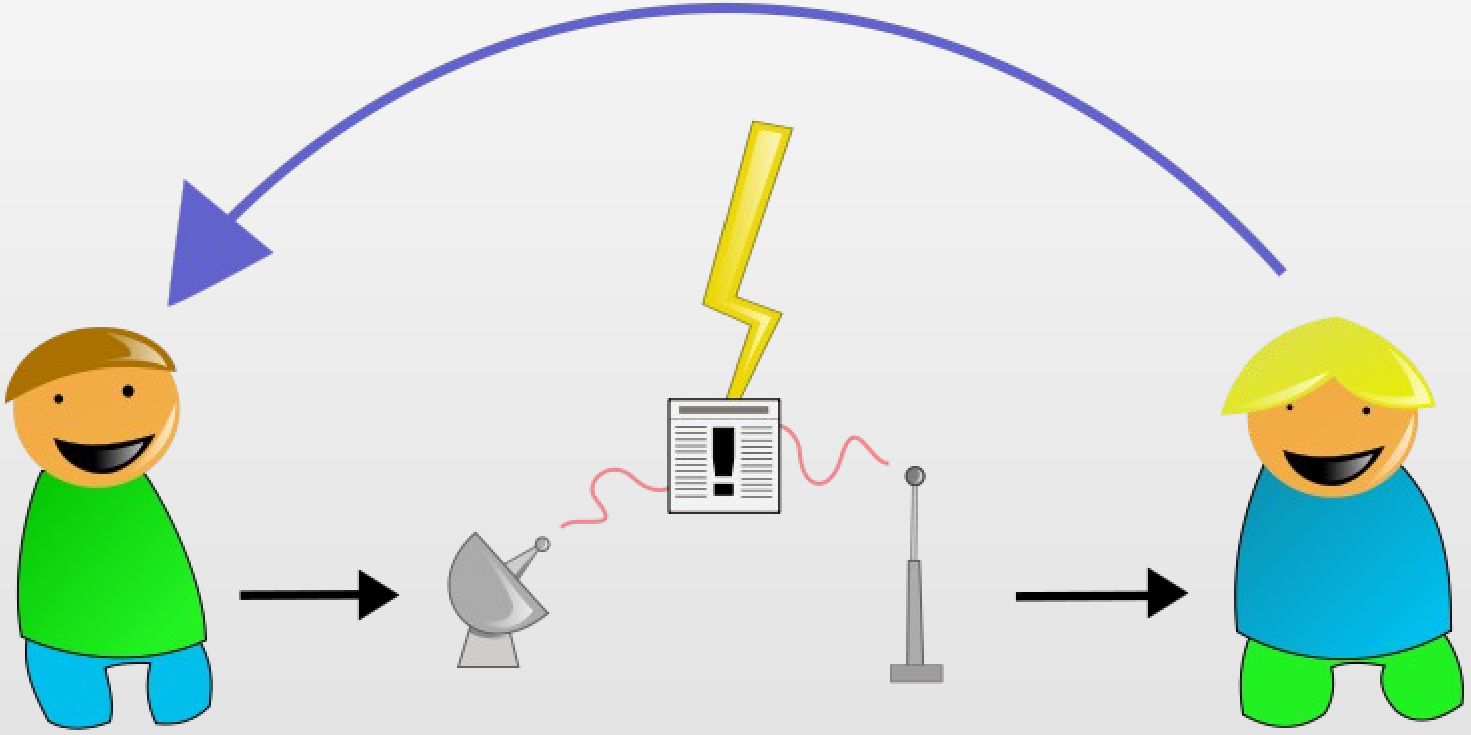
- Gather feedback
  - Testing results
  - Mentoring and guidance
  - Discussion of strategies
  - General suggestions

The Creative Cycle

- Incorporate changes
  - Use feedback to improve
  - If your contribution is rejected, find another
- Regenerate and resend your patchset

Repeat the previous two slides until
Your patchset is accepted…
Then repeat with a new patchset…

Communicating with team members and community



Communicating

- Mailing lists
-
-
-

Communicating

- Mailing lists
- Private email
-
-

Communicating

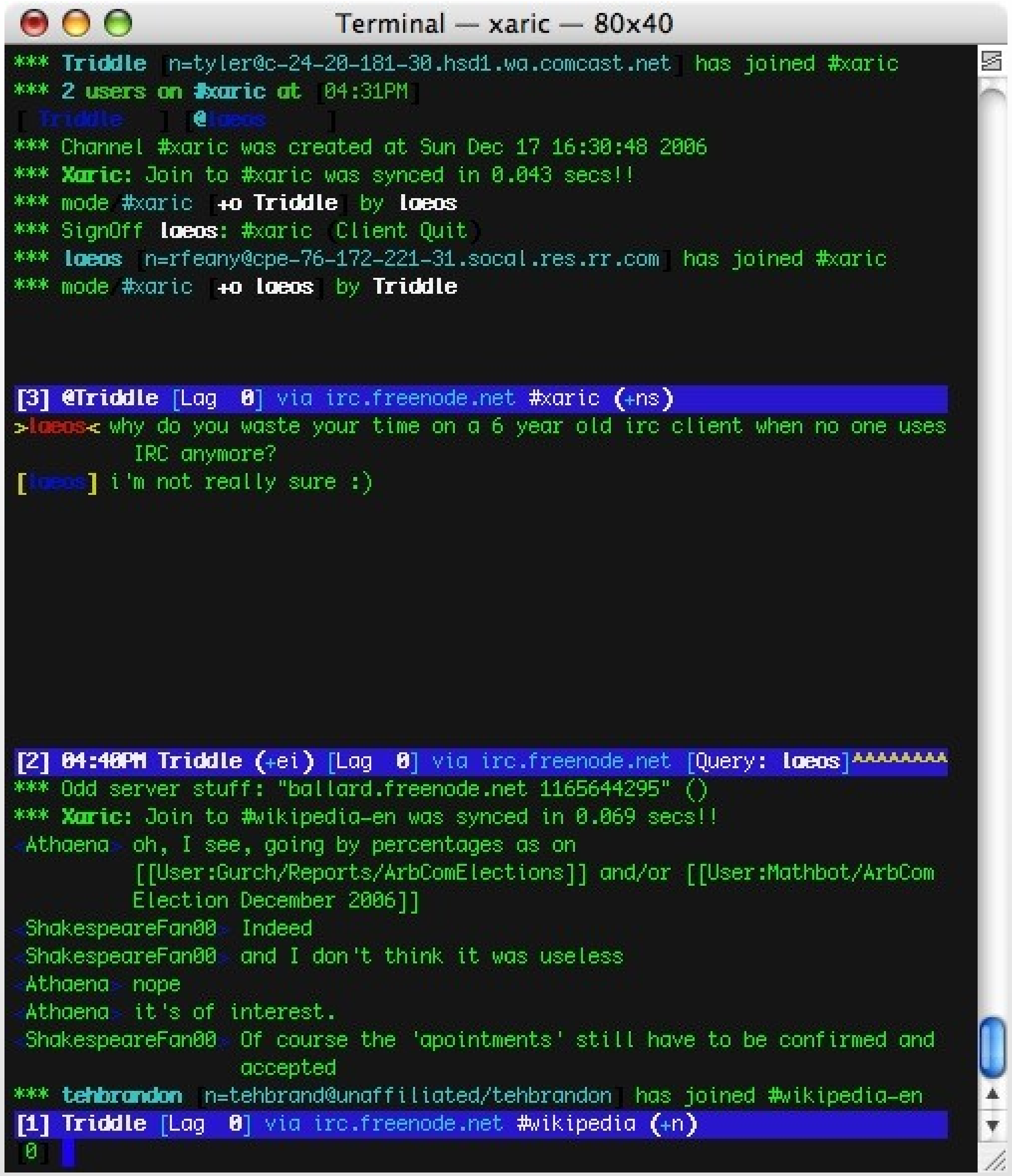
- Mailing lists
- Private email
- Conferences
-

Communicating

- Mailing lists
- Private email
- Conferences
- Linux Weekly News

Communicating

- Internet Relay Chat (IRC)
  - Looks like multi-way text messaging
  - Use a dedicated client (not a web client)
  - Connect to a network
  - Once on the network, join a channel



```
*** Triddle [n=tyler@c-24-20-181-30.hsd1.wa.comcast.net] has joined #xaric
*** 2 users on #xaric at [04:31PM]
[ Triddle    ] [@laeos      ]
*** Channel #xaric was created at Sun Dec 17 16:30:48 2006
*** Xaric: Join to #xaric was synced in 0.043 secs!!
*** mode/#xaric [+o Triddle] by laeos
*** SignOff laeos: #xaric (Client Quit)
*** laeos [n=rfeany@cpe-76-172-221-31.socal.res.rr.com] has joined #xaric
*** mode/#xaric [+o laeos] by Triddle




[3] @Triddle [Lag  0] via irc.freenode.net #xaric (+ns)
>laeos< why do you waste your time on a 6 year old irc client when no one uses
         IRC anymore?
[laeos] i'm not really sure :)









[2] 04:40PM Triddle (+ei) [Lag  0] via irc.freenode.net [Query: laeos]^^^^^^^^
*** Odd server stuff: "ballard.freenode.net 1165644295" ()
*** Xaric: Join to #wikipedia-en was synced in 0.069 secs!!
<Athaena> oh, I see, going by percentages as on
          [[User:Gurch/Reports/ArbComElections]] and/or [[User:Mathbot/ArbCom
          Election December 2006]]
<ShakespeareFan00> Indeed
<ShakespeareFan00> and I don't think it was useless
<Athaena> nope
<Athaena> it's of interest.
<ShakespeareFan00> Of course the 'apointments' still have to be confirmed and
                   accepted
*** tehbrandon [n=tehbrand@unaffiliated/tehbrandon] has joined #wikipedia-en
[1] Triddle [Lag  0] via irc.freenode.net #wikipedia (+n)
[0]
```

Coding your changes

Coding your changes

- Plan
- Code
- Compile & run
- Test & debug

Coding your changes

- Plan
  - Trace existing paths of execution
  -
  -
  -

Coding your changes

- Plan
  - Trace existing paths of execution
  - Find examples similar to your goal
  -
  -

Coding your changes

- Plan
  - Trace existing paths of execution
  - Find examples similar to your goal
  - Learn which lines are relevant
  -

Coding your changes

- Plan
  - Trace existing paths of execution
  - Find examples similar to your goal
  - Learn which lines are relevant
  - Your friends are grep and the lxr ident search

Coding your changes

- Hack your code
  - Language is C
  - Follow kernel coding style
  - Gain skill with your editor

Coding your changes

- Compile and run
  - This involves installation and rebooting!
  - ne
  -

Coding your changes

- Compile and run
  - This involves installation and rebooting!
  - Use a virtual machine or dedicated machine
  -

Coding your changes

- Compile and run
  - This involves installation and rebooting!
  - Use a virtual machine or dedicated machine
  - Maintain a known working and a test kernel

Coding your changes

- Test and debug
  - Your friend is printk()
  -
  -

Coding your changes

- Test and debug
  - Your friend is printk()
  - Find messages in /var/log/kern.log
  -

Coding your changes

- Test and debug
  - Your friend is printk()
  - Find messages in /var/log/kern.log
  - Create testing scenarios

Generating and Delivering Your Patch

Delivering Your Patch

- Git commit strategy
- Git branching strategy
- Composing the commit message
- The patch

Delivering Your Patch

- Git commit strategy
  - Make small incremental changes
  - Make changes in logical order
  - [illegible]
  -
  -

Delivering Your Patch

- Git commit strategy
  - Make small incremental changes
  - Make changes in logical order
  - [illegible]
  - Be prepared to alter previous commits

Delivering Your Patch

- Git commit strategy
  - Make small incremental changes
  - Make changes in logical order
  - Each change should be encapsulated
  - Be prepared to alter previous commits
  -

Delivering Your Patch

- Git commit strategy
  - Make small incremental changes
  - Make changes in logical order
  - Each change should be encapsulated
  - Be prepared to alter previous commits
  -

Delivering Your Patch

- Git commit strategy
  - Make small incremental changes
  - Make changes in logical order
  - Each change should be encapsulated
  - Be prepared to alter previous commits
  - Be prepared for upstream changes as you update your sources

Delivering Your Patch

- Git branching strategy
  - Keep master branch unmodified
  -
  -
  -

Delivering Your Patch

- Git branching strategy
  - Keep master branch unmodified
  - Create a working branch
  -
  -

Delivering Your Patch

- Git branching strategy
  - Keep master branch unmodified
  - Create a working branch
  - Use a throwaway branch to squash commits
  -

Delivering Your Patch

- Git branching strategy
  - Keep master branch unmodified
  - Create a working branch
  - Use a throwaway branch to squash commits
  - If this method doesn’t work for you, try another

Delivering Your Patch

- Composing the commit message
  - Work as hard on this as you do on coding!
  -
  -
  -

Delivering Your Patch

- Composing the commit message
  - Work as hard on this as you do on coding!
  - Read existing examples
  -
  -

Delivering Your Patch

- Composing the commit message
  - Work as hard on this as you do on coding!
  - Read existing examples
  - Be concise, yet thorough
  -

Delivering Your Patch

- Composing the commit message
  - Work as hard on this as you do on coding!
  - Read existing examples
  - Be concise, yet thorough
  - Use imperative language

Delivering Your Patch

- The patch itself
  - Run checkpatch.pl on the files you change

Delivering Your Patch

- The patch itself
  - Run checkpatch.pl on the files you change
  - Generate patch with a git command
  -
  -

Delivering Your Patch

- The patch itself
  - Run checkpatch.pl on the files you change
  - Generate patch with a git command
  - Cc everyone involved with the code
  -

Delivering Your Patch

- The patch itself
  - Run checkpatch.pl on the files you change
  - Generate patch with a git command
  - Cc everyone involved with the code
  - Send using plain text mail

Delivering Your Patch

- Why plain text?
  - Email formatting will break your code
  - Team members will apply your patches as is
  - Commit message will become subject line and content
  -

Delivering Your Patch

- Why plain text?
  - Email formatting will break your code
  - Team members will apply your patch as is
  -

    t
  -

Delivering Your Patch

- Why plain text?
  - Email formatting will break your code
  - Team members will apply your patch as is
  - Commit message will become subject line and content
  -

Delivering Your Patch

- Why plain text?
  - Email formatting will break your code
  - Team members will apply your patch as is
  - Commit message will become subject line and content
  - Use a specific mail client such as mutt
    (no gmail!)

Now wait for your feedback,
And do it all again…

Happy Hacking!

THE LINUX FOUNDATION

Get the slides

- Visit https://github.com/shegeek/kernel-hacking-101

- $ git clone https://github.com/shegeek/kernel-hacking-101.git

Resources

- Gnome Outreach Program for Women https://wiki.gnome.org/OutreachProgramForWomen
- Kernel.org git repositories https://git.kernel.org/cgit/
- Linux Kernel Newbies http://kernelnewbies.org/
- OPW Intro page http://kernelnewbies.org/OPWIntro
- The Eudyptula Challenge http://eudyptula-challenge.org/
- KVM Installation https://help.ubuntu.com/community/KVM/Installation
- Linux Kernel Mailing List https://lkml.org/
- Various Linux related mailing lists at Gmane http://gmane.org/find.php?list=kernel
- Linux Foundation events http://events.linuxfoundation.org/
- Linux Weekly News http://lwn.net/
- #irchelp http://www.irchelp.org/
- Irssi – The Client of the Future http://www.irssi.org/
- Wikipedia entry for the grep command http://en.wikipedia.org/wiki/Grep
- Linux Cross Reference at Free Electrons http://lxr.free-electrons.com/
- Linux Kernel Coding Style (pdf) https://computing.llnl.gov/linux/slurm/coding_style.pdf
- Vim the editor http://www.vim.org/
- Pro Git http://git-scm.com/book https://www.gitbook.io/book/gitbookio/progit
- The Mutt E-Mail Client http://www.mutt.org/

Image Credits

- Arrows Circle by Freepik CC BY 3.0
- Communication shannon-weaver2 by Einar Faanes CC BY-SA 3.0
- Xaric screen shot by Triddle BSD License http://commons.wikimedia.org/wiki/File:Xaric_screen_shot.jpg
- Coding All Night Long by Snatcherdudette http://snatcherdudette.deviantart.com/art/Coding-all-night-long-183815498
- Music present by Marta Crowe Creative Commons Attribution 2.0 Generic
- Linux Foundation Logo is in the public domain
- Linux “Tux” Logo is in the public domain