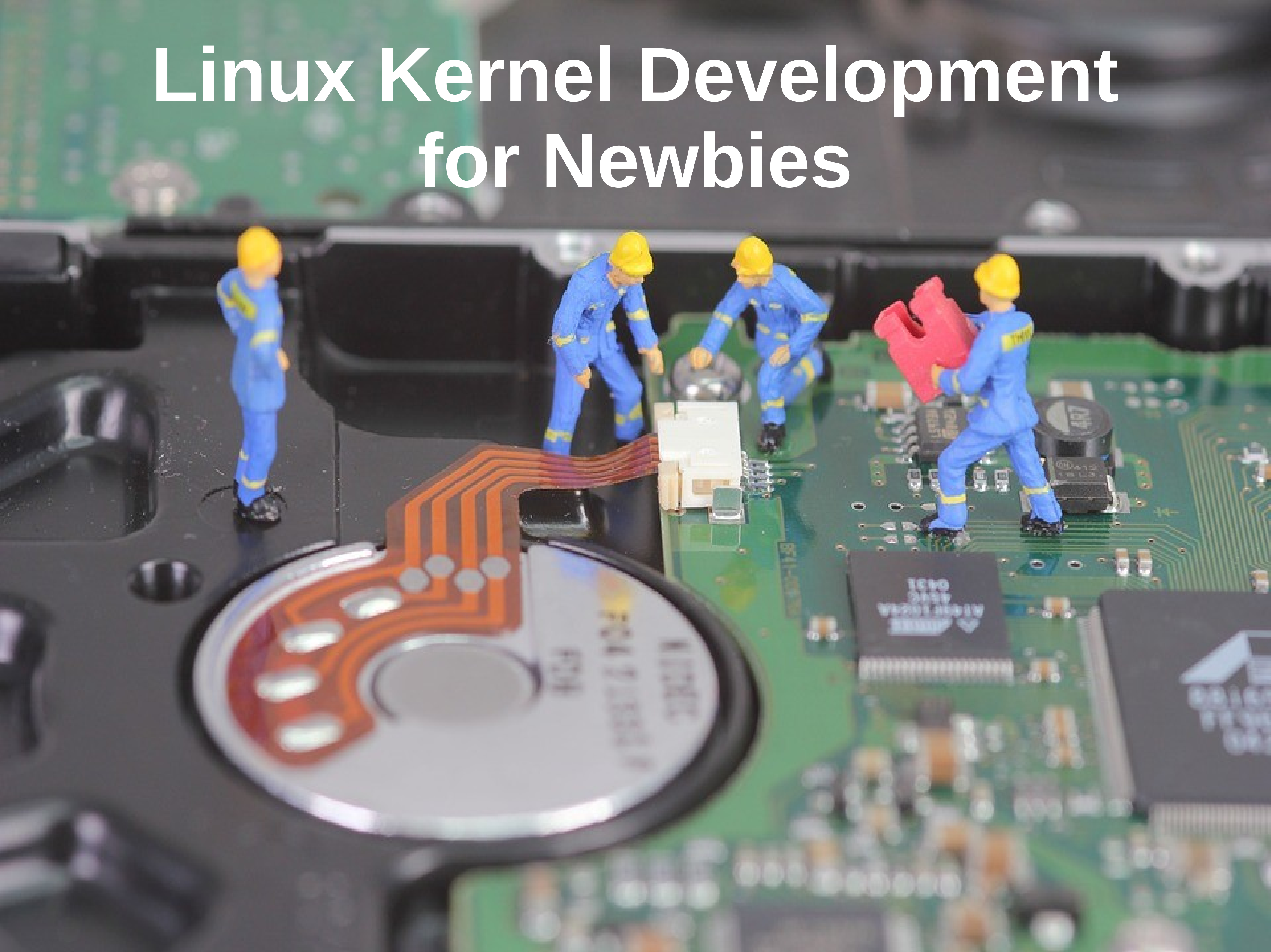
Linux Kernel Development for Newbies



Linux Release Cycle

- New kernel release every 9-10 weeks
- Merge window for new development: ~2 weeks
  - Linus merges ~1,000 commits a day
- Bug fixing/hardening: ~7-8 weeks
  - New release candidate build each week
- Meanwhile ongoing new development in '-next' repos
- 5.x is released and -stable branch created
  - New stable release every ~1 week (15-20 total)

Sub-system maintainers

- Kernel code is fairly modular
- Split up into 200-300 different maintainer git trees
  - bug-fix trees go into stable
  - '-next' trees go into the next merge window
  - Details all in ./MAINTAINERS file
- Linux-next: Merge of ~300 trees recreated each day
  - Used for Integration testing

Basic development process

- Get the code
- Configure
- Build it
- Make mistakes
- Test it
- Fix most of the mistakes
- Deliver it

1. Get the code

- Kernel repos located at:
  git://git.kernel.org/pub/scm/linux/kernel/git/
- For bug fixes or general tinkering, use stable, i.e.

```
git clone git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git
git checkout linux-5.2.y
```

- For testing/finding bugs, use Linus's repo, i.e. torvalds/linux.git
- For new development, use the relevant sub-system tree
  e.g. for network protocols, use davem/net-next.git

2. Configure

- Each driver/feature has a corresponding Kconfig option
- The .config file controls what drivers/features to build, e.g.

```
CONFIG_NAMESPACES=y
# CONFIG_CRYPTO_HW is not set
CONFIG_EMBEDDED=y
...
```

- To create the .config:

```
cp /boot/config-$(uname -r) .config
make menuconfig  # save and exit
```

3. Build it

```
make -j $(nproc)
```

- Note: build system dependencies

```
sudo apt-get install build-essential libncurses-dev \
      bison flex libssl-dev libelf-dev
```

4. Install it

- To install and run the compiled kernel:

```
sudo make modules_install
sudo make install
reboot
```

- Extra steps needed if your distro doesn't support an `installkernel` script
- Familiarize yourself with the bootloader (i.e. GRUB) first
- Maybe best not to do this on a work PC...

5. Test it

- Be generous with VMs: 4Gb RAM, 40Gb disk space, multiple cores
- Install the Ubuntu Mainline Kernel PPA for the UI
- Familiarize yourself with dropping to the shell

  E.g. on VirtualBox: right-ctrl + F2

6. Deliver it

- Read the documentation first, e.g.
  ./Documentation/process/coding-style.rst
  ./Documentation/process/5.Posting.rst
- `./scripts/checkpatch.pl` checks your patch for mistakes
  `./scripts/get_maintainer.pl` works out maintainers
- Use ./MAINTAINERS to work out relevant sub-system/tree
  - Always sanity-check your patch applies and compiles

git send-email

- Install it and set it up

```
sudo apt-get install git-email
git config --global sendemail.smtpuser <your-email>
git config --global sendemail.smtpserver <mail-server>
git config --global sendemail.smtpencryption <ssl-or-tls>
git config --global sendemail.smtpserverport <smtp-port>
```

- Note that `--smtp-encryption=tls` means STARTTLS
- git send-email ***will*** prompt you for your email password
- Use `--smtp-debug=1` if necessary

Where to get started…

- Read the docs
  - ./Documentation/process/howto.rst
  - Linux In a Nutshell. Free copy: www.kroah.com/lkn
  - https://kernelnewbies.org/FAQ/WhereDoIBegin
- Eudyptula (Little penguin) Challenge
  - 20 tasks: https://github.com/agelastic/eudyptula
  - Ran 2014-2017. 19,000+ participants. Only 149 finished
- Test the latest kernel and find bugs
- Look for a bug you might be able to fix
- Find an area of code that interests you

Questions?