

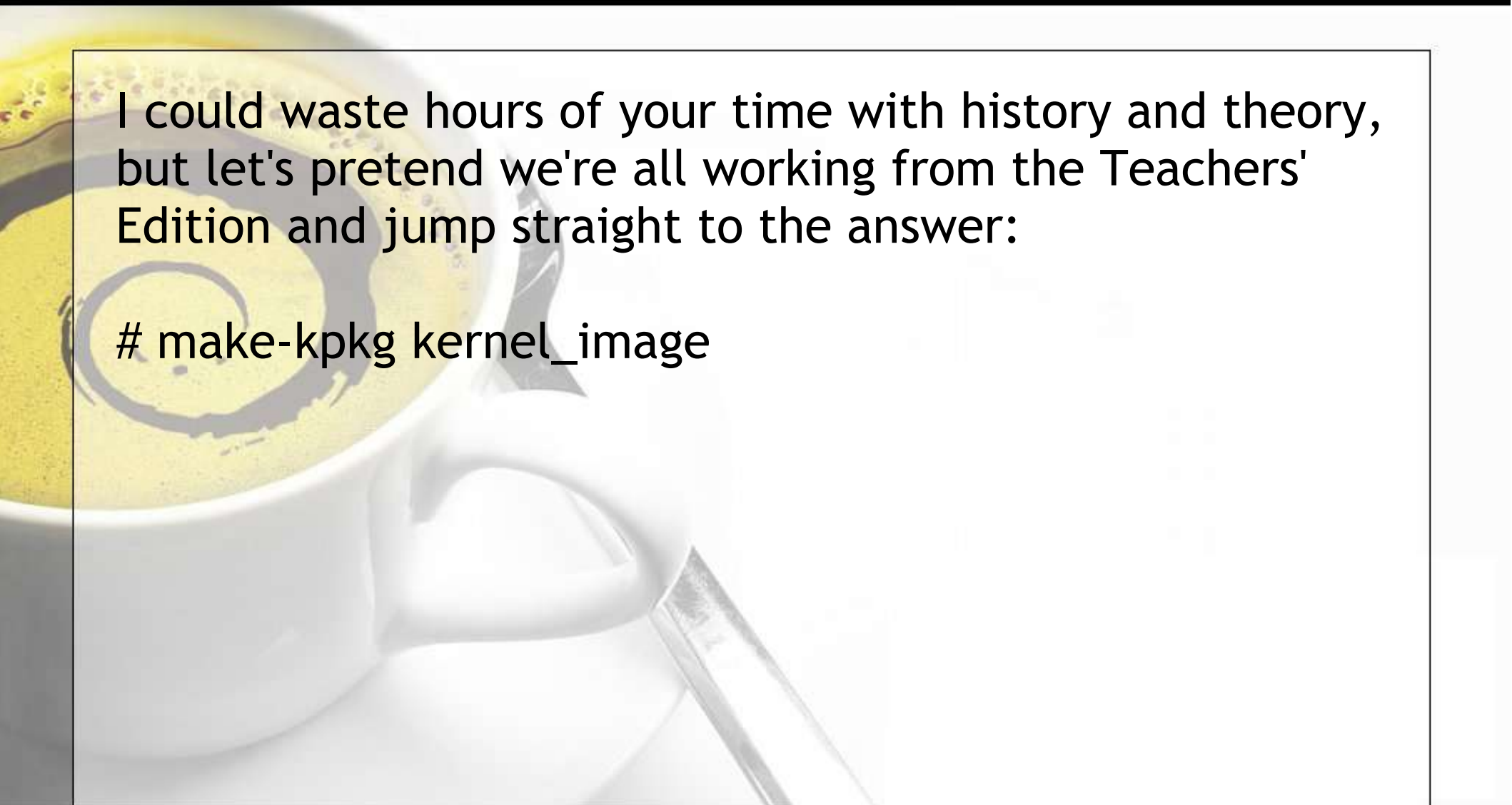


Compiling Kernels The Debian Way

Jonathan Oxe
LinuxTag 2004, Karlsruhe

debian

Let's Cheat: Starting At The End



I could waste hours of your time with history and theory, but let's pretend we're all working from the Teachers' Edition and jump straight to the answer:

```
# make-kpkg kernel_image
```

Getting The Source

- Get pristine source from kernel.org

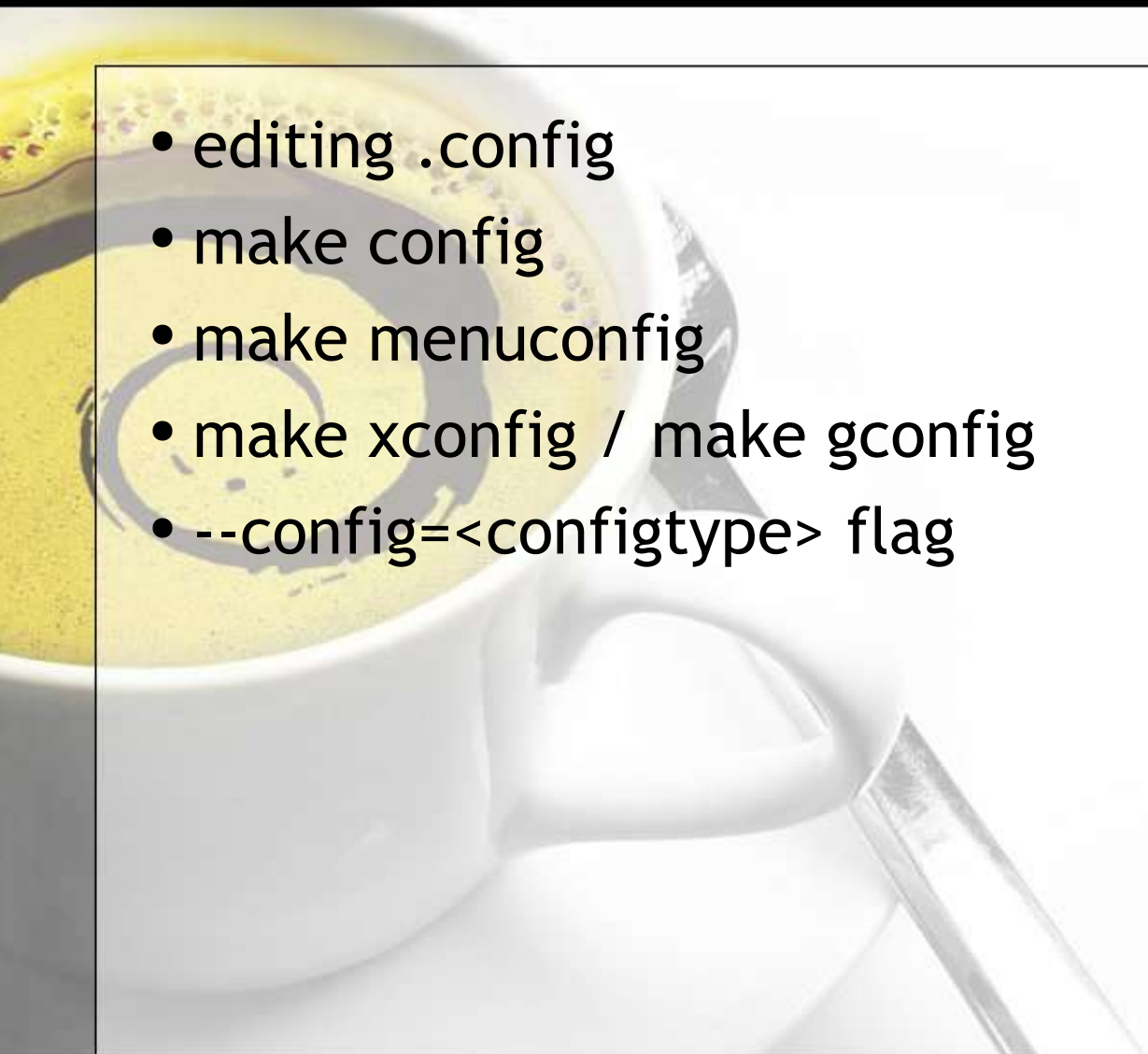
OR

- Use a Debian kernel source package:
apt-cache search kernel-source
apt-get install kernel-source-2.4.23
cd /usr/src
tar -jxf kernel-source-2.4.23.tar.bz2
ln -s kernel-source-2.4.23 linux
cd /usr/src/linux

Tools You Will Need

- Basic tools are in the 'kernel-package' package ;-)
apt-get install kernel-package
- Extra tools for configuration etc:
apt-get install libncurses5-dev tk8.3
- Module tools for 2.6 kernels:
apt-get install module-init-tools

Configuration Methods

- 
- editing `.config`
 - `make config`
 - `make menuconfig`
 - `make xconfig` / `make gconfig`
 - `--config=<configtype> flag`

Migrating Your Configuration



Moving to newer kernel source loses all your previous configuration work. Ouch!

Copy `'.config'` to your new kernel source tree, then do `'make oldconfig'` to see just the options that have changed between kernel versions.

Compiling And Packaging

At last, here's the fun bit:

```
# make-kpkg --append-to-version=-jon20-pm kernel_image
```

Then wait. Then make a coffee. Then wait some more.

Finally:

```
# cd ..  
# ls -l
```

Woohoo!

Installing Your Kernel Package

Install the kernel package just like any other package:

```
# dpkg -i kernel-image-2.4.23-jon20-pm_123.Custom_i386.deb
```

Man, that was hard! Not.

This package can now be copied around to other machines you want to run it on, and just installed like any other package. All the modules etc are installed automagically by dpkg.

Updating LILO and GRUB

The kernel package will search for and try to automatically configure LILO and GRUB to recognise the new kernel image.

If you have to do it manually, check the LILO config in:

`/etc/lilo.conf`

and then run 'lilo' to activate the new config.

Likewise for GRUB, check the config in:

`/boot/grub/menu.lst`

Reboot And Test

Reboot, and you should see the new kernel appear as an option in the LILO or GRUB menu.

Once booted you can check the currently running kernel revision either by taking a peek in:

```
/var/log/dmesg
```

or running 'uname -r'.

If some things didn't work out as planned, look through dmesg output to see if anything failed to load.

Random Junk

- You can set revision numbers using the `--revision` flag.
- Between builds you may need to do a cleanup if you've changed the package version etc. Use:
`# make-kpkg clean`
- Building as a normal user:
`--rootcmd fakeroot`
- Using external module packages:
`--added-modules <module1>,<module2>`

More Random Junk

- The .configs of currently installed kernels are stuck in:
/boot/config-<kernel-version>

Very handy to find what options are set in the current kernel! As a useful trick when starting with clean kernel source, you can just do:

```
# cp /boot/config-`uname -r` /usr/src/linux/.config
```

- Alternative build targets include 'modules_image', etc.

Run:

```
# make-kpkg -targets  
to see a list.
```

Blatant Advertising

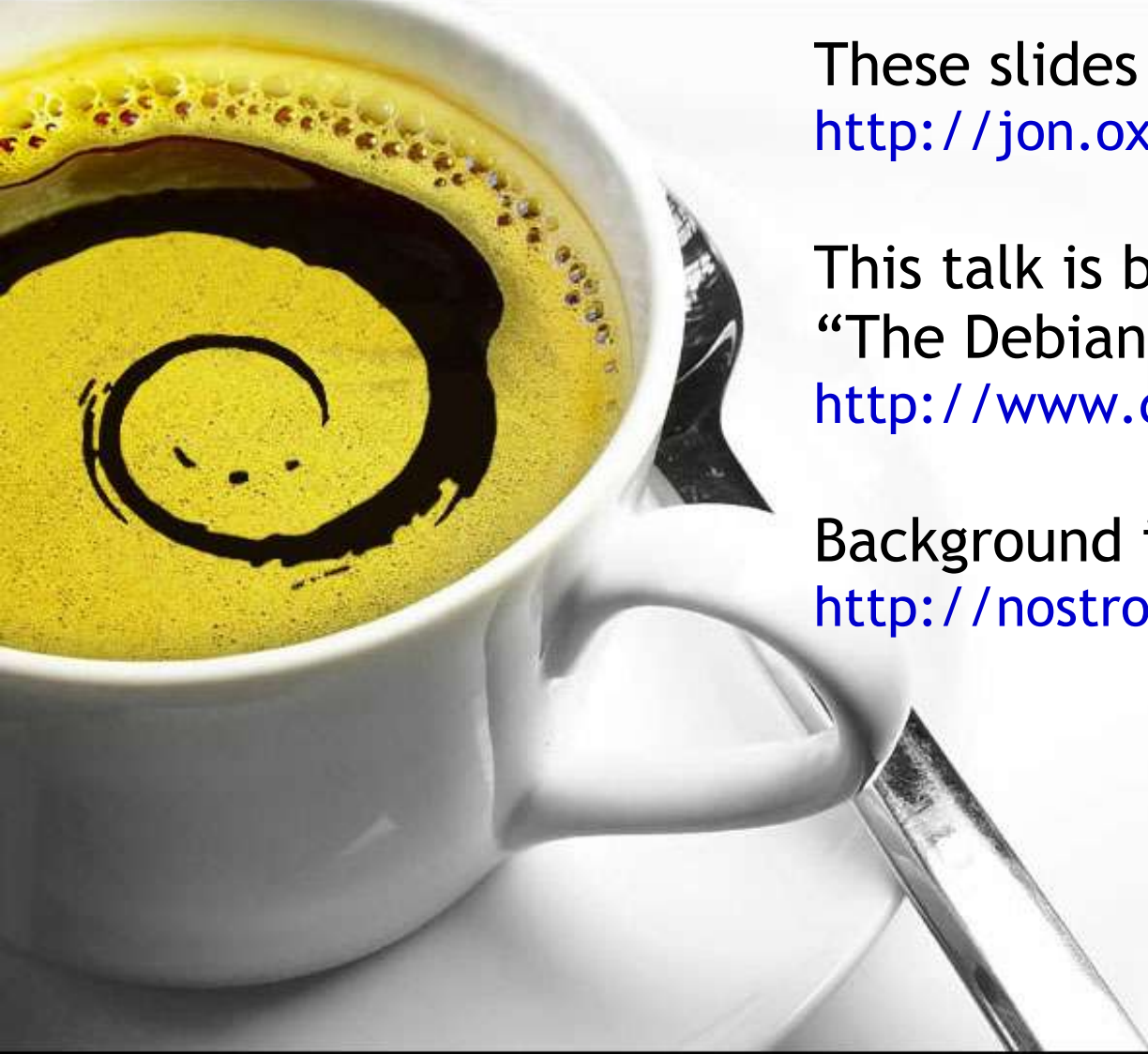


I'm currently working on “Running Debian/GNU Linux” for O'Reilly, which will contain lots more information about kernel packaging plus about 1000 pages of other handy Debian info.

It should be out late this year.

Buy a copy! Buy two! Tell all your friends!

More Information



These slides are online at:
<http://jon.oxer.com.au/talks/>

This talk is based on a chapter from
“The Debian Universe”:
<http://www.debianuniverse.com/>

Background image by Andre Gonzalez:
<http://nostromo2k3.deviantart.com/gallery/>

debian