

Linux Build Infrastructure



ECE 373

You want to build a kernel...

- First you need one...
- What is upstream? What is distro?
- Useful to tune for different installations
- Include your driver in the build

Prep the machine

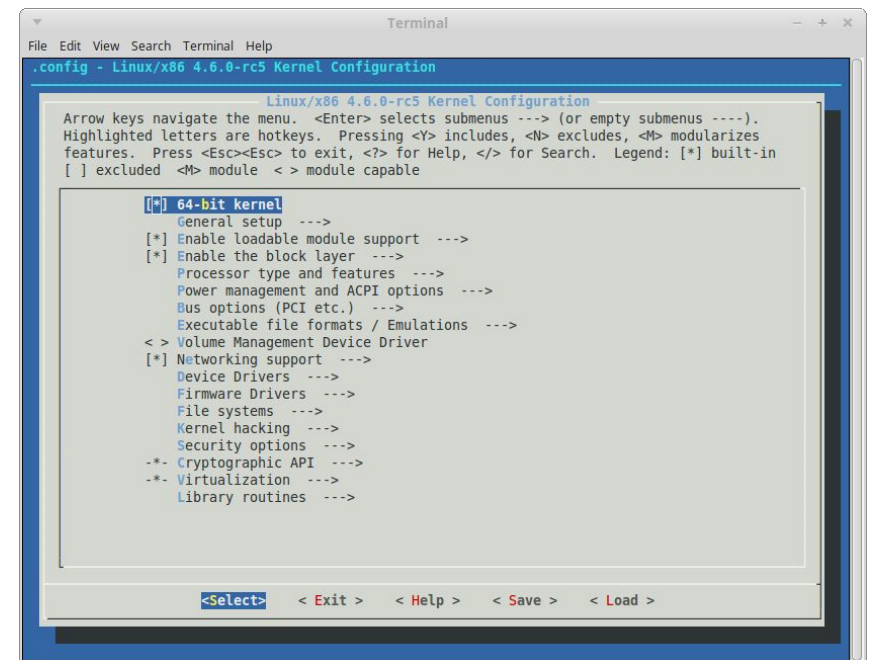
- Download code
 - <http://kernel.org> and unpack the kernel code archive
 - or
 - git <https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/>
 - or
 - apt-get source linux-image-\$(uname -r)
- Get distro build headers
 - Ubuntu: apt-get install linux-headers-generic libssl-dev
 - Fedora/RedHat/Centos yum install kernel-headers kernel-devel openssl-devel
- Start with distro config for most likely success
 - Copy from /boot/config... to .config
 - Make olddefconfig
- https://fedoraproject.org/wiki/Building_a_custom_kernel
- <https://wiki.ubuntu.com/Kernel/BuildYourOwnKernel>

How to configure a kernel

- Start with an existing .config that you know works
 - /boot/config-3.13.0-37-generic or
 - /usr/src/linux-headers-3.13.0-37-generic/.config
 - 'make olddefconfig' to update it

- Tweak the config for your needs
 - Interactive terminal 'make config'
 - Curses-based 'make menuconfig'
 - GUI-based 'make gconfig'

- Or just edit .config directly



- Where does this all come from?

The Kconfig framework

- Infrastructure to enable/disable kernel features
- Source directory structure
- Used to manipulate makefiles
- Layered, like an onion (and stinky too!)
- Can implement multiple dependencies



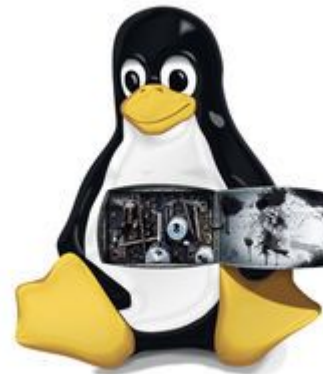
Building and booting a kernel

- make, make modules_install, make install
- GRUB, various bootloaders
- vmlinuz and vmlinux images
- Modules installation
- Initial RAM disks (initrd)
- Debian pkgs for other machine
 - make deb-pkg



Add your own code!

- New driver directory – where?
- Editing/adding Kconfig for inclusion
- Creating your makefile



Wrap-up

- Practice makes perfect
- Trimming the fat for embedded
- Dealing with the bootloader
- Dropping code into the kernel