Development environment setup

Carlos Maiolino

Michal Schmidt

Rado Vrbosky

Vratislav Bendel

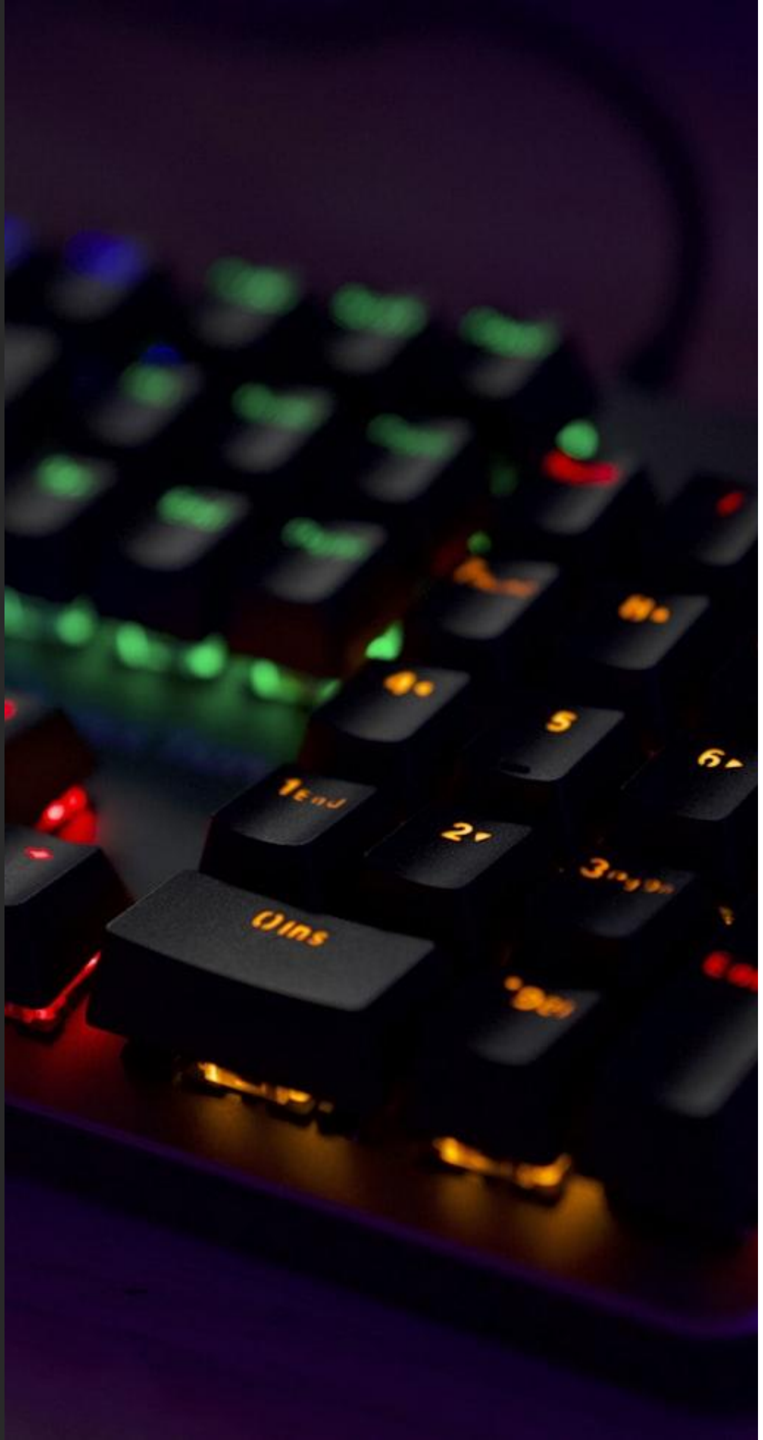
Leonardo Vaz

Overview

- Development and testing machines
- Development tools
- Kernel configuration
- Installing the Linux kernel
- Patch formatting and submission

Disclaimer: Have your own way

- Every developer has his/her own way to work
- Don’t take the instructions here as hardcoded
- The only requirements are:
  - You submit patches in the proper format
  - Your patches apply to the specified repository
  - Your patches build
  - (Strongly encouraged) It works as expected



Development and testing machines

Why two different machines?

- If you screw up your code, you don’t lose your dev environment
- You can use bare-metal for development if you have a Linux machine
- We recommend Fedora, but you can use any distro as long as you know how to use it.
- Why Fedora?
  - Bleeding edge tools available (we don’t need to compile anything other than Linux itself).
  - We know how the package manager works

Development machine

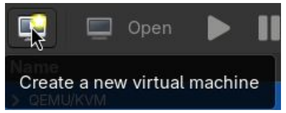
- Can be your bare-metal machine if you use Linux
- If you don’t use Linux, you will need to install a virtual machine
- Setup the development environment (more on this later)
- NFS server: Easy way to build the kernel in one place and install in another

Test machine

- A Linux machine where we will install and test the Linux kernel
- Don’t need to be powerful
- At least 2 vCPUs would be great so we can use SMP
- As much memory as you have available
- NFS client

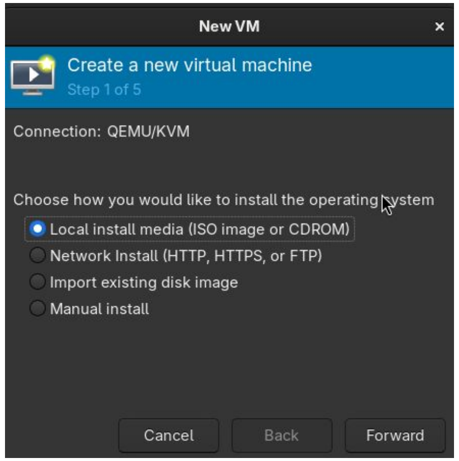
VM setup example

- virt-manager (qemu, kvm, libvirt, .. virsh)
- … or anything else



Create a new virtual machine

Don’t underestimate storage
go 40+ GB

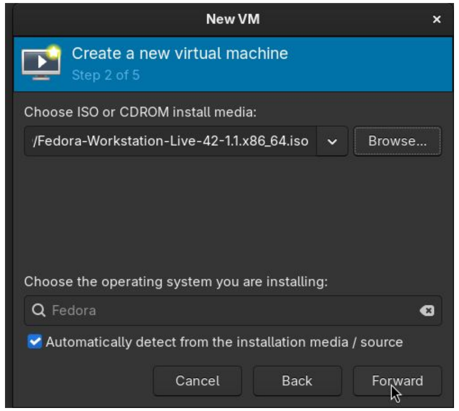


Create a new virtual machine
Step 1 of 5

Connection: QEMU/KVM

Choose how you would like to install the operating system

● Local install media (ISO image or CDROM)
○ Network Install (HTTP, HTTPS, or FTP)
○ Import existing disk image
○ Manual install

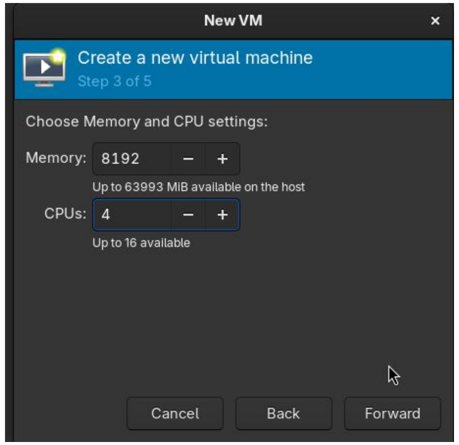


Create a new virtual machine
Step 2 of 5

Choose ISO or CDROM install media:
/Fedora-Workstation-Live-42-1.1.x86_64.iso

Choose the operating system you are installing:
Fedora

☑ Automatically detect from the installation media / source

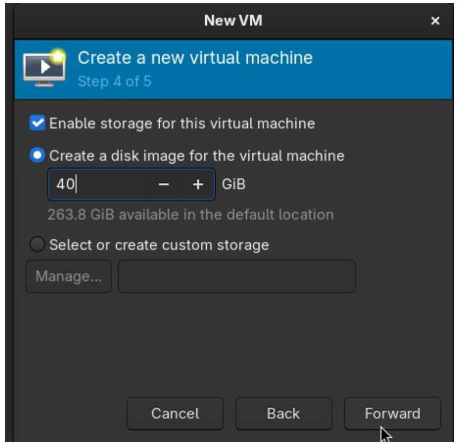


Create a new virtual machine
Step 3 of 5

Choose Memory and CPU settings:

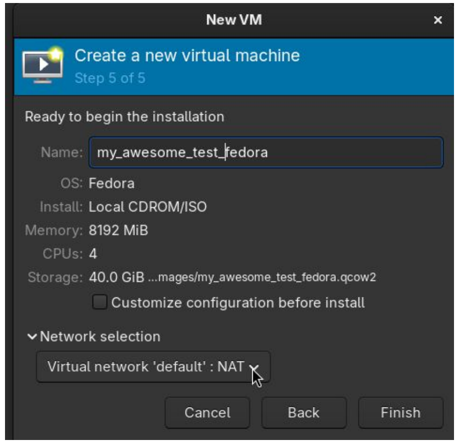
Memory: 8192 − +
Up to 63993 MiB available on the host

CPUs: 4 − +
Up to 16 available



Create a new virtual machine
Step 4 of 5

☑ Enable storage for this virtual machine
● Create a disk image for the virtual machine
40 − + GiB
263.8 GiB available in the default location
○ Select or create custom storage



Create a new virtual machine
Step 5 of 5

Ready to begin the installation

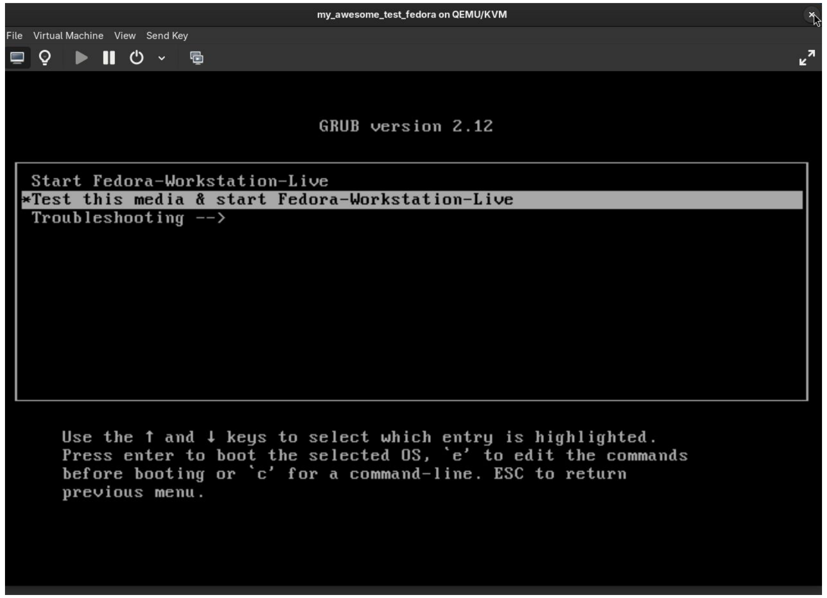
Name: my_awesome_test_fedora
OS: Fedora
Install: Local CDROM/ISO
Memory: 8192 MiB
CPUs: 4
Storage: 40.0 GiB ...mages/my_awesome_test_fedora.qcow2
□ Customize configuration before install

Network selection
Virtual network 'default' : NAT

For build VM
go full RAM and CPUs

Default network suffice

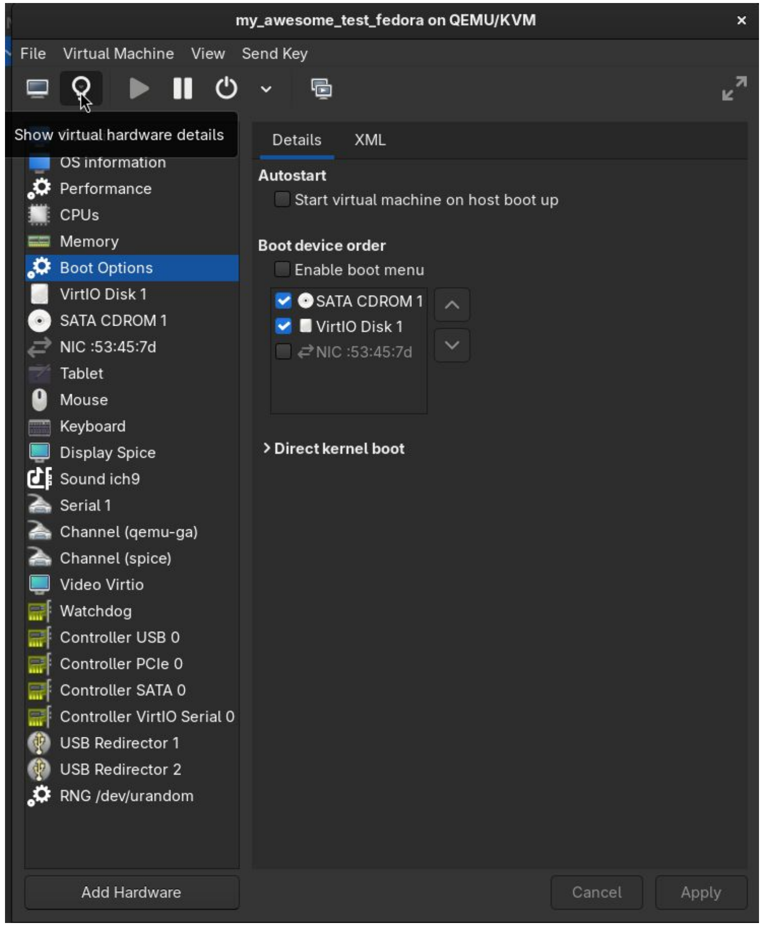
VM setup example



```
GRUB version 2.12

 Start Fedora-Workstation-Live
*Test this media & start Fedora-Workstation-Live
 Troubleshooting -->

   Use the ↑ and ↓ keys to select which entry is highlighted.
   Press enter to boot the selected OS, `e' to edit the commands
   before booting or `c' for a command-line. ESC to return
   previous menu.
```



Autostart

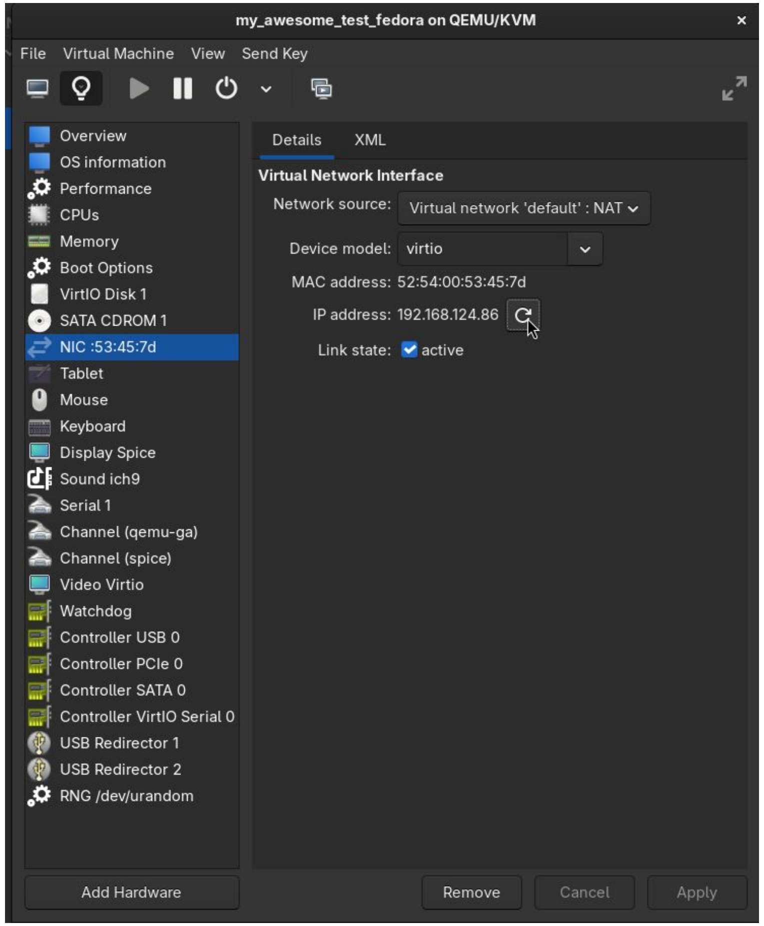
☐ Start virtual machine on host boot up

Boot device order

☐ Enable boot menu

☑ SATA CDROM 1
☑ VirtIO Disk 1
☐ NIC :53:45:7d

› **Direct kernel boot**



Virtual Network Interface

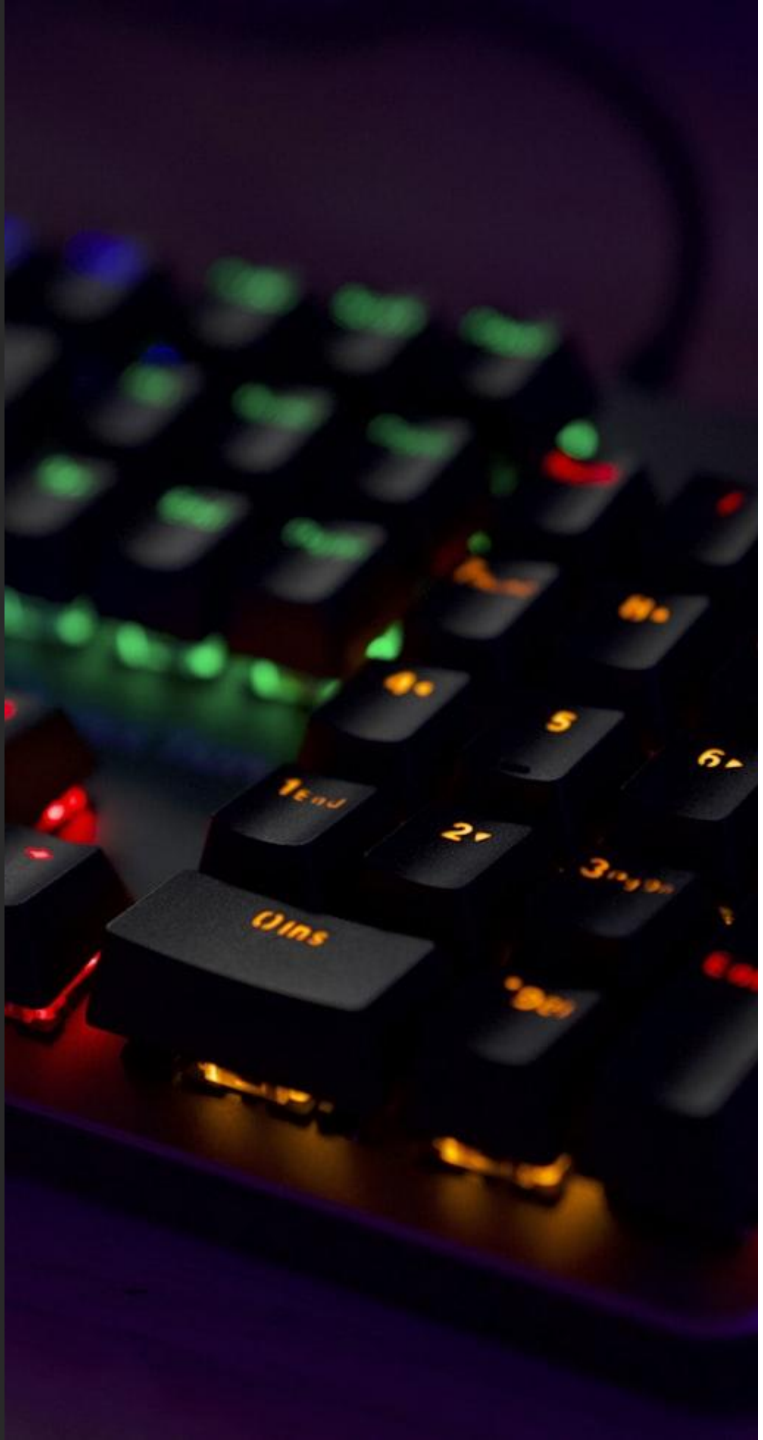
Network source: Virtual network 'default' : NAT

Device model: virtio

MAC address: 52:54:00:53:45:7d

IP address: 192.168.124.86

Link state: ☑ active



Development Tools

Some useful tools

- git (mandatory)
- Linux source tree (of course)
- Compiler (gcc, clang)
- code editor (vim, emacs, whatever else you want to use)
- Navigation tools
- Debug tools (To be discussed later)

Obtaining the source code

- Linux development is split into the main tree and subsystem trees
- We will use Linus’s main tree for the purposes of the course
    - Your local copy should be cloned from Linus’s tree
    - Check-out v6.16 tag.

git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git

Linux Kernel Flavours

Maintainer Subtree

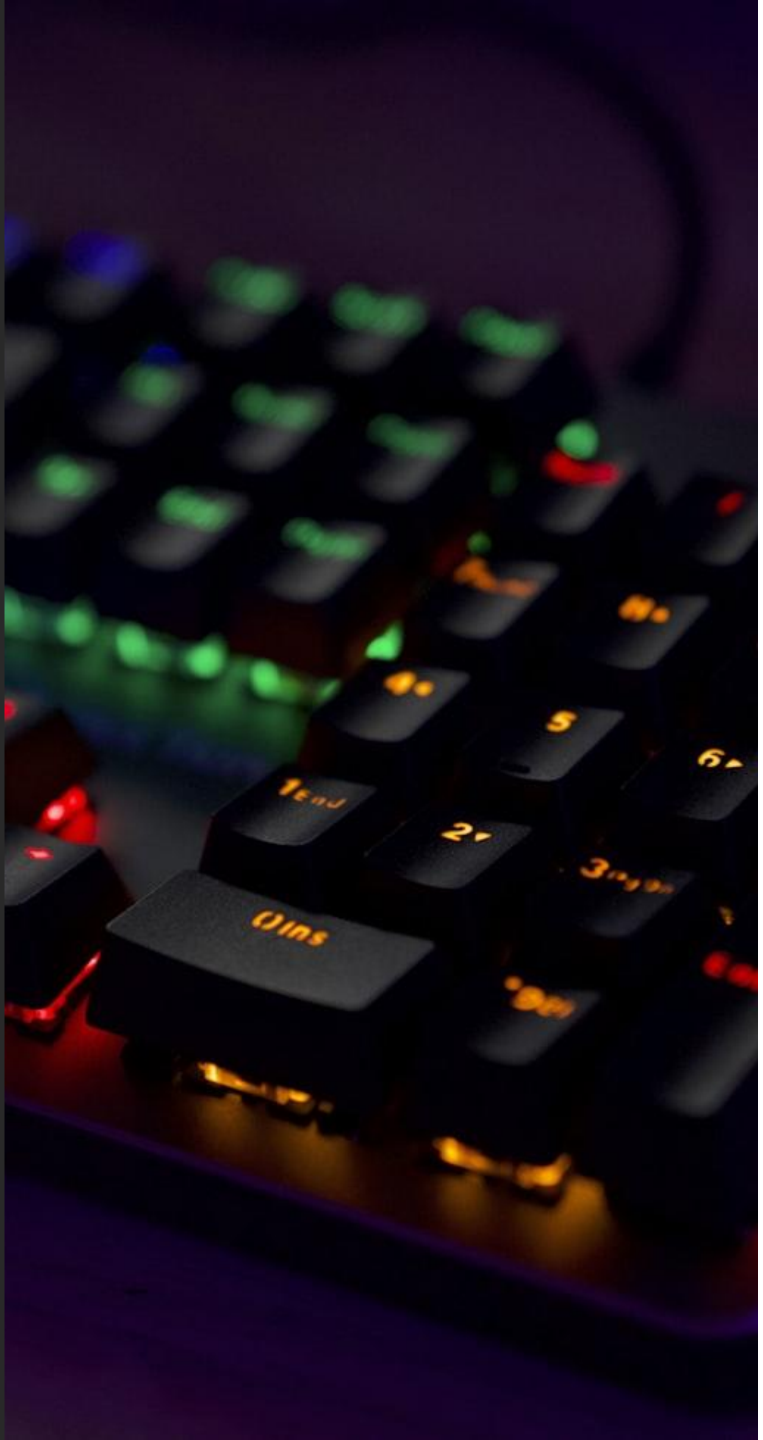
Receive and merge patches related to a specific subsystem or subcomponent

Linux-Next tree

Aggregate bleeding edge patches, usually used to test before merging into mainline

Vanilla

Linus' main tree, the end point of all Linux’s patches.

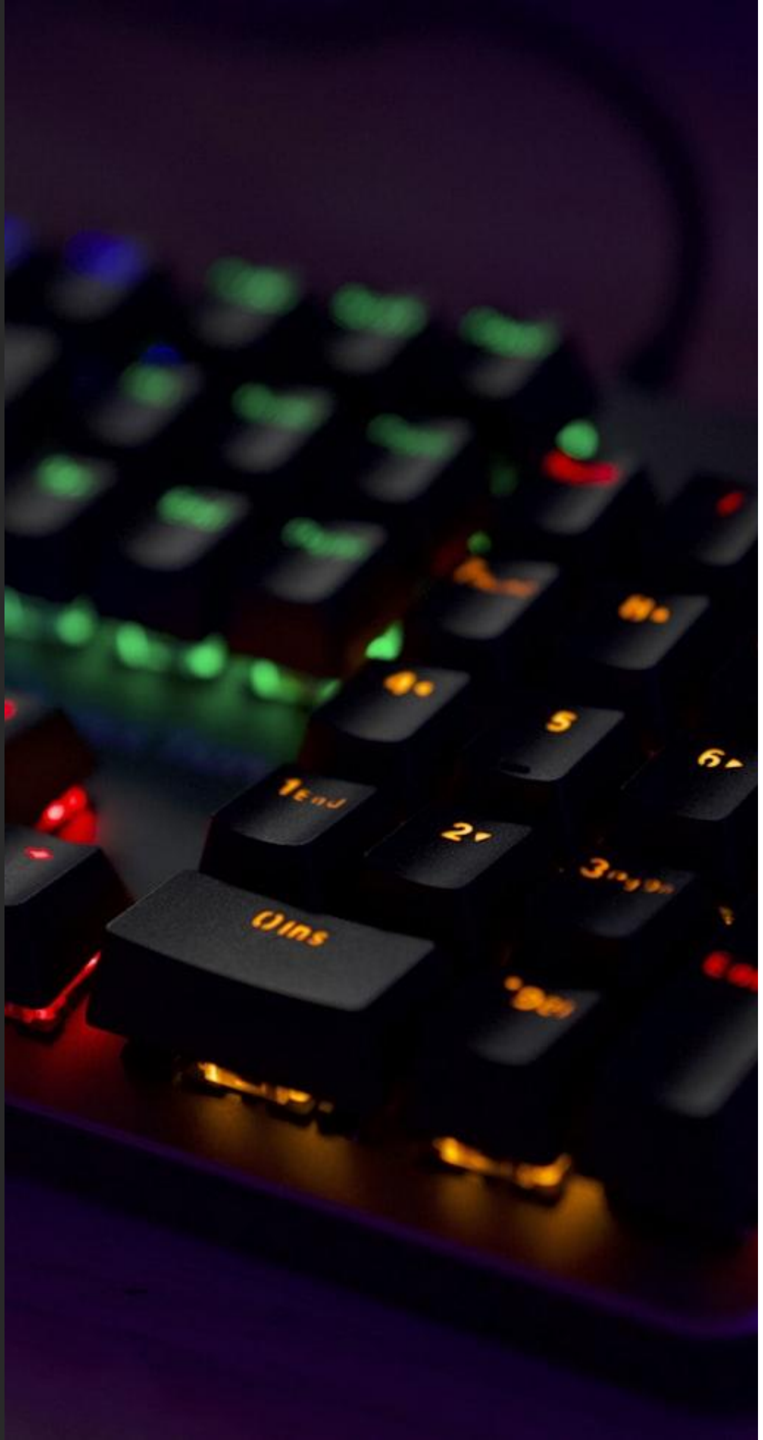


Quick look into kernel configuration

Red Hat

Configure your kernel

- How the kernel `.config` file works
- How to create the config file the easy way
  - Copy from a distribution and change it
  - Use kernel config generator
  - Graphical tools (xconfig, gconfig, nconfig, menuconfig)
- The hard way: `$make config`
- The spartan way: write the `.config` yourself!



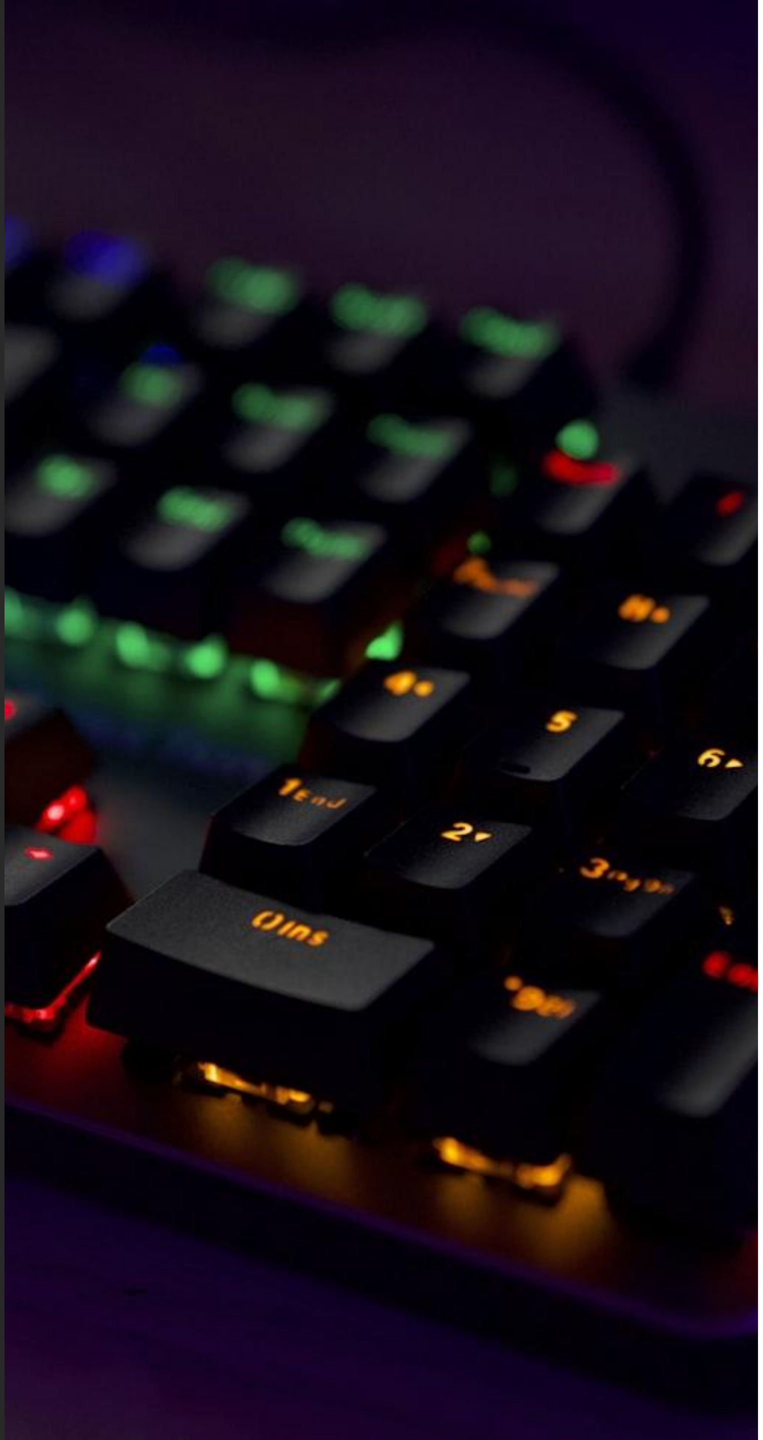
Building and installing the kernel

Building

- In order to build the kernel, the .config should be ready
  - You can tweak the version if you want (see `localversion` file)
- Distribution package vs standard build vs Tarball
  - Run `$ make help` and look for the options
- Run `$ make -jX >/dev/null` to start building the kernel
  - Where **X** depends on how many CPUs you have available
- Wait a long time
- Hope for no errors (otherwise you'll need to start it over).

Installing

- Transferring the built kernel image to the test machine
  - Copying the package
  - Packaging the executables (kernel image/modules) and copying them
  - Accessing the dev environment via NFS
- The development environment should be the same architecture
  - Unless you are cross-compiling a kernel for a different architecture
- Make sure your kernel is finally bootable
  - Disabling graphical boot and enabling console helps



Browsing the Kernel Tree

Kernel Directory Structure

Documentation/

scripts/

tools/

MAINTAINERS

README

arch/

crypto/

include/

kernel/

lib/

block/

drivers/

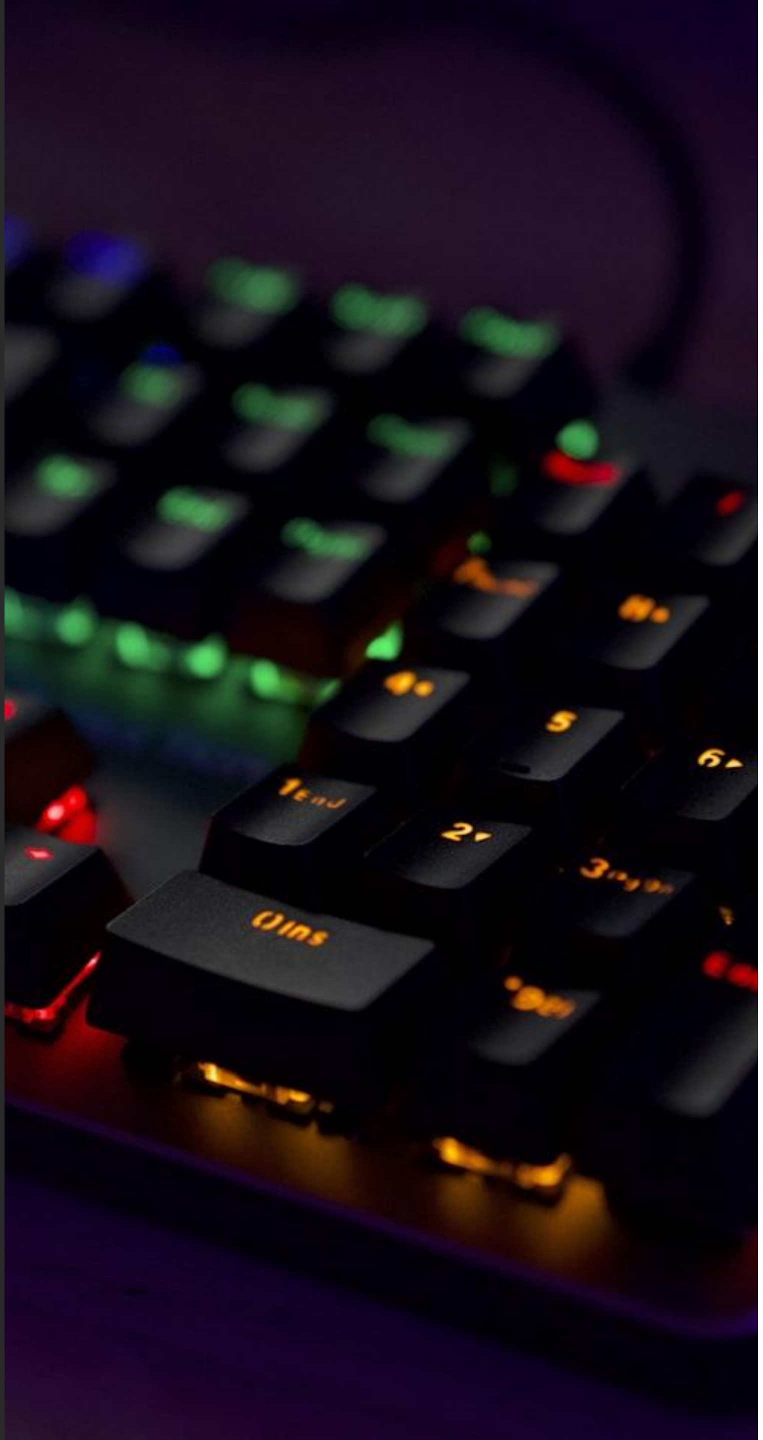
fs/

mm/

net/

virt/

https://makelinux.github.io/kernel/map/



Environment examples

Carlos

- Git
  - git-worktree
  - guilt
- tmux
- vim + nerdtree + tagbar
- cscope
- neomutt

Rado

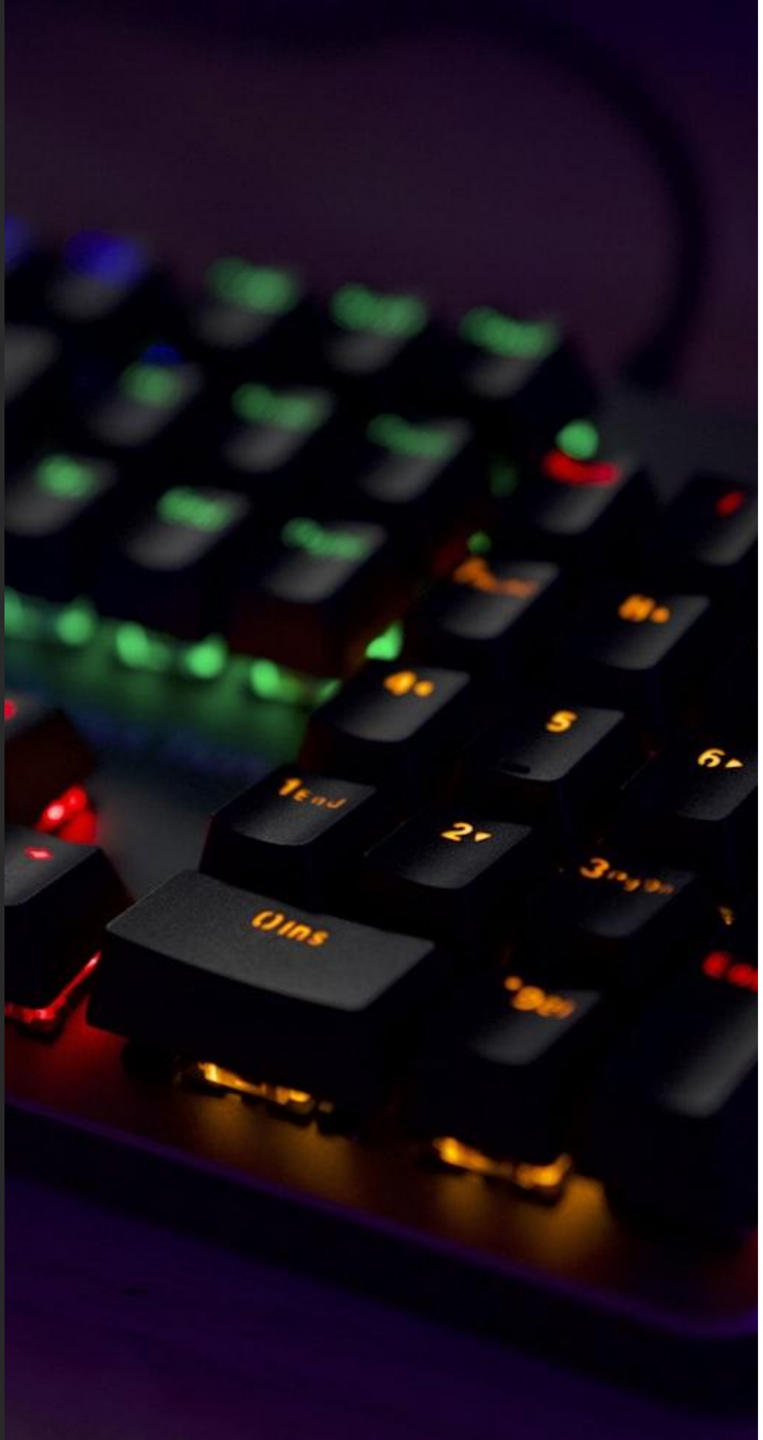
- git
- screen
- vim + nerdtree + tagbar
- make tags, grep
- mutt, gitlab
- qemu

Vraťo

- git
- vim (quite raw TBH)
- cscope, gtags, grep
- perf, trace-cmd, systemtap
- bash & python scripts

Michal

- git (with ~8 git-worktree trees)
- vim
- cscope, grep
- bpftrace, trace-cmd, systemtap
- bash & python scripts



Linux coding style and patch submission process

Linux Kernel coding style

- Linux maintainers are strict regarding coding style
- Make sure your code follows it
- There are tools for checking the code style
  - Coding style check script (`scripts/checkpatch.pl`)
  - vim plugin (if you use vim)
- Coding Style in the following URL:
  https://docs.kernel.org/process/coding-style.html

Prepare your patch for submission

- Avoid heated discussions in the mailings
- Make sure that
  - Your patch applies against the tree you are submitting it
  - It builds
  - The kernel boots and it doesn't crash the system immediately
- Beginner friendly tool: `$ git format-patch`

https://git-scm.com/docs/git-format-patch

Where should we send the patch?

- Look for a mailing list related to what you are changing
  - Most of the time, patches are not submitted against the main tree
- Make sure your patch is tested on the right tree before submitting
- Use `scripts/get_maintainer.pl` to find the subsystem maintainer

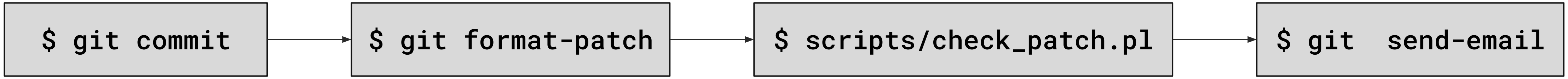
https://www.kernel.org/doc/html/latest/process/submitting-patches.html

Send it!

- Linux upstream community is email based
- You can use `git send-email`
- Configure your `~/.gitconfig` to submit patches
- See the documentation for examples
- DON'T SEND PATCHES AS ATTACHMENT
- DON'T SEND EMAILS ON ANY FORMAT OTHER THAN text/plain

https://git-scm.com/docs/git-send-email

Recap



```
$ git commit
```

```
$ git format-patch
```

```
$ scripts/check_patch.pl
```

```
$ git  send-email
```

Step 1:
Commit your
changes

Step 2:
Extract the patch
to a file

Step 3:
Check the
Coding Style

Step 4:
Submit the
Patch

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat