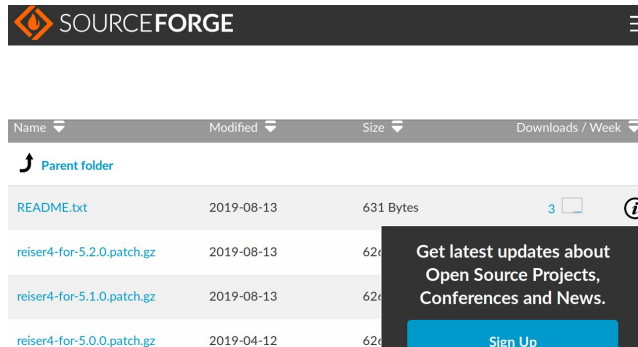


## Build a new kernel

**Grab the latest stable release available from kernel . org, Grab some patch e.g. (exempli grātiā) support for Reiser4 and apply it**

I grabbed this one stable 5.2.17 because last patch for the reiser4 was for this version:



Patched using `$patch -p1 < path/to/patch`

**What make targets are interesting to you and why?**

It is possible to get the targets with `$make help`:

```
banana@banana-VirtualBox:/usr/src/linux$ make help
Cleaning targets:
  clean          - Remove most generated files but keep the config and
                   enough build support to build external modules
  mrproper       - Remove all generated files + config + various backu
p files
  distclean      - mrproper + remove editor backup and patch files

Configuration targets:
  config         - Update current config utilising a line-oriented pro
gram
  nconfig       - Update current config utilising a ncurses menu base
d program
  menuconfig    - Update current config utilising a menu based progra
m
  xconfig       - Update current config utilising a Qt based front-en
d
  gconfig       - Update current config utilising a GTK+ based front-
end
  oldconfig     - Update current config utilising a provided .config
as base
```

Although all of them are pretty interesting and helpful I find config family and install the most useful, because we can control the building process through it.

**Use your distribution's . config as a starting point and the olddefconfig target. Bonus: use Slackware-current's or another distros' config as a starting point, even on CentOS or Ubuntu**

Used configs grabbed configs from here:

[https://mirrors.slackware.com/slackware/slackware-14.2/source/k/config-x86\\_64/config-generic-4.4.14.x64](https://mirrors.slackware.com/slackware/slackware-14.2/source/k/config-x86_64/config-generic-4.4.14.x64)

**Make sure some options are enabled either built-in or as module:**

Found missing options using [this script](#)

Added them to the .config file with =y

**Also enable your patched feature and show how you did it**

Added line in .config file : CONFIG\_REISER4\_FS=y

**How many cores do you have available?**

4 cores:

```
banana@banana-VirtualBox:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 4
On-line CPU(s) list:   0-3
Thread(s) per core:     1
Core(s) per socket:     4
Socket(s):              1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  142
Model name:             Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
Stepping:               10
CPU MHz:                1991.999
BogoMIPS:               3983.99
Hypervisor vendor:      KVM
Virtualization type:    full
L1d cache:              32K
L1i cache:              32K
L2 cache:               256K
L3 cache:               8192K
```

**Build the kernel with -jCORES+1, deploy it and install its afferent modules**

Used:

\$make -j 4

\$ sudo make modules\_install

\$ sudo make install

**In what booting mode is your computer, BIOS/CSM or EFI? How did you find out?  
What boot-loader are you currently using? Eventually add your new kernel in the boot-menu**

Used \$[ -d /sys/firmware/efi ] && echo UEFI || echo BIOS to determine if booting mode, the output in my case is bios:

```
banana@banana-VirtualBox:~$ [ -d /sys/firmware/efi ] && echo UEFI || echo BIOS
BIOS
banana@banana-VirtualBox:~$
```

Bootloader:

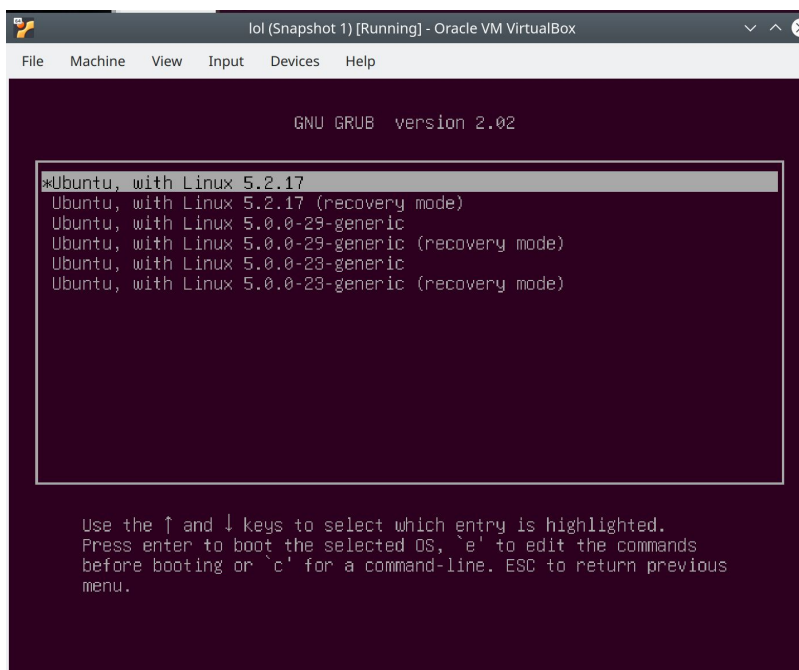
```
banana@banana-VirtualBox:~$ grub-install --version
grub-install (GRUB) 2.02-2ubuntu8.13
```

Change grub config:

```
$ sudo update-initramfs -c -k 5.2.15
```

```
$ sudo update-grub
```

New kernel is in the GRUB menu:



- **Check that you're now running the newly built kernel**

```
banana@banana-VirtualBox:~/Downloads$ uname -mrs
Linux 5.2.17 x86_64
```

- **Validate the features you added e.g. as for Reiser4, `dd if=/dev/zero of=fakedisk.sparse bs=1M count=0 seek=1000` will emulate a one gigabyte disk as thin provisioning, similarly to what the QCOW2 format allows.**

```
banana@banana-VirtualBox:~/Downloads$ dd if=/dev/zero of=fakedisk.sparse
e bs=1M count=0 seek=1000
0+0 records in
0+0 records out
0 bytes copied, 0,000288664 s, 0,0 kB/s
```

• **Bonus x2: –or– switch to another boot-loader, and according to your booting mode, BIOS/CSM vs EFI. Which one did you choose? What problems are you facing? Your system does not boot anymore? You messed up? Show the errors and how far you went**

Firstly I did something wrong and OS did not boot, however I created a new VM and managed to remove grub and instal lilo.

```
$sudo apt-get remove --purge --autoremove grub\*
```

```
$sudo apt-get install lilo
```

```
$sudo liloconfig
```

```
$sudo lilo
```

