Código fuente del kernel Linux

El código de Linux se mantiene en un gran repositorio: https://git.kernel.org/

Más bien es un servidor **git** que mantiene muchos repositorios relacionados al desarrollo de Linux. El repositorio donde se encuentra todo el código oficial/main de Linux, es el de Linus Torvalds: https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/ (Linux kernel source tree)

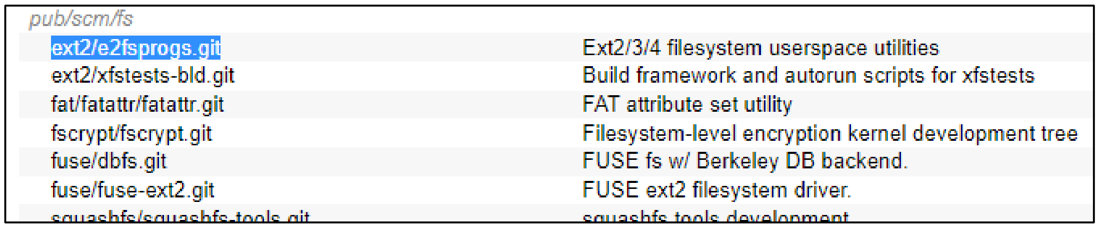
Las contribuciones que se hagan sobre ese repo, provienen de los distintos forks/clones que se hagan de él en *pub/scm/linux*: https://git.kernel.org/pub/scm/linux/kernel/git/. Cuando Linus y su equipo lo aprueben, se hace un '*merge*' en el repo de Linux, para incluir los nuevos cambios.

Por ejemplo, podemos encontrar los clones donde se desarrollan los drivers de los distintos **fs**:

- *Repo del driver de ext4*:
  https://git.kernel.org/pub/scm/linux/kernel/git/tytso/ext4.git/commit/
- *Repo del driver de ntfs (igual parece que ya no se usa, vienen de otros lados los commit)*:
  https://git.kernel.org/pub/scm/linux/kernel/git/aia21/ntfs.git/commit/

Podemos descargar todo el código '*main*' de Linux desde acá: https://www.kernel.org/

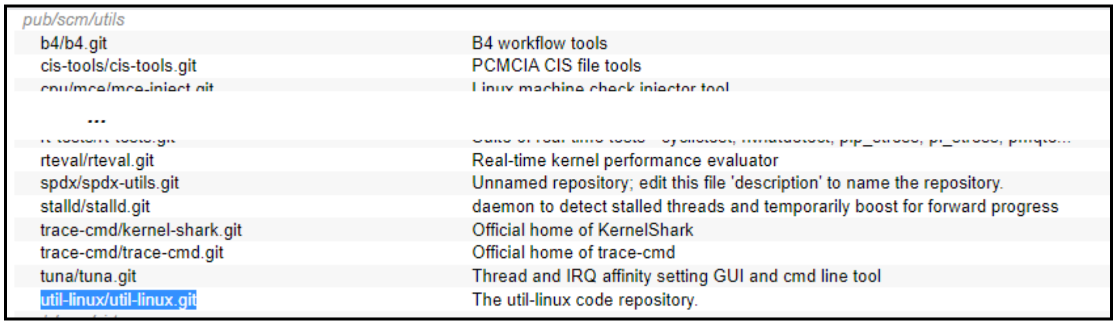
También acá podemos encontrar los repos del desarrollo de los paquetes '*e2fsprogs*' y '*util-linux*':



pub/scm/fs

ext2/e2fsprogs.git	Ext2/3/4 filesystem userspace utilities
ext2/xfstests-bld.git	Build framework and autorun scripts for xfstests
fat/fatattr/fatattr.git	FAT attribute set utility
fscrypt/fscrypt.git	Filesystem-level encryption kernel development tree
fuse/dbfs.git	FUSE fs w/ Berkeley DB backend.
fuse/fuse-ext2.git	FUSE ext2 filesystem driver.
squashfs/squashfs-tools.git	squashfs tools development

(*oficial*) https://git.kernel.org/pub/scm/fs/ext2/e2fsprogs.git/tree/
(*github*) https://github.com/tytso/e2fsprogs



pub/scm/utils

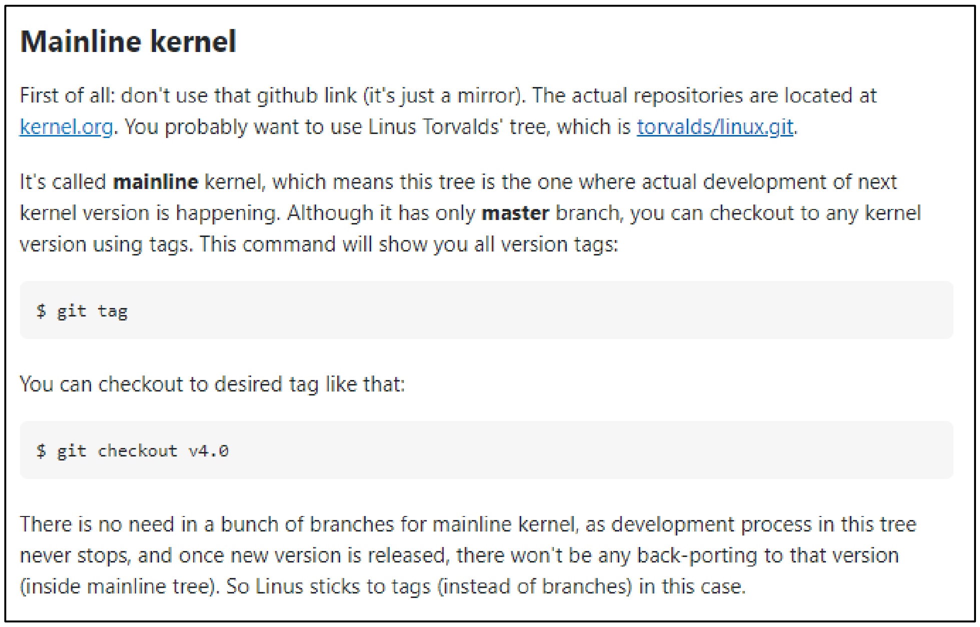
b4/b4.git	B4 workflow tools
cis-tools/cis-tools.git	PCMCIA CIS file tools
cpu/mce/mce-inject.git	Linux machine check injector tool
...	
rt-tests/rt-tests.git	Suite of real-time tests - cyclictest, hwlatdetect, pip_stress, pi_stress, pmqte...
rteval/rteval.git	Real-time kernel performance evaluator
spdx/spdx-utils.git	Unnamed repository; edit this file 'description' to name the repository.
stalld/stalld.git	daemon to detect stalled threads and temporarily boost for forward progress
trace-cmd/kernel-shark.git	Official home of KernelShark
trace-cmd/trace-cmd.git	Official home of trace-cmd
tuna/tuna.git	Thread and IRQ affinity setting GUI and cmd line tool
util-linux/util-linux.git	The util-linux code repository.

(*oficial*) https://git.kernel.org/pub/scm/utils/util-linux/util-linux.git/tree/
(*github*) https://github.com/karelzak/util-linux

O sea, en este gran servidor git hay varios conjuntos de repositorios, por un lado el del kernel y sus clones para hacer contribuciones, y por otro cosas extras como en este caso utilidades para el filesystem y utilidades generales.

Nota: *Linus también mantiene el repo 'main' de Linux en github: https://github.com/torvalds/linux Lo suele ir sincronizando, pero el repo oficial oficial está en kernel.org, el que puse antes.*

Info: https://stackoverflow.com/questions/30268332/why-does-the-linux-kernel-repository-have-only-one-branch



Mainline kernel

First of all: don't use that github link (it's just a mirror). The actual repositories are located at kernel.org. You probably want to use Linus Torvalds' tree, which is torvalds/linux.git.

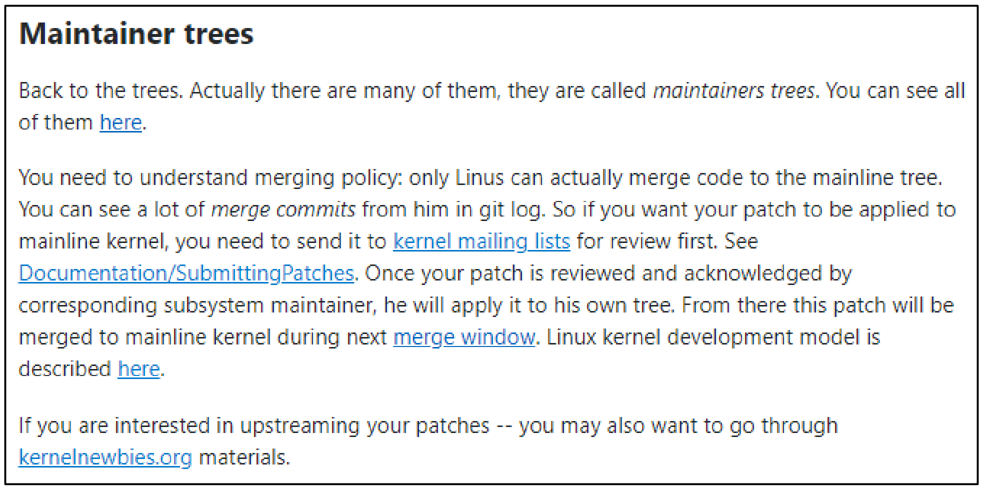
It's called **mainline** kernel, which means this tree is the one where actual development of next kernel version is happening. Although it has only **master** branch, you can checkout to any kernel version using tags. This command will show you all version tags:

```
$ git tag
```

You can checkout to desired tag like that:

```
$ git checkout v4.0
```

There is no need in a bunch of branches for mainline kernel, as development process in this tree never stops, and once new version is released, there won't be any back-porting to that version (inside mainline tree). So Linus sticks to tags (instead of branches) in this case.

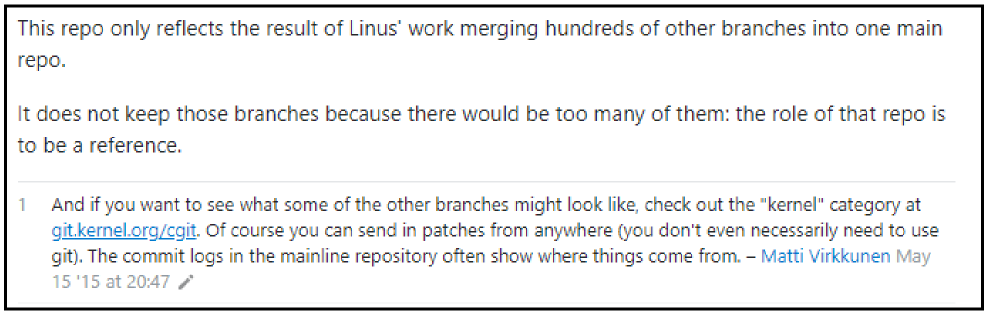


Maintainer trees

Back to the trees. Actually there are many of them, they are called *maintainers trees*. You can see all of them here.

You need to understand merging policy: only Linus can actually merge code to the mainline tree. You can see a lot of *merge commits* from him in git log. So if you want your patch to be applied to mainline kernel, you need to send it to kernel mailing lists for review first. See Documentation/SubmittingPatches. Once your patch is reviewed and acknowledged by corresponding subsystem maintainer, he will apply it to his own tree. From there this patch will be merged to mainline kernel during next merge window. Linux kernel development model is described here.

If you are interested in upstreaming your patches -- you may also want to go through kernelnewbies.org materials.

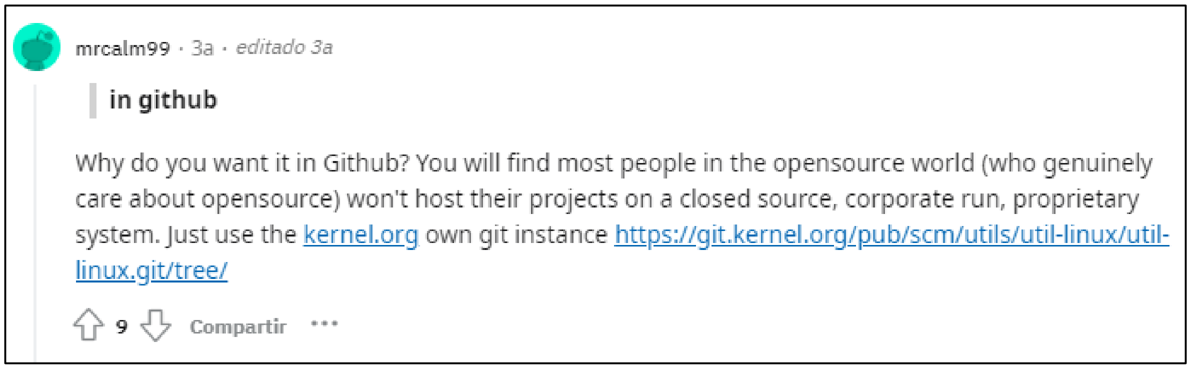


This repo only reflects the result of Linus' work merging hundreds of other branches into one main repo.

It does not keep those branches because there would be too many of them: the role of that repo is to be a reference.

1 And if you want to see what some of the other branches might look like, check out the "kernel" category at git.kernel.org/cgit. Of course you can send in patches from anywhere (you don't even necessarily need to use git). The commit logs in the mainline repository often show where things come from. – Matti Virkkunen May 15 '15 at 20:47

Comentario interesante jeje:
https://www.reddit.com/r/linux/comments/9qcf9o/how_do_i_find_the_source_code_of_linux_for_the/



mrcalm99 · 3a · *editado 3a*

> in github

Why do you want it in Github? You will find most people in the opensource world (who genuinely care about opensource) won't host their projects on a closed source, corporate run, proprietary system. Just use the kernel.org own git instance https://git.kernel.org/pub/scm/utils/util-linux/util-linux.git/tree/

Links extras:
https://www.quora.com/Is-Linus-Torvalds-about-to-move-the-Linux-Kernel-from-Github-to-Gitlab
https://ext4.wiki.kernel.org/index.php/Ext4_patchsets