Come contribuire al kernel Linux

Rigel Di Scala
Linux Day ~~2022~~ 2024

unixMiB

Chi siamo

Rigel

Currently: Head of Development, VEDRAI

Previously: Engineering Manager, Red Hat

Primo computer: Commodore Vic 20 (1984)

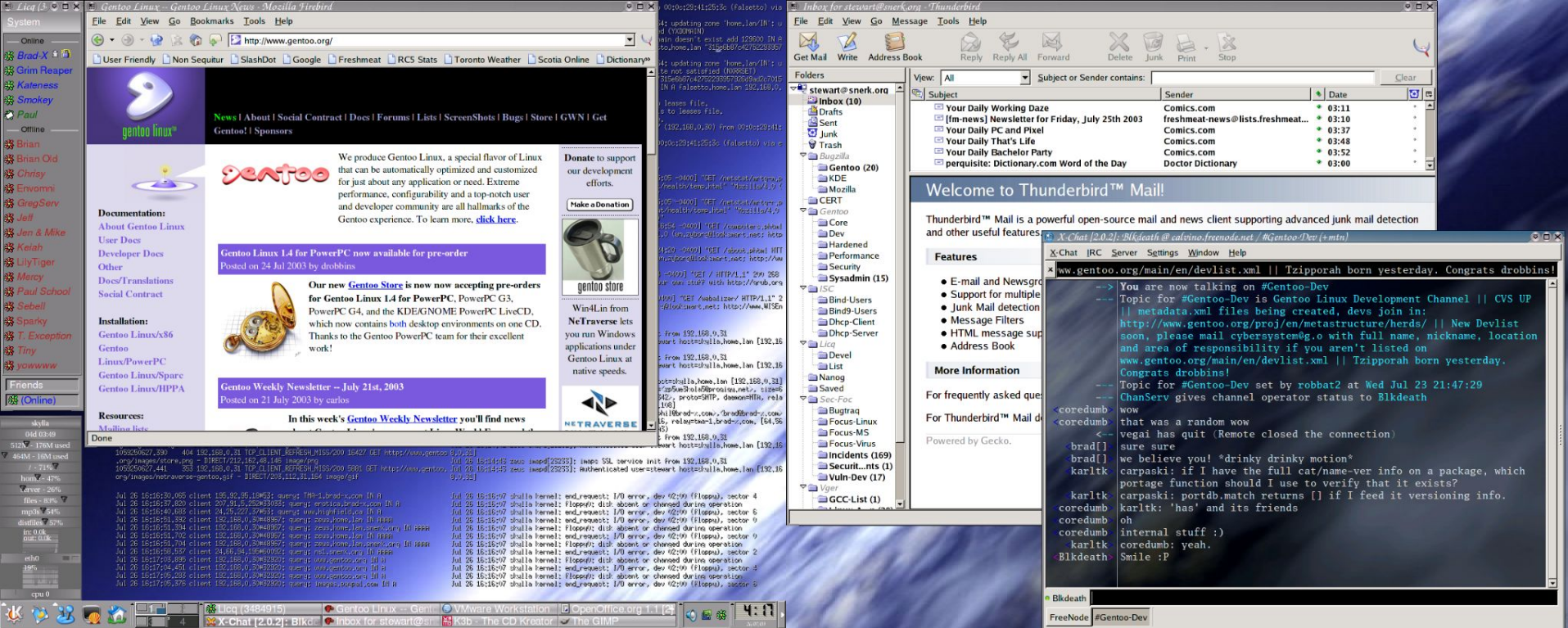Prima distro: Mandrake 7.2 (2001 - 2003)

github.com/zedr

www.vedrai.com
(psst… we're hiring!)

Typical Gentoo/KDE desktop – 2003/2004 (2.4 → 2.6)

gentoo

2004.2

```
raid6: sse2x4   19019 MB/s
raid6: using algorithm sse2x4 (19019 MB/s)
md: raid6 personality registered as nr 8
md: multipath personality registered as nr 7
md: md driver 0.90.0 MAX_MD_DEVS=256, MD_SB_DISKS=27
device-mapper: 4.1.0-ioctl (2003-12-10) initialised: dm@uk.sistina.com
Advanced Linux Sound Architecture Driver Version 1.0.4 (Mon May 17 14:31:44 2004 UTC).
ALSA device list:
  No soundcards found.
NET: Registered protocol family 2
IP: routing cache hash table of 4096 buckets, 32Kbytes
TCP: Hash tables configured (established 32768 bind 32768)
NET: Registered protocol family 1
NET: Registered protocol family 10
IPv6 over IPv4 tunneling driver
NET: Registered protocol family 17
md: Autodetecting RAID arrays.
md: autorun ...
md: ... autorun DONE.
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 184k freed
EXT2-fs warning: checktime reached, running e2fsck is recommended
>> Loading modules...
    :: Scanning for ehci-hcd...ehci-hcd loaded.
    :: Scanning for uhci... module not found.
    :: Scanning for usb-ohci... module not found.
    :: Scanning for hid... module not found.
    :: Scanning for usb-storage...usb-storage loaded.
    :: Scanning for ieee1394...ieee1394 loaded.
    :: Scanning for ohci1394...ohci1394 loaded.
    :: Scanning for sbp2...sbp2 loaded.
>> Mounting filesystems...
Started device management daemon v1.3.25 for /newroot/dev
>> Attempting to mount CD:- /newroot/dev/cdroms/*
>> Attempting to mount CD:- /newroot/dev/ide/cd/*


BusyBox v1.00-pre7 (2004.07.18-21:56+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

/bin/ash: can't access tty; job control turned off
/ # _
```

gentoo

2004.2

Unix beards

Default: Manual Configuration

Introduction

Manually configuring a kernel is often seen as the most difficult course every Linux users ever has to go through. Nothing is less true -- after configuring a couple of kernels you don't even remember that it was difficult ;)

However, one thing *is* true: you must know your system when you start configuring a kernel manually. Most information can be gathered by viewing the contents of `/proc/pci` (or by using `lspci` if available). You can also run `lsmod` to see what kernel modules the LiveCD uses (it might provide you with a nice hint on what to enable).

Now go to your kernel source directory and execute `make menuconfig`. This will fire up an ncurses-based configuration menu.

Code Listing 5: Invoking menuconfig

```
# cd /usr/src/linux
# make menuconfig
```

You will be greeted with several configuration sections. We'll first list some options you must activate (otherwise Gentoo will not function, or not function properly without additional tweaks).

Activating Required Options

First of all, activate the use of development and experimental code/drivers. You need this, otherwise some very important code/drivers won't show up:

Code Listing 6: Selecting experimental code/drivers

```
Code maturity level options --->
  [*] Prompt for development and/or incomplete code/drivers
```

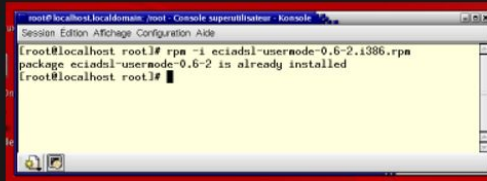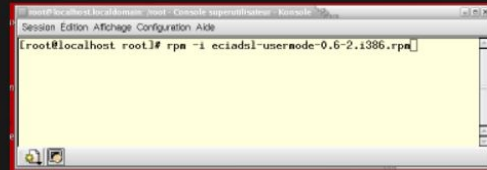Make sure that you compile your kernel with the correct processor family:

Code Listing 7: Selecting correct processor family

Gentoo Handbook (2004)

…otherwise on a root console (like un*x users):

```
# rpm -i eciadsl-usermode-0.10-1.i386.rpm
```


Disable or delete dabusb module: if /etc/hotplug/blacklist exists, edit it and add a line containing the word 'dabusb' (without the quotes) to it. Restart Linux.

Otherwise, type:

```
# modprobe -r dabusb && rm -f $(modprobe -l | grep dabusb) && depmod -a
```

DistroWatch.com

English • Español • Français • Deuts

- Home Page, Headlines
- DW Weekly, Comments
- Packages, Package Management
- Glossary, FAQ, Mobile Site

- Searc
- Major
- Subm
- Upco

Dist

Type Distribution Name | Go

Select Distribution | Go

DuckDuckGo Site Search | Go

Random Distribution

News/Opinions/Reviews ▼   Packages ▼   Find/Submit Distro ▼   Tutorials & Learning ▼

Related Resources ▼   DistroWatch.com ▼

3CX

📶 Latest News and Updates

3CX

News Filtering Options

Distribution: All   **Release:** All   **Month:**
All   **Year:** All   Refresh

2024-10-23   NEW • **Distribution Release: Parrot 6.2**

Parrot (formerly Parrot Security OS) is a Debian-based, security-oriented distribution featuring a collection of utilities designed for penetration testing, computer forensics, reverse engineering,
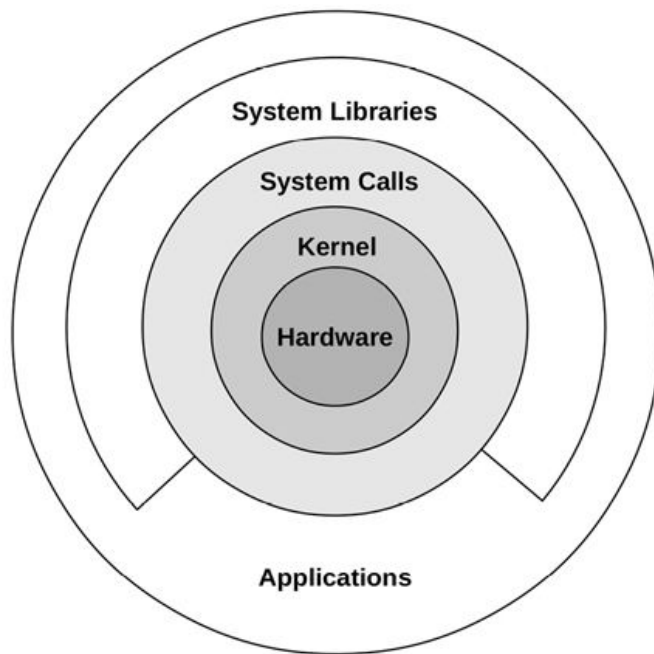
Introduzione

Obiettivi

1. Capire dati e trend interessanti del kernel
2. Capire come scaricare e compilare il kernel
3. Capire come creare, verificare ed inviare una patch
4. ~~Capire come scrivere un piccolo modulo in Rust~~

Il kernel Linux

Le responsabilità del kernel

The role of a monolithic operating system kernel
Brendan Gregg, Kernel Internals in Operating Systems

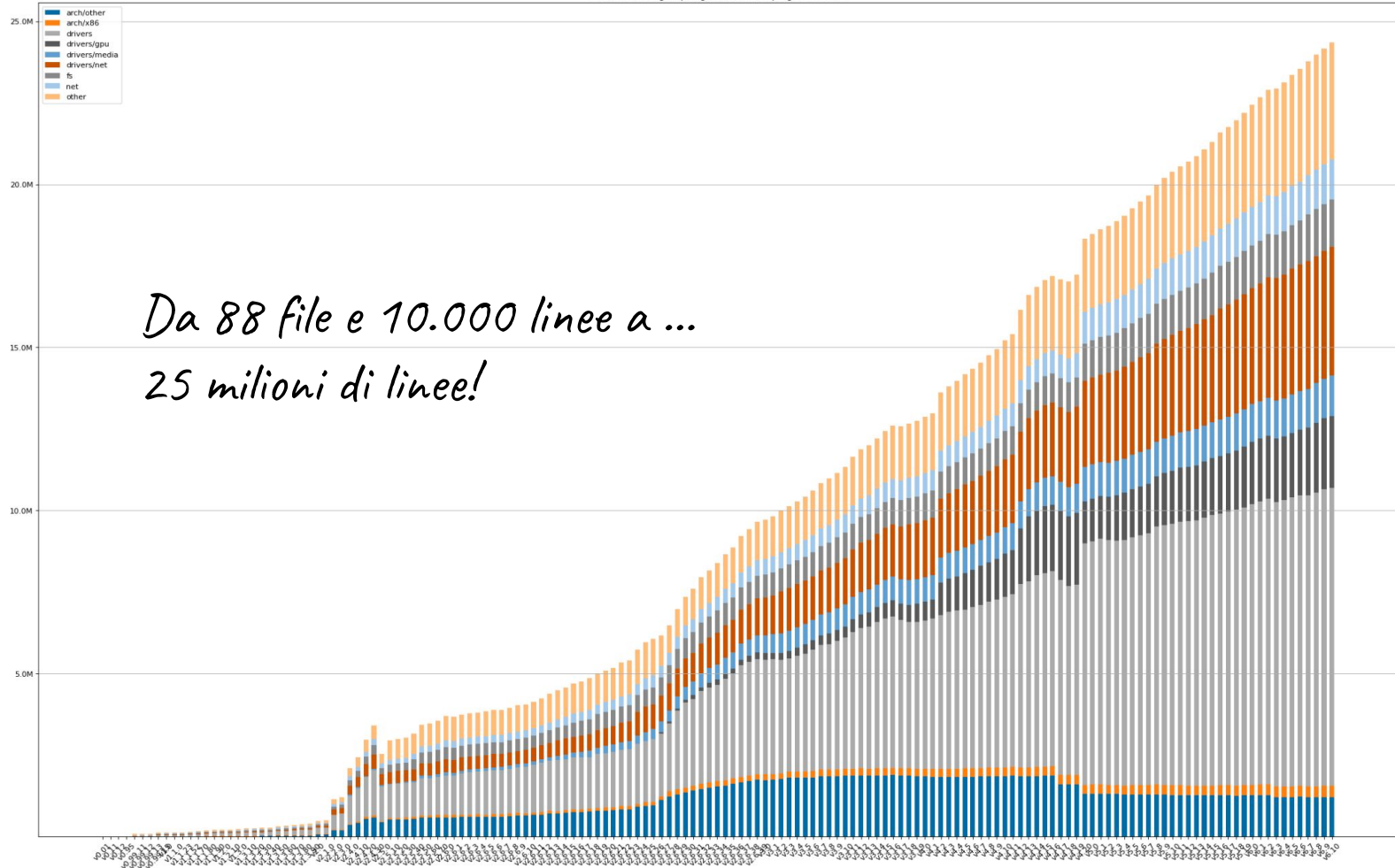"The Linux kernel is the largest collaborative software project ever."

```
Statistics:

It's 88 files with about ten thousand lines, written by
yours truly except for the vsprintf routine which was
co-written with Lars Wirzenius.

(...)

Hardware needed for running linux:
    - 386 AT
    - VGA/EGA screen
    - AT-type harddisk controller (IDE is fine)
    - Finnish keyboard (oh, you can use a US keyboard, but
      not without some practise :-)
```
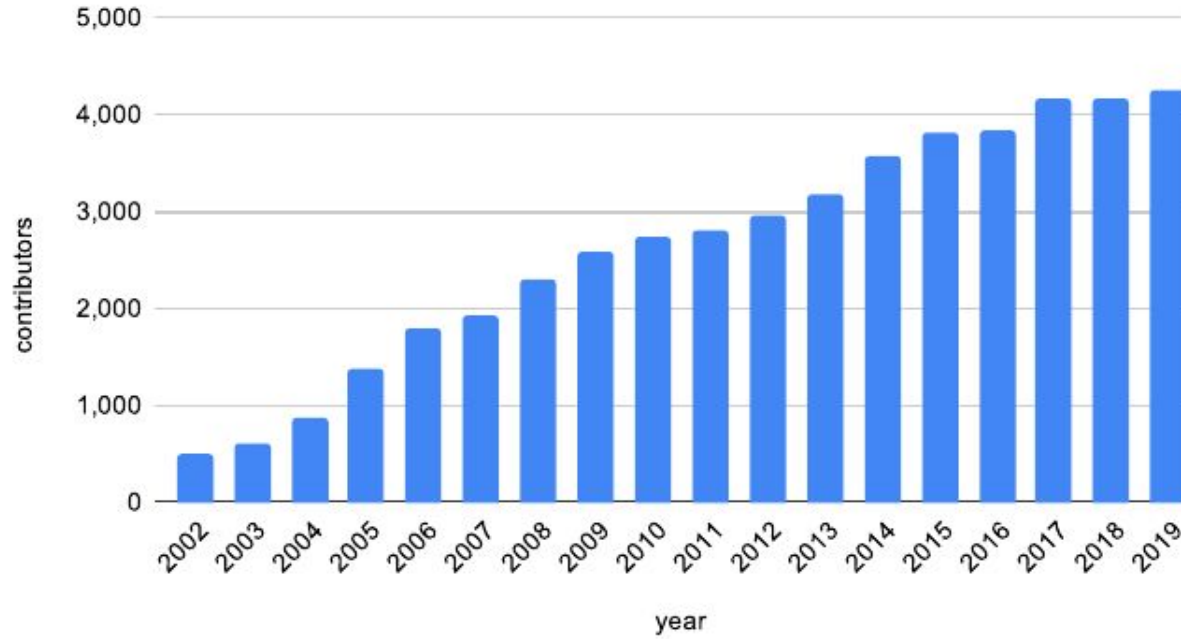
Linus Torvalds, 17 novembre 1991 – Linux 0.01.txt

Lines of code in the Linux kernel
Generated using https://github.com/udoprog/kernelstats

Da 88 file e 10.000 linee a …
25 milioni di linee!

TIOBE Programming Community Index

Rating basato sul tipo e numero di ricerche sul web

Contributors to the Linux Kernel per Year
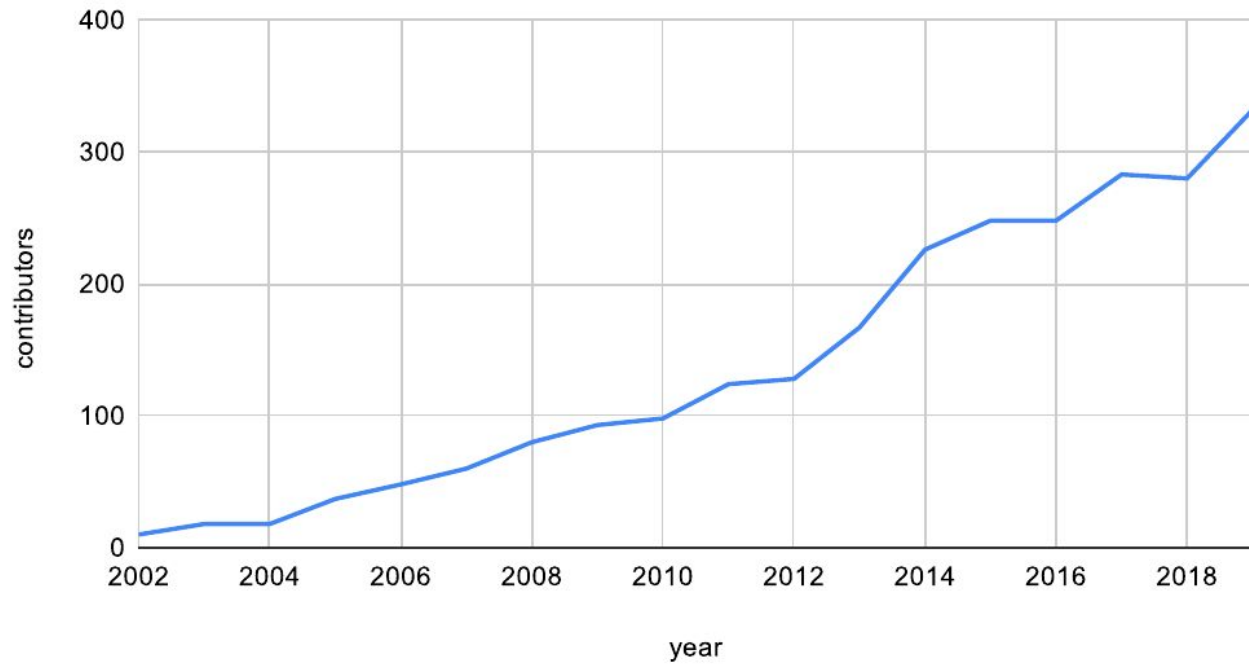
Fonte: Linux Foundation (2020)

Being maintainer feels like a punishment, and that cannot stand. We need help.
— Darrick Wong

Maintainers/longtime developers are burning out.
— Josef Bacik

Estimated Female Contributors to the Linux Kernel per Year

Fonte: Linux Foundation (2020)

Most active employers, 5.16 through 6.1

By changesets			By lines changed		
Intel	9295	10.7%	AMD	1653237	26.8%
(Unknown)	6134	7.1%	Intel	744059	12.1%
Google	5597	6.5%	Google	345341	5.6%
Red Hat	4916	5.7%	Linaro	302581	4.9%
AMD	4474	5.2%	(Unknown)	212408	3.4%
Linaro	4373	5.0%	Meta	209040	3.4%
(None)	4029	4.6%	Red Hat	202698	3.3%
Huawei Technologies	3649	4.2%	Realtek	185932	3.0%
Meta	2904	3.4%	NVIDIA	179426	2.9%
NVIDIA	2563	3.0%	(None)	165999	2.7%

Fonte: Jonathan Corbet, LWN.net (2022)

Occurrences of words in the Linux kernel source code over time

Legend:
- hack*
- workaround*
- todo
- fixme
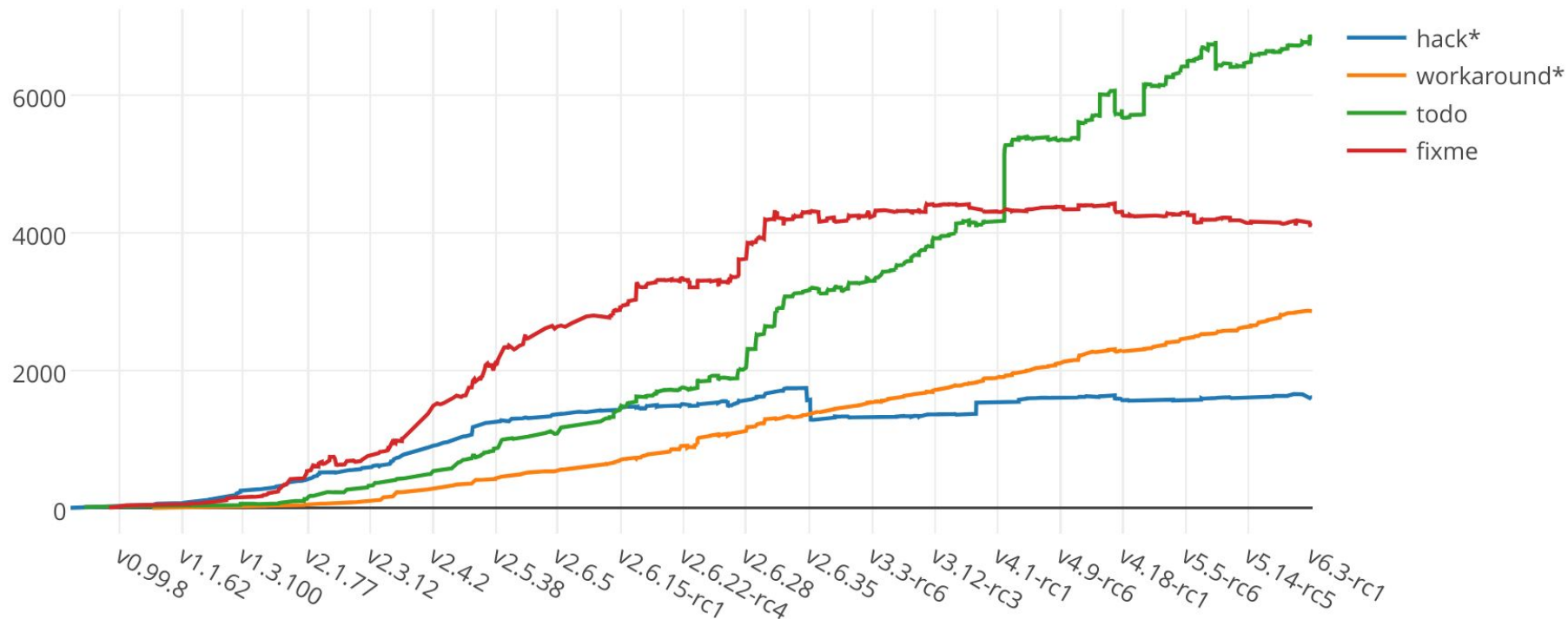
Occurrences of words in the Linux kernel source code over time
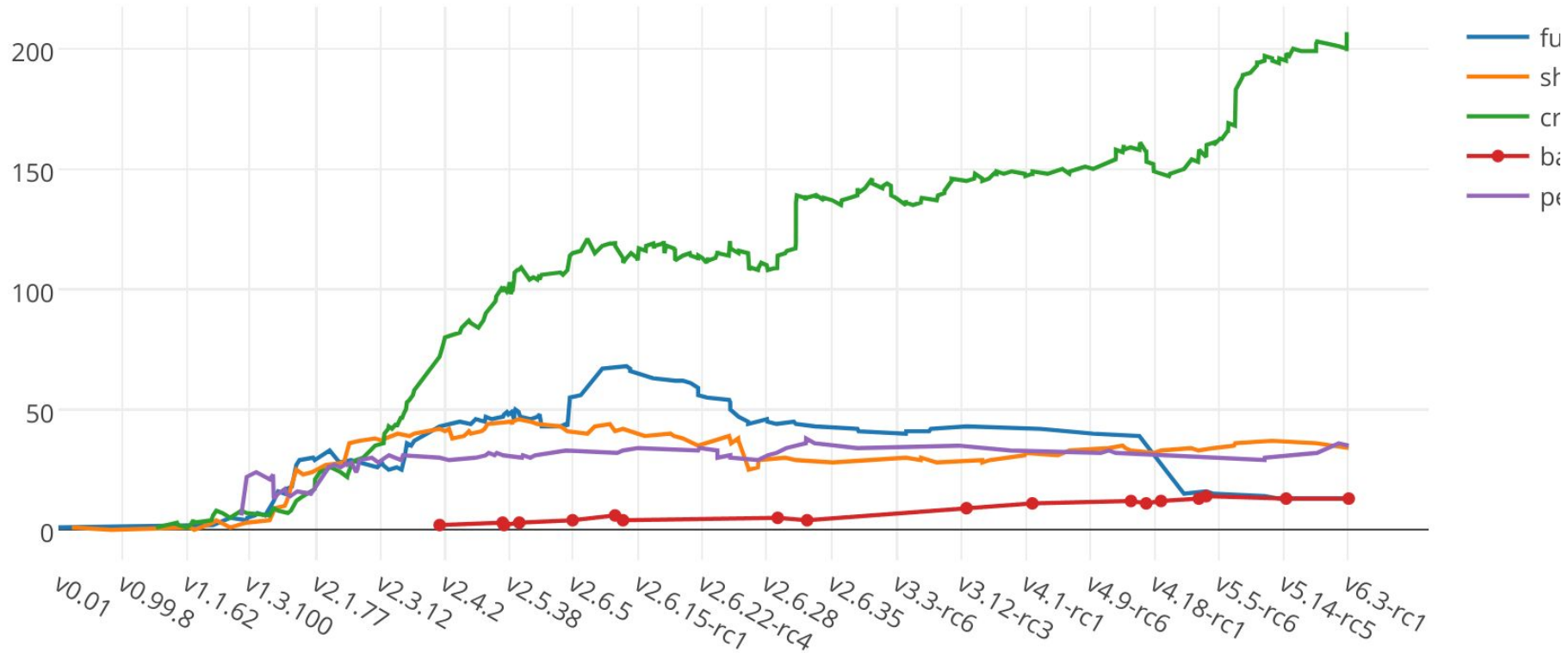
Come compilare il kernel

Greg Kroah-Hartman
Linux Kernel Developer and
Maintainer

Pre-requisiti (minimi)

GNU C	util-linux	PPP
Clang/LLVM (optional)	kmod	nfs-utils
Rust (optional)	e2fsprogs	procps
bindgen (optional)	jfsutils	udev
GNU make	reiserfsprogs	grub
bash	xfsprogs	mcelog
binutils	squashfs-tools	iptables
flex	btrfs-progs	openssl & libcrypto
bison	pcmciautils	bc
pahole	quota-tools	cpio

Fonte: Documentation/process/changes.rst (4.18)

Pre-requisiti (per la demo)
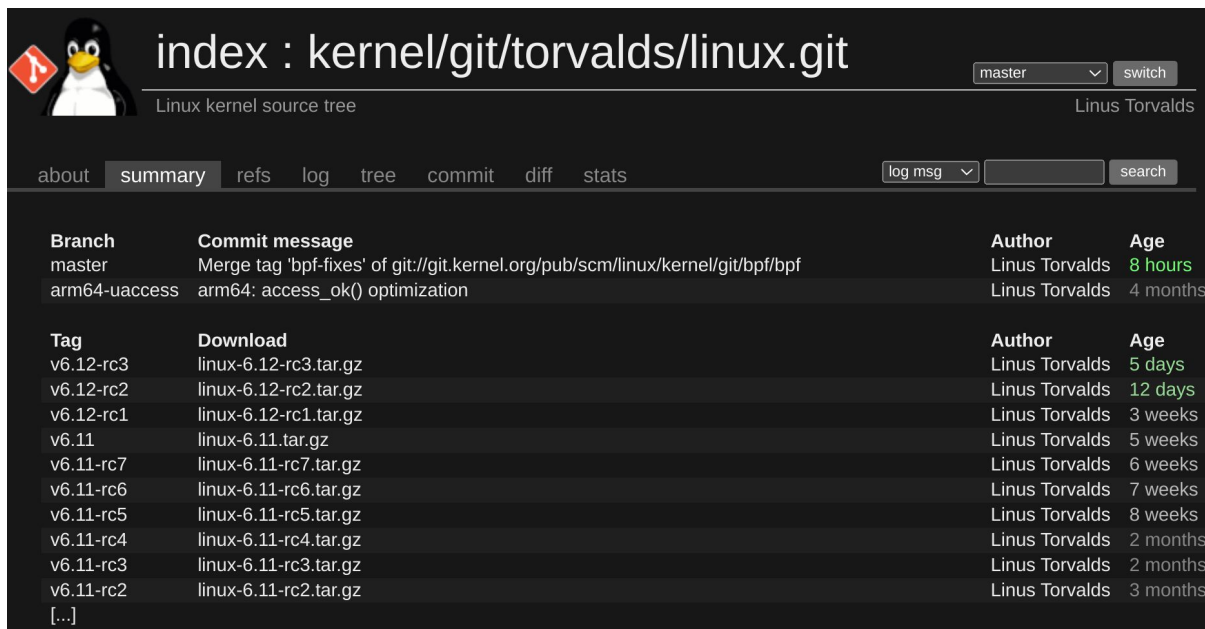
- Git
- Make
- gcc

Su Fedora:

```
$ sudo dnf install -y \

    git gcc make bc elfutils-libelf-devel \

    openssl-devel flex bison ncurses-devel zstd
```

Oppure, semplicemente:

```
$ sudo dnf group install -y "Development Tools" zstd
```

Dove scaricare i sorgenti del kernel

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

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

Partire da zero

↓ Pulisci i file di build (e.g. .ko) ma non la configurazione (.config)

```
$ make clean
```

↓ Pulisci i file di configurazione (incluso .config, occhio!)

```
$ make mrproper
```

↓ Boom 💥

```
$ make distclean
```

Creare una configurazione

```
$ ARCH=x86_64 make defconfig
```

oppure

```
$ cp /boot/config-... .config && yes "" | make oldconfig
```

infine

```
$ make menuconfig
```

Dubbi?

```
$ make help
```

Linux/x86 6.12.0-rc3 Kernel Configuration

Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters
are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?>
for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

```
        General setup  --->
    [*] 64-bit kernel
        Processor type and features  --->
    [*] Mitigations for CPU vulnerabilities  --->
        Power management and ACPI options  --->
        Bus options (PCI etc.)  --->
        Binary Emulations  --->
    [*] Virtualization  --->
        General architecture-dependent options  --->
    [*] Enable loadable module support  --->
    [*] Enable the block layer  --->
        Executable file formats  --->
        Memory Management options  --->
    [*] Networking support  --->
        Device Drivers  --->
        File systems  --->
        Security options  --->
    -*- Cryptographic API  --->
    v(+)
```

    <Select>   < Exit >   < Help >   < Save >   < Load >

Compilare il kernel

Compilare ed installare il kernel

```
$ sed -i "s/EXTRAVERSION = /EXTRAVERSION = <MYTAG>/"


$ make bzImage

$ sudo make modules_install

$ sudo make install
```

Intervallo

Come aderire alle linee guida per C

Regole base

1. Indentazione con tab a 8 caratteri
2. Ogni linea ha un massimo di 80 caratteri
3. I blocchi e le parentesi graffe seguono lo stile K&R
4. Usare in modo giudizioso gli spazi ed i commenti
5. I nomi sono espressivi e usano solo lettere minuscole (eccetto le costanti, macro e gli enum)
6. Non_t usare_t typedef per_t struct_t e puntatori_t
7. Funzioni brevi che fanno (bene) una cosa sola
8. Usare goto solo in alcuni casi
9. Ritornare 0 o 1? vedi predicati vs. comandi
10. Usare le macro in include/linux/kernel.h

Ed in particolare…

if (cond) {

    free(ptr);

} else {

    do_something_else();

}

⟶

if (cond)

    free(ptr);

else

    do_something_else();

free(ptr);

"Do not unnecessarily use braces where a single statement will do."
https://www.kernel.org/doc/html/v4.10/process/coding-style.html#placing-braces-and-spaces

Come trovare qualcosa da fare

Visita Kernel Janitors

https://kernelnewbies.org/KernelJanitors/Todo

* `printk()` calls should include appropriate `KERN_*` constant (of course

    only at beginning of lines). Also consider using dev_printk() and friends.

* `pr_debug()` from `kernel.h` could replace a lot of `DPRINTK` and similar macros.

* same for `pr_info()`

* Lots of unnecessary casts (mostly void* pointers) in drivers can be removed.

    Example:

```
- struct netdev_private *np = (struct netdev_private *)dev->priv;
+ struct netdev_private *np = dev->priv;
```

    use `netdev_priv()` for network devices
    See /VoidPointerConvs for more details.

* `ARRAY_SIZE` macro has duplicates, remove them.

    (2007/10/27 - the 4 remaining duplicate defs have good reasons for existing. Someone please verify agreement and then remove this todo item)

* put BIT macro into kernel.h (?) and remove it from drivers

* Purely cosmetic, but far nicer to read.

```
- for (list = ymf_devs.next; list != &ymf_devs; list = list->next) {
+ list_for_each(list, &ymf_devs) {
```

* Convert comments to C99 initializers:

```
-        /* 10 */ DECLARE_PIIX_DEV("ICH2"),
+        [10] = DECLARE_PIIX_DEV("ICH2"),
```

Guarda i TODO

Specialmente in drivers/staging

Trova piccole cose da sistemare

Prova con checkpatch:

```
$ pip install ply gitpython

$ ./scripts/checkpatch.pl -f drivers/staging/**/*.c
```

Come creare una patch

1. Crea un branch

$ git checkout -b test-linux-day

2. Prova a ricompilare il modulo

Abilitare il modulo in `make menuconfig`

Ricompila il kernel (per le nuove dipendenze)

```
$ make prepare && make
```

In seguito...

```
$ make modules
```

Oppure anche...

```
$ make M=drivers/staging/...
```

> *"prepare: things we need to do before we recursively start building the kernel"*

3. Crea il commit con un messaggio idoneo
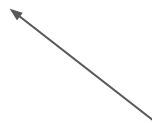
$ git commit -m "Staging: <nome_modulo>: <cosa fa>"

anzi...

$ git commit --amend

aggiungere:

Signed-off-by: Autore <autore@example.com>

Developer's Certificate of Origin

4. Creare la patch

$ git format-patch master..test-linux-day

Every program attempts to expand until it can read mail.

Those programs which cannot so expand are replaced by ones which can.

Zawinski's Law

5. Installare e configurare il client email di Git

```
$ dnf install -y git-email

$ git config --global sendemail.smtppass l1n00xd4y

$ git config --global sendemail.smtpuser zedr@zedr.com

$ git config --global sendemail.smtpencryption tls

$ git config --global sendemail.smtpserver mail.example.com

$ git config --global sendemail.serverport 465
```

6. Identificare i maintainer

```
$ ./scripts/get_maintainer.pl 0001-mia.patch
```

```
Greg Kroah-Hartman <gregkh@linuxfoundation.org>
(supporter:STAGING SUBSYSTEM,commit_signer:4/5=80%)

linux-staging@lists.linux.dev (open list:STAGING SUBSYSTEM)

linux-kernel@vger.kernel.org (open list)
```

7. Inviare la patch

```
$ git send-email \

--smtp-debug \

--to gregkh@linuxfoundation.org \

--cc linux-staging@lists.linux.dev \

--cc linux-kernel@vger.kernel.org \

0001-mia.patch
```

Il risultato…

From: **Greg KH** <gregkh@linuxfoundation.org>
Date: Thu, 13 Oct 2022, 07:56
Subject: Re: [PATCH] Staging: rtl8192e: fix a brace style issue
To: Rigel Di Scala <zedr@zedr.com>

On Wed, Oct 12, 2022 at 09:16:12PM +0000, Rigel Di Scala wrote:
> Fixed a coding style issue affecting a conditional if statement.
>
> Signed-off-by: Rigel Di Scala <zedr@zedr.com>
> ---
> drivers/staging/rtl8192e/rtl819x_BAProc.c | 5 ++---
> 1 file changed, 2 insertions(+), 3 deletions(-)
(...)

Hi,

This is the friendly semi-automated patch-bot of Greg Kroah-Hartman.
You have sent him a patch that has triggered this response.

Right now, the development tree you have sent a patch for is "closed"
due to the timing of the merge window. Don't worry, the patch(es) you
have sent are not lost, and will be looked at after the merge window is
over (after the -rc1 kernel is released by Linus).

So thank you for your patience and your patches will be reviewed at this
later time, you do not have to do anything further, this is just a short
note to let you know the patch status and so you don't worry they didn't
make it through.

thanks,

greg k-h's patch email bot

Prima email automatica da Greg KH

From: **Philipp Hortmann** <philipp.g.hortmann@gmail.com>
Date: Fri, 14 Oct 2022, 18:56
Subject: Re: [PATCH] Staging: rtl8192e: fix a brace style issue
To: Rigel Di Scala <zedr@zedr.com>, <gregkh@linuxfoundation.org>
Cc: <linux-staging@lists.linux.dev>

On 10/12/22 23:16, Rigel Di Scala wrote:
> Fixed a coding style issue affecting a conditional if statement.
>
> Signed-off-by: Rigel Di Scala <zedr@zedr.com>
> ---
>   drivers/staging/rtl8192e/
rtl819x_BAProc.c | 5 ++---
>   1 file changed, 2 insertions(+), 3 deletions(-)
(...)

Tested-by: Philipp Hortmann <philipp.g.hortmann@gmail.com>

Seconda email automatica da PH

From: **Greg KH** <gregkh@linuxfoundation.org>
Date: Thu, 20 Oct 2022, 17:35
Subject: patch "Staging: rtl8192e: fix a brace style issue" added to staging-testing
To: <zedr@zedr.com>, <gregkh@linuxfoundation.org>, <philipp.g.hortmann@gmail.com>

This is a note to let you know that I've just added the patch titled

        Staging: rtl8192e: fix a brace style issue

to my staging git tree which can be found at
        git://git.kernel.org/pub/scm/linux/kernel/git/gregkh/staging.git
in the staging-testing branch.

The patch will show up in the next release of the linux-next tree
(usually sometime within the next 24 hours during the week.)

The patch will be merged to the staging-next branch sometime soon,
after it passes testing, and the merge window is open.

If you have any questions about this process, please let me know.

From 06b764cf60877bdd6604c173963501e16236ccdf Mon Sep 17 00:00:00 2001
From: Rigel Di Scala <zedr@zedr.com>
Date: Wed, 12 Oct 2022 21:16:12 +0000
Subject: Staging: rtl8192e: fix a brace style issue

Fixed a coding style issue affecting a conditional if statement.

Signed-off-by: Rigel Di Scala <zedr@zedr.com>
Tested-by: Philipp Hortmann <philipp.g.hortmann@gmail.com>
Link: https://lore.kernel.org/r/20221012211612.75871-1-zedr@zedr.com
Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

Infine, altra email da Greg KH – patch accettata!

```
zedr@zaonce:~/src/linux                                    ×

$ git log --author="Rigel"
commit 06b764cf60877bdd6604c173963501e16236ccdf
Author: Rigel Di Scala <zedr@zedr.com>
Date:   Wed Oct 12 21:16:12 2022 +0000

    Staging: rtl8192e: fix a brace style issue

    Fixed a coding style issue affecting a conditional if statement.

    Signed-off-by: Rigel Di Scala <zedr@zedr.com>
    Tested-by: Philipp Hortmann <philipp.g.hortmann@gmail.com>
    Link: https://lore.kernel.org/r/20221012211612.75871-1-zedr@zedr.com
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
```

Conclusioni

- Linux è uno dei progetti software più prestigiosi mondo

- Linux ha un bisogno forte di collaboratrici e collaboratori

- La barriera d'ingresso per contribuire è molto bassa

- I contributi significativi vi distinguono da altri candidati

Make your first contribution today!