A quarter century of Linux open source

MY JOURNEY AND LESSONS

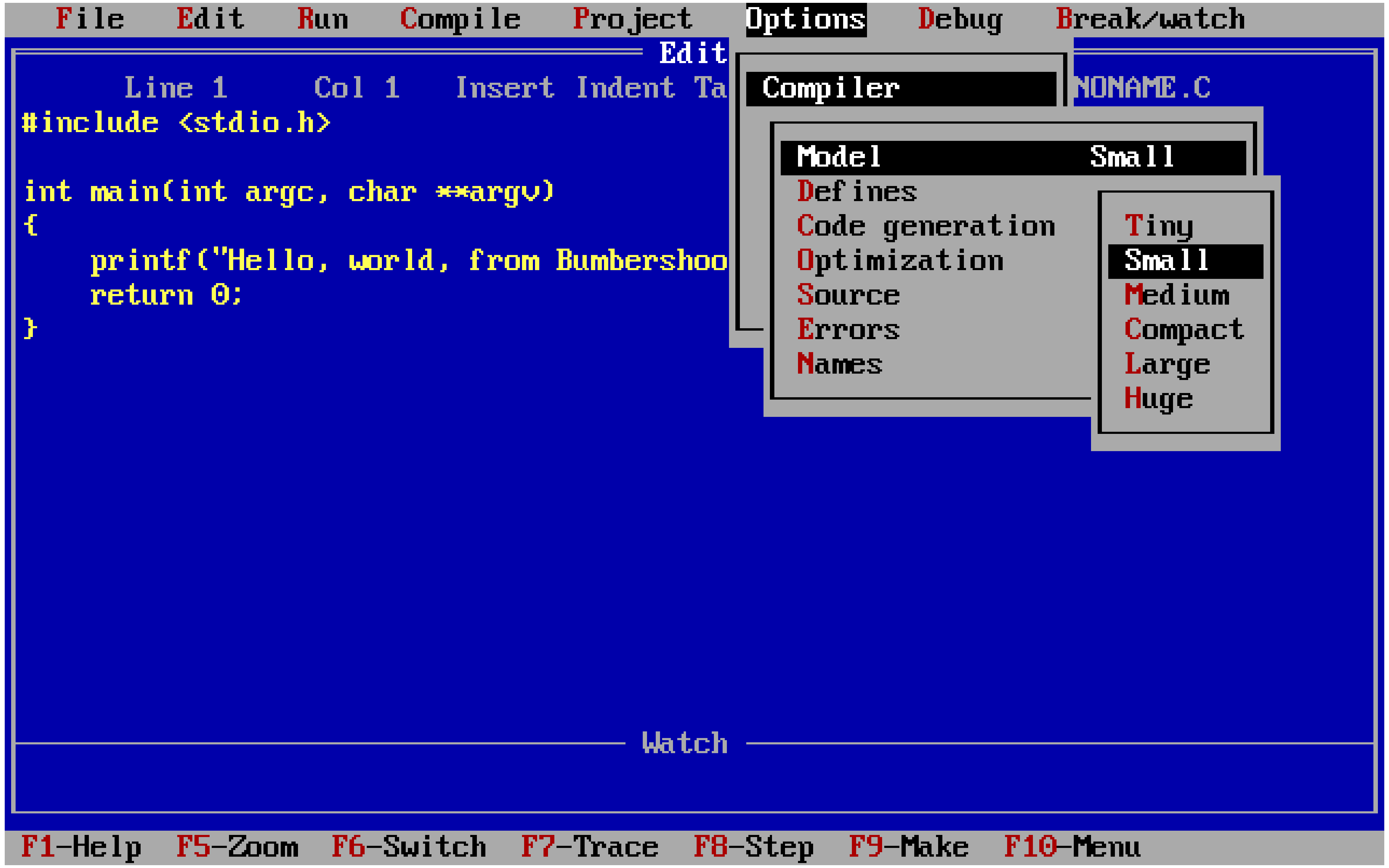
Jens Axboe
Software Engineer

Meta

Early days

Programming C/asm in DOS

Early days



Line 1   Col 1   Insert Indent Ta   NONAME.C

```
#include <stdio.h>

int main(int argc, char **argv)
{
    printf("Hello, world, from Bumbershoo
    return 0;
}
```

Model   Small

Early days

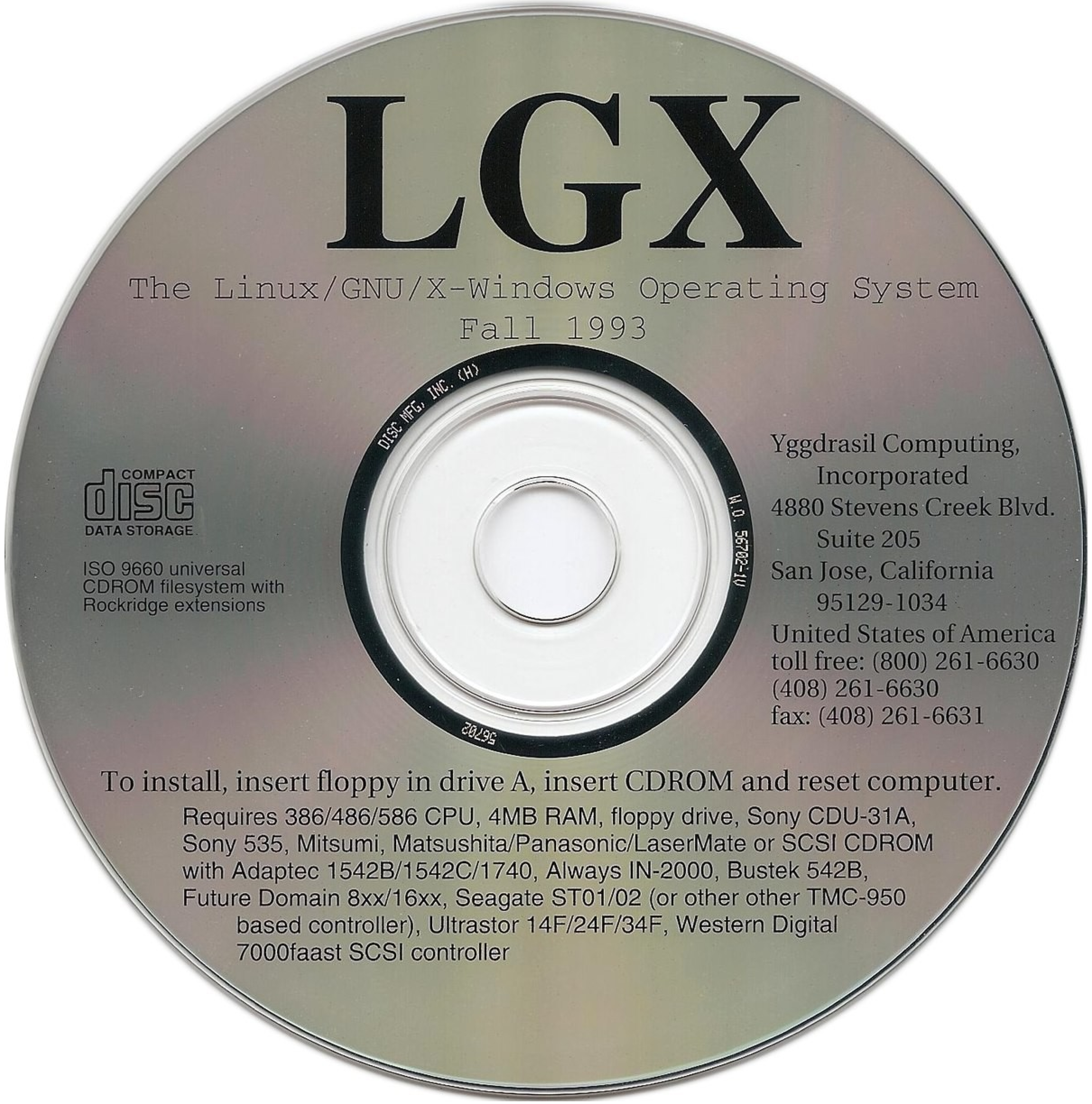
Programming C/asm in DOS

Read an article about Linux in my dads "engineering weekly"

Download slackware from a Swedish BBS

- Dad sad no
- Did it anyway

No internet, found Yggdrasil Fall 1993 CD



LGX

The Linux/GNU/X-Windows Operating System
Fall 1993

ISO 9660 universal
CDROM filesystem with
Rockridge extensions

Yggdrasil Computing,
Incorporated
4880 Stevens Creek Blvd.
Suite 205
San Jose, California
95129-1034
United States of America
toll free: (800) 261-6630
(408) 261-6630
fax: (408) 261-6631

To install, insert floppy in drive A, insert CDROM and reset computer.

Requires 386/486/586 CPU, 4MB RAM, floppy drive, Sony CDU-31A, Sony 535, Mitsumi, Matsushita/Panasonic/LaserMate or SCSI CDROM with Adaptec 1542B/1542C/1740, Always IN-2000, Bustek 542B, Future Domain 8xx/16xx, Seagate ST01/02 (or other other TMC-950 based controller), Ultrastor 14F/24F/34F, Western Digital 7000faast SCSI controller

Early days

Programming C/asm in DOS

Read an article about Linux in my dads "engineering weekly"

Download slackware from a Swedish BBS

- Dad sad no
- Did it anyway

No internet, found Yggdrasil Fall 1993 CD

Mind blown

- Flat memory model, waaat
- Source available
- People doing this kind of thing for free?!

Obviously the kernel was the coolest thing ever

Moving towards Linux

Study at university of copenhagen

- 2 years of moscow ML killed me

Student job

- Worked just enough to pay my cost of living
- All other time spent on Linux

Religiously follow LKML

- CDROM maintainer looking for new maintainer
- Sent him an email offering my services

Moving towards Linux

CDROM maintainer

- DVD support, better burning, CD-DA DMA support

September 9, 1999	**none**	**2.3.17**	**none**

- Merged in Richard Sharman new CD-ROM changer support. This unifies changer handling across ATAPI and SCSI, previously only ATAPI supported changers. This isn't done yet, still need to do a bit of rearranging.
- Looked into why it isn't possible to play VCD/CDi movies with the current code. I haven't quite figured it out yet - SCSI seems to work and so does ATAPI over ide-scsi, but not ATAPI alone. Give me a mail if you experience otherwise (both stock 2.3.17 and with my patch).
- Fixed a couple of memory leaks in cdrom.c.
- No 2.2 patch today and I'm not sure that I'll continue putting out 2.2 patches. I just spend too much time merging code back from 2.3. In the future I will provide a 2.2 patch that contains only the DVD stuff and assorted bug fixes. Send me a mail if you would like me to continue providing up-to-date patches for 2.2 as well.
- Ported to 2.3.17.

September 2, 1999	**2.2.12**	**2.3.16**	**none**

- David Miller reported that the Uniform layer set negative transfer lengths for some packets and this was confusing some SCSI drivers when doing DMA. It did so for historical reasons, since ide-cd has always viewed data transfers going to the drive as having negative buffer lengths. This was fixed up and ide-cd now behaves as the SCSI CD-ROM driver. It cleaned up the code a bit too, which is always a good thing.
- Changed the warning for applications not opening O_NONBLOCK when they should to be a little more informative (before it just told you that the program was buggy, not why).

Moving towards Linux

CDROM maintainer

- DVD support, better burning, CD-DA DMA support
- Taught me a lot about IDE
- Block layer in common, started dabbling in that

Block layer maintainer

- It worked, what was there to do?
- Highmem (back then, above ~1G)
  - Even capable devices bounced highmem for IO
- Scaling
  - Single spinlock protecting everything
- IO schedulers
  - CFQ, deadline, noop
- Kicked off the 2.5 kernel series with this rework

Block layer maintainer by default

Kernel maintainer

Sets direction for the project / subsystem

- Does this change make sense?
- Testing

Reviews changes

Rewrites commit messages...

Commits contributor changes

- This release?
- Stable?

Stable backports

Deals with upstream maintainer

Pull requests

Bug reports

Bug reports

What’s a bad bug report?

“I upgraded to version x.y.z and now my application doesn’t work anymore!”

Bug reports

What’s a bad bug report? One that includes zero details.

What should a good bug report include?

Bug reports

What’s a bad bug report? One that includes zero details.

What should a good bug report include?

1) Specifics on version upgraded **from** and **to**, if applicable.
2) Specifics on exactly **what** broke.
3) Other relevant details. Hardware used? Libraries? Distro?
4) Steps to reproduce! Bonus points for a reproducer, single most valuable thing you can provide.

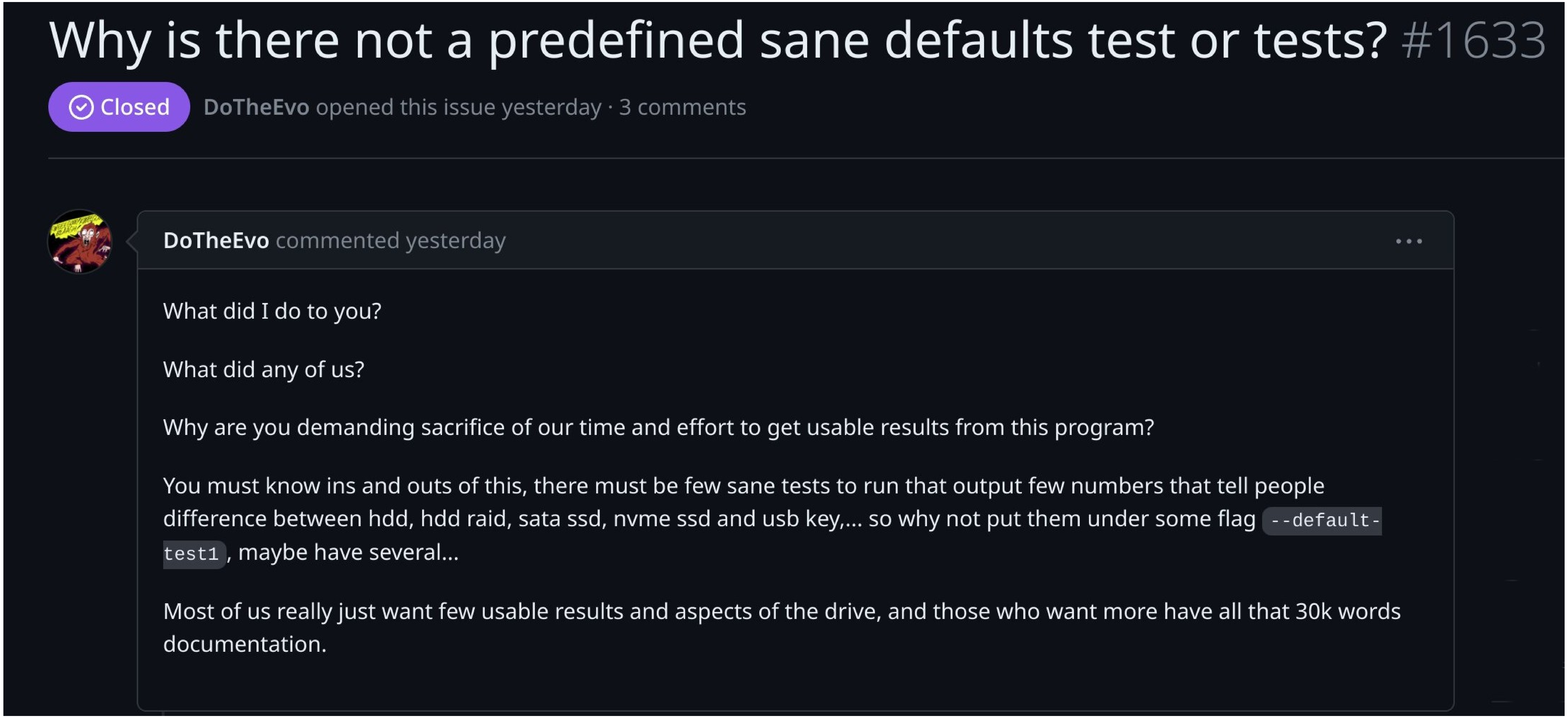
Bug reports

What’s a bad bug report? One that includes zero details.

What should a good bug report include?

1) Specifics on version upgraded **from** and **to**, if applicable.
2) Specifics on exactly **what** broke.
3) Other relevant details. Hardware used? Libraries? Distro?
4) Steps to reproduce! Bonus points for a reproducer, single most valuable thing you can provide.
5) Actually a reproducer isn’t the most valuable thing you can include – a fix is!

Make it as **easy** and enticing as possible for someone to help you.



Why is there not a predefined sane defaults test or tests? #1633

DoTheEvo opened this issue yesterday · 3 comments

DoTheEvo commented yesterday

What did I do to you?

What did any of us?

Why are you demanding sacrifice of our time and effort to get usable results from this program?

You must know ins and outs of this, there must be few sane tests to run that output few numbers that tell people difference between hdd, hdd raid, sata ssd, nvme ssd and usb key,... so why not put them under some flag `--default-test1`, maybe have several...

Most of us really just want few usable results and aspects of the drive, and those who want more have all that 30k words documentation.

Kernel rate of change

New release every ~2 months

- Merge window

Factor in time to write change and get it upstream

Stable and distros

Kernel bypass solutions

- Bypassing more than just the running kernel

Kernel simultaneously really fast and very slow

Maintainer fatigue

Always on

- All times zones, weekends

Security issues

Release cadence

Juggling two jobs

Dealing with people

Choice of email

Maintainer fatigue

Always on

- All times zones, weekends

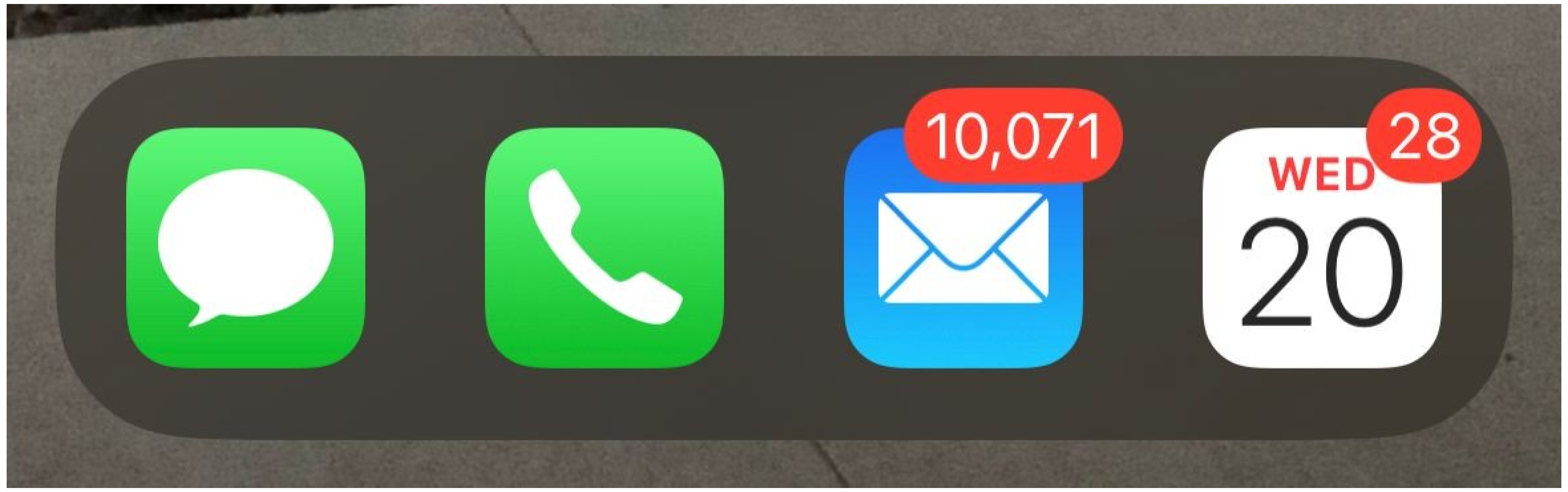
Security issues

Release cadence

Juggling two jobs

Dealing with people

Choice of email



Maintainer fatigue

Always on
- All times zones, weekends

Security issues

Release cadence

Juggling two jobs

Dealing with people

Choice of email
- Inbox zero? lol

Solutions?
- Build up a group of trusted reviewers
- Maintainer group
- Doing too much? Give it to someone else

Hard to find a new/co maintainer

New contributor

All patch review still done over email

- Party like 1999. Email is increasingly unreliable
- **b4** great for reviewing/applying patches
- **lei** [1] / lore for local review
- **git** send-email for sending patches

Email client setup

- `Documentation/process/email-clients.rst`
- **git** send-email for replies (with Message-Id)

scripts/get_maintainer.pl

Help review other changes!

[1] https://josefbacik.github.io/kernel/2021/10/18/lei-and-b4.html

Upstreaming changes

Nobody cares about your needs

- Do something good for the project as a prep
- Create shared goals

Nobody pays attention to your change

- Deliberately half-ass the change

Upstream maintainer is lukewarm on your change

- Limbo

Upstream maintainer is unresponsive

Get Mariokart'ed

Upstreaming changes

Complicated patchsets

Changes across subsystem boundaries

Changes that depend on someone elses changes

- When are those going upstream?
- Help get dependencies upstream
- How to manage these dependencies

“[PATCHSET v39 0/17] Solve world peace and hunger”

- If possible, break into multiple smaller series

PR checklist

Make sure the code compiles, no warnings

- Common sense not so common

Is this a forward ported fix?

Good commit messages

- Explain the WHY
- If it’s a fix, explain why the current code is broken

Going all in

```
$ git show -s --format=%b 11933cf1d91d57da9e5c53822a540bbdc2656c16
| wc -l
615

$ git show 11933cf1d91d57da9e5c53822a540bbdc2656c16 | diffstat -s
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Going all in

```
$ git show -s --format=%b 11933cf1d91d57da9e5c53822a540bbdc2656c16
| wc -l
615

$ git show 11933cf1d91d57da9e5c53822a540bbdc2656c16 | diffstat -s
 1 file changed, 1 insertion(+), 1 deletion(-)

$ git show 11933cf1d91d57da9e5c53822a540bbdc2656c16 | head -n5
commit 11933cf1d91d57da9e5c53822a540bbdc2656c16
Author: Christian Brauner <brauner@kernel.org>
Date:   Sat Dec 17 22:28:40 2022 +0100

    pnode: terminate at peers of source
```

PR checklist

Make sure the code compiles, no warnings

- Common sense not so common

Is this a forward ported fix?

Good commit messages

- Explain the WHY
- If it’s a fix, explain why the current code is broken
- Don’t make the maintainer rewrite it. Most don’t.

Double check!

- Surely it’s just a trivial change...

Double check

```
$ git show e1dd23a0012c3929737798fda9fede0e783f4ff3

commit e1dd23a0012c3929737798fda9fede0e783f4ff3
Author: Jens Axboe <axboe@kernel.dk>
Date:   Wed Jun 8 13:02:25 2005 +0200

    [PATCH] sata_sil: Fix FIFO PCI Bus Arbitration kernel oops

    Correct this.

    diff --git a/drivers/scsi/sata_sil.c b/drivers/scsi/sata_sil.c

diff --git a/drivers/scsi/sata_sil.c b/drivers/scsi/sata_sil.c
[...]
```

PR checklist

Make sure the code compiles, no warnings

- Common sense not so common

Is this a forward ported fix?

Good commit messages

- Explain the WHY
- If it’s a fix, explain why the current code is broken
- Don’t make the maintainer rewrite it. Most don’t.

Double check!

- Surely it’s just a trivial change...

Testing

- Make it clear how it was tested
- Surrounding code changed?
- Does a test suite exist? Use it

Trust is hard go gain, easy to lose

Write a good cover letter

Questions?

Meta