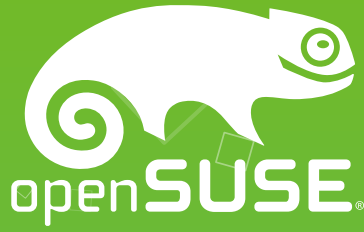


Upstream Collaboration and Linux Distributions Collaboration – Is that excluded?

Sarah Julia Kriesch

openSUSE

sarah.kriesch@opensuse.org

AdaLovelace on LiberaChat

@sjkriesch

The Linux Distributions Working Group @ The Open Mainframe Project

www.openmainframeproject.org

Agenda

- About me
- Mainframes
- The Open Mainframe Project
- The Linux Distributions Working Group
- Collaboration of Linux Distributions
- How to include Upstream Projects
- Q & A

About me

- Sarah Julia Kriesch
- openSUSE Contributor since around 10 years
- Member of the Release Engineering Team (s390x)
- Teamlead for s390x
- Bachelor Thesis at IBM
- DevOps Consultant (+ Open Source Contributor) at Accenture
- Founder of the Linux Distributions Working Group

Mainframes

- Large high-performance computer systems
- Big Endians
- Architecture s390x
- Used for mission-critical data
- Thousands of VMs can run on such a system

https://www.ibm.com/it-infrastructure/z/hardware

The Open Mainframe Project

- Founded 2015
- Focal point for deployment and usage of Linux and Open Source in a mainframe computing environment
- Project under the Linux Foundation

Mainframe centric Projects

OPEN MAINFRAME PROJECT
Zowe

OPEN MAINFRAME PROJECT
ADE

OPEN MAINFRAME PROJECT
GenevaERS
The Single-Pass Optimization Engine

OPEN MAINFRAME PROJECT
Ambitus

OPEN MAINFRAME PROJECT
ZOROW

OPEN MAINFRAME PROJECT
Feilong

OPEN MAINFRAME PROJECT
Tessia

OPEN MAINFRAME PROJECT
Software Discovery Tool

OPEN MAINFRAME PROJECT
ATOM
Language Syntax Highlighting for z/VM

OPEN MAINFRAME PROJECT
MENTORSHIP

OPEN MAINFRAME PROJECT
COBOL
Programming Course

Working Groups

OPEN MAINFRAME PROJECT

COBOL

Working Group

OPEN MAINFRAME PROJECT

OPEN z/OS ENABLEMENT

Working Group

OPEN MAINFRAME PROJECT

Linux Distributions

Working Group

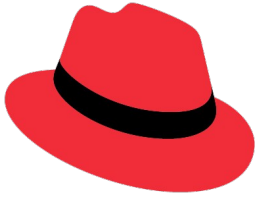
Linux Distributions Working Group

Rocky Linux™

ubuntu®

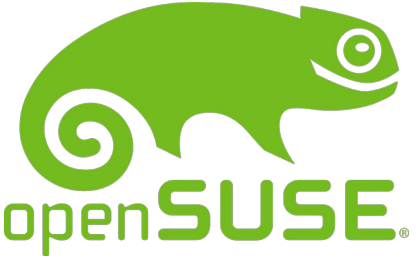
AlmaLinux

debian



Red Hat

IBM



Structure

- Founders/ (Co-)Chairs:
  - Sarah Julia Kriesch (openSUSE, Accenture)
  - Elizabeth K. Joseph (IBM)
- One Representative for every Linux distribution (required for input)
- Sponsor: SUSE

Our Goals

- Create a place to collaborate across Linux Distributions via an OMP mailing list, wiki, and chat
- Provide a space for distributions to request for help on their port
- Ensure any and all infrastructure required is available for supporting the ports
- Better support from IBM to fix s390x specific bugs

https://wiki.openmainframeproject.org/display/LinuxDistrosWG

Our collaborative process

- Problem discussions on mailing lists
- Reproducing issues (sometimes)
- Open discussions on the mailing list
- Forwarding issues/ideas of improvement for IBM
- Monthly meetings (come together) with review

Collaboration as a benefit

- Upstream contributions available for all
- Lowering research & development costs

  (at IBM and in the community)
- Same solutions for all Linux distributions
- Sharing knowledge between communities
- Increasing innovation (diverse community ideas)
- Accelerating Linux development for s390x

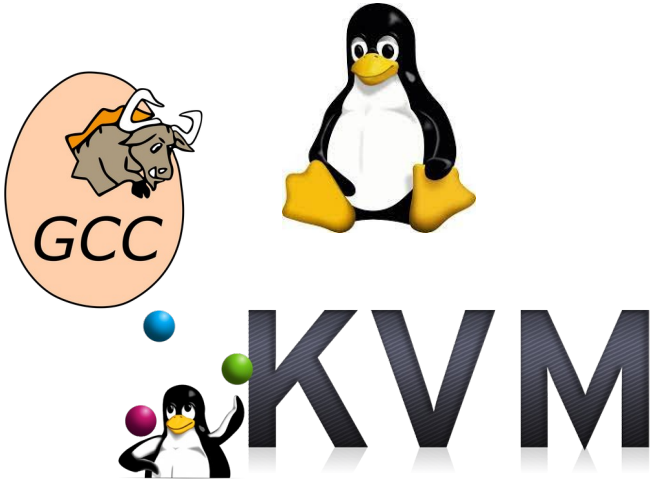
You can achieve more together than alone!

LinuxONE (OSS) Community Cloud

- Vms on LinuxONE sponsored by IBM
- Free access (120 days) for single open-source contributors (SLES, RHEL, Ubuntu)
- Long-term access for open-source projects available
- https://linuxone.cloud.marist.edu/#/login

Idea to include Upstream Projects

- Easy to include are base projects (examples):
  - Linux kernel
  - gcc
  - KVM



But there are many other projects in Linux included!

Example gcc

Richard Biener 2022-06-27 09:22:11 UTC Description

Testcase reduced from opie

```
> ./cc1 -quiet /tmp/y.tab.i -O -I include -w

during RTL pass: combine
/tmp/y.tab.i: In function 'yyparse':
/tmp/y.tab.i:59:1: internal compiler error: Segmentation fault
   59 | }
      | ^
0x14010ac crash_signal
        /space/rguenther/src/gcc/gcc/toplev.cc:322
0x21932ef reg_bitfield_target_p
        /space/rguenther/src/gcc/gcc/combine.cc:14142
0x219473f distribute_notes
        /space/rguenther/src/gcc/gcc/combine.cc:14653
0x2173f8f try_combine
        /space/rguenther/src/gcc/gcc/combine.cc:4485
0x216a2f6 combine_instructions
        /space/rguenther/src/gcc/gcc/combine.cc:1268
0x2195541 rest_of_handle_combine
        /space/rguenther/src/gcc/gcc/combine.cc:14976
0x2195602 execute
        /space/rguenther/src/gcc/gcc/combine.cc:15021
Please submit a full bug report, with preprocessed source (by using -freport-bug).
Please include the complete backtrace with any bug report.
See <https://gcc.gnu.org/bugs/> for instructions.
```

Richard Biener 2022-06-27 09:22:57 UTC Comment 1

Created attachment 53207 [details]
reduced testcase

Richard Biener 2022-06-27 09:26:55 UTC Comment 2

```
Program received signal SIGSEGV, Segmentation fault.
0x00000000021932ef in reg_bitfield_target_p (x=0x7ffff6a4ee10, body=0x7ffff6a4e450)
at /space/rguenther/src/gcc/gcc/combine.cc:14142
14142         if (GET_CODE (target) == SUBREG)
(gdb) p target
$1 = (rtx) 0xafafaf0100000047
```

Andreas Krebbel 2022-07-14 10:02:24 UTC Comment 10

We generate the movstrict target operand with gen_lowpart. If the operand for gen_lowpart is already a paradoxical subreg the two subregs cancel each other out and we end up with a plain reg. I'm testing the following patch right now. It falls back to a normal move in that case and fixes the testcase:

```
diff --git a/gcc/config/s390/s390.cc b/gcc/config/s390/s390.cc
index 5aaf76a9490..d90ec1a6de1 100644
--- a/gcc/config/s390/s390.cc
+++ b/gcc/config/s390/s390.cc
@@ -6523,6 +6523,14 @@ s390_expand_insv (rtx dest, rtx op1, rtx op2, rtx src)
          rtx low_dest = gen_lowpart (smode, dest);
          rtx low_src = gen_lowpart (smode, src);

+         /* In case two subregs cancelled each other out, do a normal
+            move.  */
+         if (!SUBREG_P (low_dest))
+           {
+             emit_move_insn (low_dest, low_src);
+             return true;
+           }
+
          switch (smode)
            {
            case E_QImode: emit_insn (gen_movstrictqi (low_dest, low_src)); return
true;
```

https://gcc.gnu.org/bugzilla/show_bug.cgi?id=106101

Process at gcc in this case

- Upstream bug report
- IBM Maintainers are receiving all s390x specific
- Maintainers of IBM are interacting

  - > Hint: They are also open to join our Linux Distributions Working Group for discussions!

Example with qore s390x Enablement

davidnich commented on Dec 12, 2022 • edited — Contributor

@skriesch with the "best" set of workarounds (not really a fix, because I don't understand the root cause), I have it working on all s390 builds except openSES Factory zSystems. I'm still looking for a way to make it work there.

My s390x qemu VM is unfortunately not usable for that, as building the RPM crashes there - although manual builds work and were partially useful for debugging - just very slow and in some cases the real HW on build.opensuse.org would show different behavior. So currently I'm pushing workaround attempts to build.opensuse.org and hoping that I can find the right combination that will make it work in all cases.

The problem is that for some reason qore is not able to get the pointer to the current stack position reliably in new threads on s390x. It looks like it's related to optimizations somehow, and ensuring a non-inlined call to the function that sets up this data worked up to f37, but is not working on openSUSE Tumbleweed zSystems - I have the feeling that the newer the compiler, the more aggressive the optimizations related to the stack.

skriesch commented on Dec 12, 2022 — Author

I can forward this issue to IBM for s390x support.

davidnich commented on Dec 12, 2022 — Contributor

@skriesch that would be awesome - I would be very happy for any support on how to do this correctly - basically what Qore does is get the current stack address at the start of each thread - this is done as follows:

```
static inline size_t get_stack_pos() {
    size_t addr;
    __asm("lr %0, 15" : "=r" (addr) );
    return addr;
}
```

Qore also knows the size of the thread's stack. At runtime, Qore will compare the current stack position against the limit for the thread and throw an exception if the position gets too close to the stack's limit.

The problem is that the call to `get_stack_pos()` at the start of the thread does not work properly on openSUSE Tumbleweed zSystems. I have no idea why. Also my qemu s390x VM does not appear to be reliable enough to use for debugging this problem - also it's very slow.

Thanks a lot in advance for any support you can arrange for this issue - I'd be very happy to make this work properly.

Hello together,

I have seen, that the programming language qore does not work any more an s390x.
I have created this issue: https://github.com/qorelanguage/qore/issues/4655

The Developer david tried to enable it. The crazy situation has been, that his solution has worked with older compilers on openSUSE Leap 15.4.
He has figured out a solution for Fedora 37 as a workaround. But the same issue has been happening with a latest compiler version on openSUSE Tumbleweed.

Statement:
"@skriesch with the "best" set of workarounds (not really a fix, because I don't understand the root caus), I have it working on all s390 builds except openSES Factory zSystems. I'm still looking for a way to make it work there.

My s390x qemu VM is unfortunately not usable for that, as building the RPM crashes there - although manual builds work and were partially useful for debugging - just very slow and in some cases the real HW on build.opensuse.org would show different behavior. So currently I'm pushing workaround attempts to build.opensuse.org and hoping that I can find the right combination that will make it work in all cases.

The problem is that for some reason qore is not able to get the pointer to the current stack position reliably in new threads on s390x. It looks like it's related to optimizations somehow, and ensuring a non-inlined call to the function that sets up this data worked up to f37, but is not working on openSUSE Tumbleweed zSystems - I have the feeling that the newer the compiler, the more aggressive the optimizations related to the stack."

Error message:
```
[ 199s] STACK-LIMIT-EXCEEDED: this thread's stack has exceeded the stack size limit (516096 bytes)
[ 199s] call stack:
[ 199s]  2: MaxThreadCountTest::t() (examples/test/qore/threads/max-threads-count.qtest:44 (Qore user code))
[ 199s]  1: "thread start"
[ 199s] QUnit Test "Max thread count test" v1.0
[ 199s] FAILURE: Test max thread count: 3 assertions, 2 succeeded
[ 199s] Assertion failure at examples/test/qore/threads/max-threads-count.qtest:57 [MaxThreadCountTest::testMaxThreadCount()] <- examples/test/qore/threads/max-threads-count.qtest:20 [MaxThreadCountTest::constructor()]
```

I would be happy, if IBM can support us here.

uweigand commented on Dec 12, 2022

Hi **@davidnich** it seems the core problem is this:

```
    __asm("lr %0, 15" : "=r" (addr) );
```

which copies only the low 32 bits of the stack pointer (instead of all 64 bits). This may still appear to work if the destination register just so happened to have the correct upper 32 bits already (e.g. if it previously held a pointer to something on the stack). That's probably why you might accidentally get the correct result depending on the exact compiler version and random code changes.

You should be doing

```
    __asm("lgr %0, 15" : "=r" (addr) );
```

instead. Once you've done that, none of the other s390-specific changes should be necessary at all.

1 1

davidnich commented on Dec 12, 2022 • edited — Contributor

@uweigand great - thanks a lot!!!

https://github.com/qorelanguage/qore/issues/4655

Reasons for such required forwarding

- Many open source projects everywhere
- IBM does not know all Linux integrated software (especially new ones)
- IBM Maintainers are only for base projects (strategic open source projects) available
- Linux Distribution Maintainers know their requirements
- Connection between IBM and upstream projects is missing

Do you want to join?

- Wiki:

  https://wiki.openmainframeproject.org/display/LinuxDistrosWG/Linux+Distributions+Working+Group

- ML:

  https://lists.openmainframeproject.org/g/wg-linux-distros

- Monthly meetings on second Tuesday (invitations are sent to the mailing list)

Questions?

How do you want to get involved as Upstream Projects?


openSUSE®