CHINA LINUX KERNEL
中国Linux内核开发者大会

vivo

Improvements to Eppic and Rust support in the crash-utility

Lianbo Jiang
红帽高级工程师

CHINA LINUX KERNEL
中国Linux内核开发者大会

目录

CONTENTS

CHINA LINUX KERNEL
中国Linux内核开发者大会

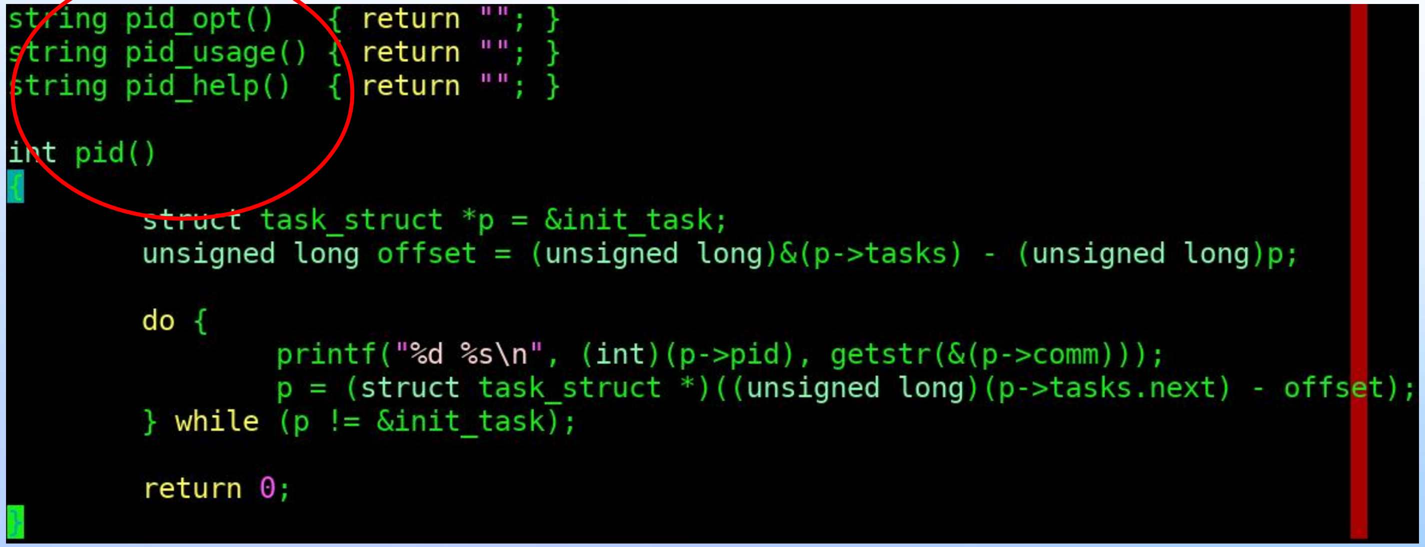
Improvements to Eppic

1) what is eppic
2) what’ s the relationship between crash and eppic
3) Why improve eppic
4) How to use eppic

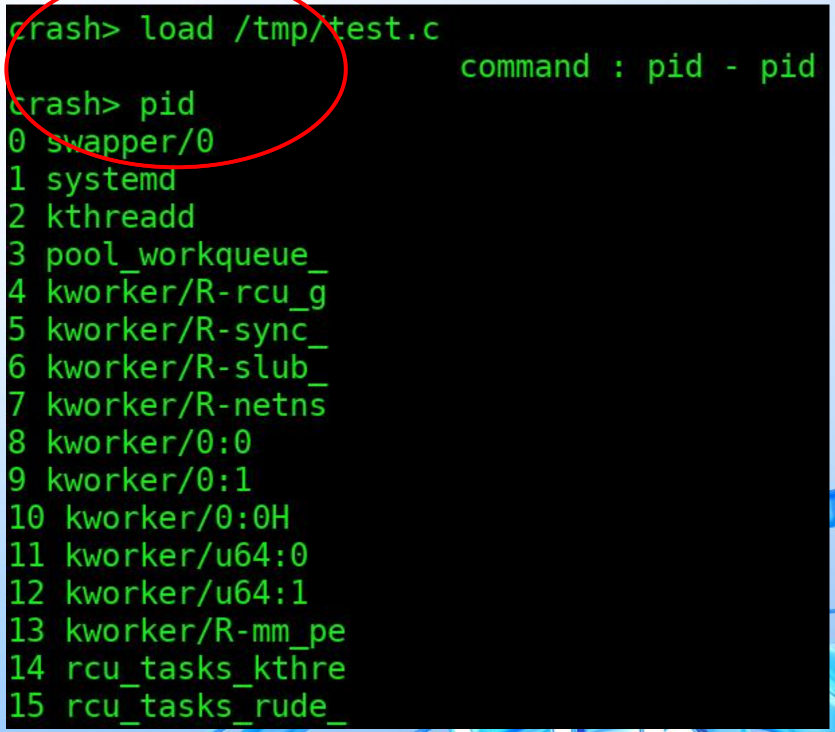
What is eppic

- Embedable Pre-Processor and Interpreter for C

1) A c-like language interpreter, pre-improvement:



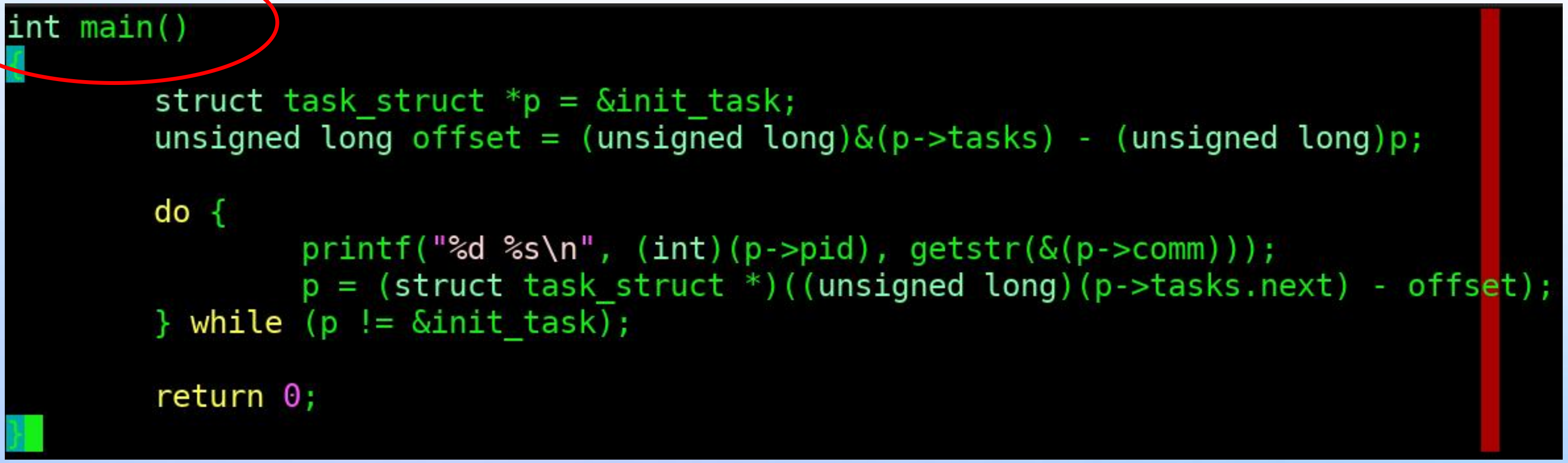
```
string pid_opt()   { return ""; }
string pid_usage() { return ""; }
string pid_help()  { return ""; }

int pid()
{
        struct task_struct *p = &init_task;
        unsigned long offset = (unsigned long)&(p->tasks) - (unsigned long)p;

        do {
                printf("%d %s\n", (int)(p->pid), getstr(&(p->comm)));
                p = (struct task_struct *)((unsigned long)(p->tasks.next) - offset);
        } while (p != &init_task);

        return 0;
}
```



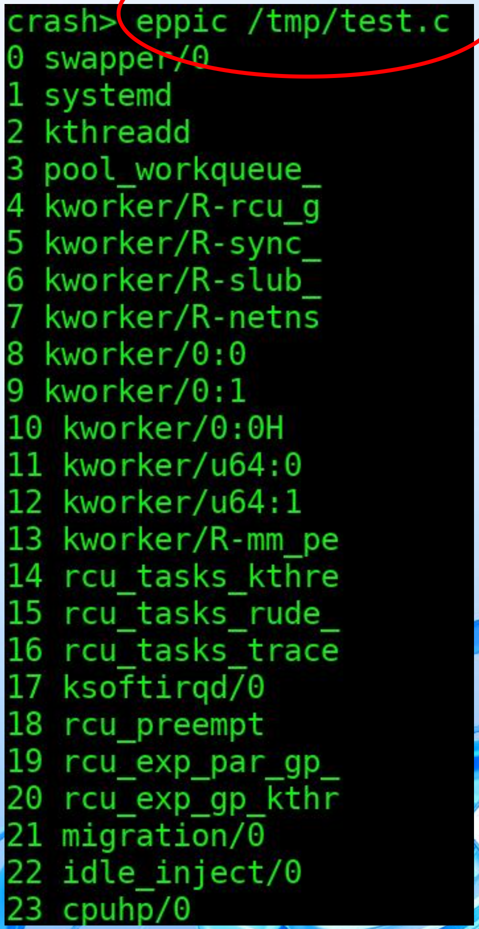
```
crash> load /tmp/test.c
                        command : pid - pid
crash> pid
0 swapper/0
1 systemd
2 kthreadd
3 pool_workqueue_
4 kworker/R-rcu_g
5 kworker/R-sync_
6 kworker/R-slub_
7 kworker/R-netns
8 kworker/0:0
9 kworker/0:1
10 kworker/0:0H
11 kworker/u64:0
12 kworker/u64:1
13 kworker/R-mm_pe
14 rcu_tasks_kthre
15 rcu_tasks_rude_
```

What is eppic

1) A c-like language interpreter, after improvement:



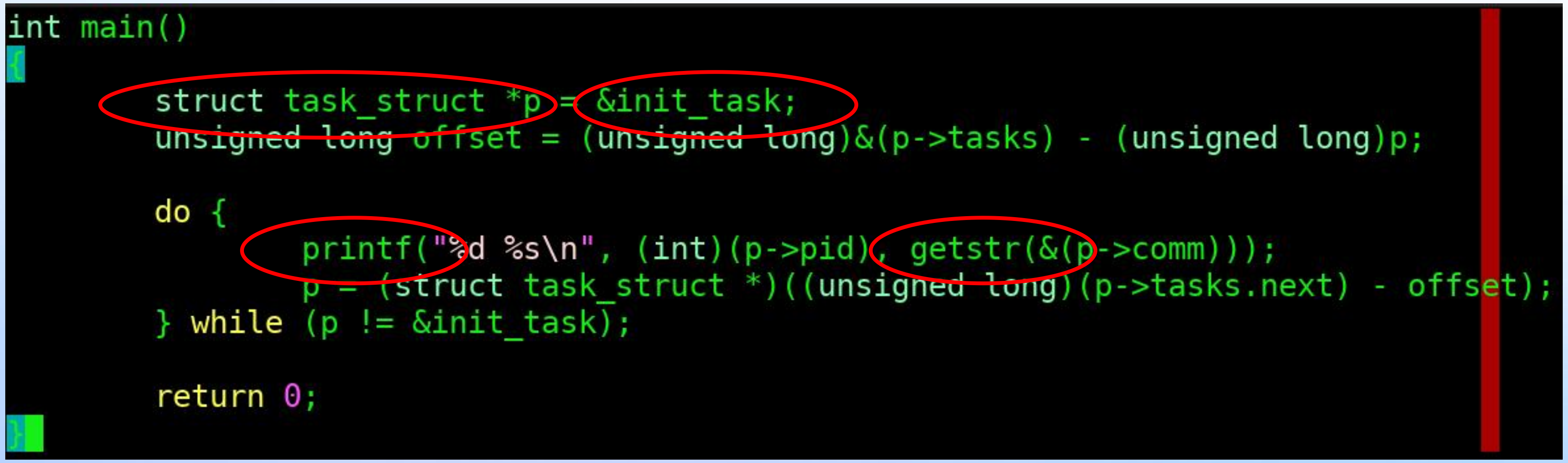
```
int main()
{
        struct task_struct *p = &init_task;
        unsigned long offset = (unsigned long)&(p->tasks) - (unsigned long)p;

        do {
                printf("%d %s\n", (int)(p->pid), getstr(&(p->comm)));
                p = (struct task_struct *)((unsigned long)(p->tasks.next) - offset);
        } while (p != &init_task);

        return 0;
}
```



```
crash> eppic /tmp/test.c
0 swapper/0
1 systemd
2 kthreadd
3 pool_workqueue_
4 kworker/R-rcu_g
5 kworker/R-sync_
6 kworker/R-slub_
7 kworker/R-netns
8 kworker/0:0
9 kworker/0:1
10 kworker/0:0H
11 kworker/u64:0
12 kworker/u64:1
13 kworker/R-mm_pe
14 rcu_tasks_kthre
15 rcu_tasks_rude_
16 rcu_tasks_trace
17 ksoftirqd/0
18 rcu_preempt
19 rcu_exp_par_gp_
20 rcu_exp_gp_kthr
21 migration/0
22 idle_inject/0
23 cpuhp/0
```

What is eppic

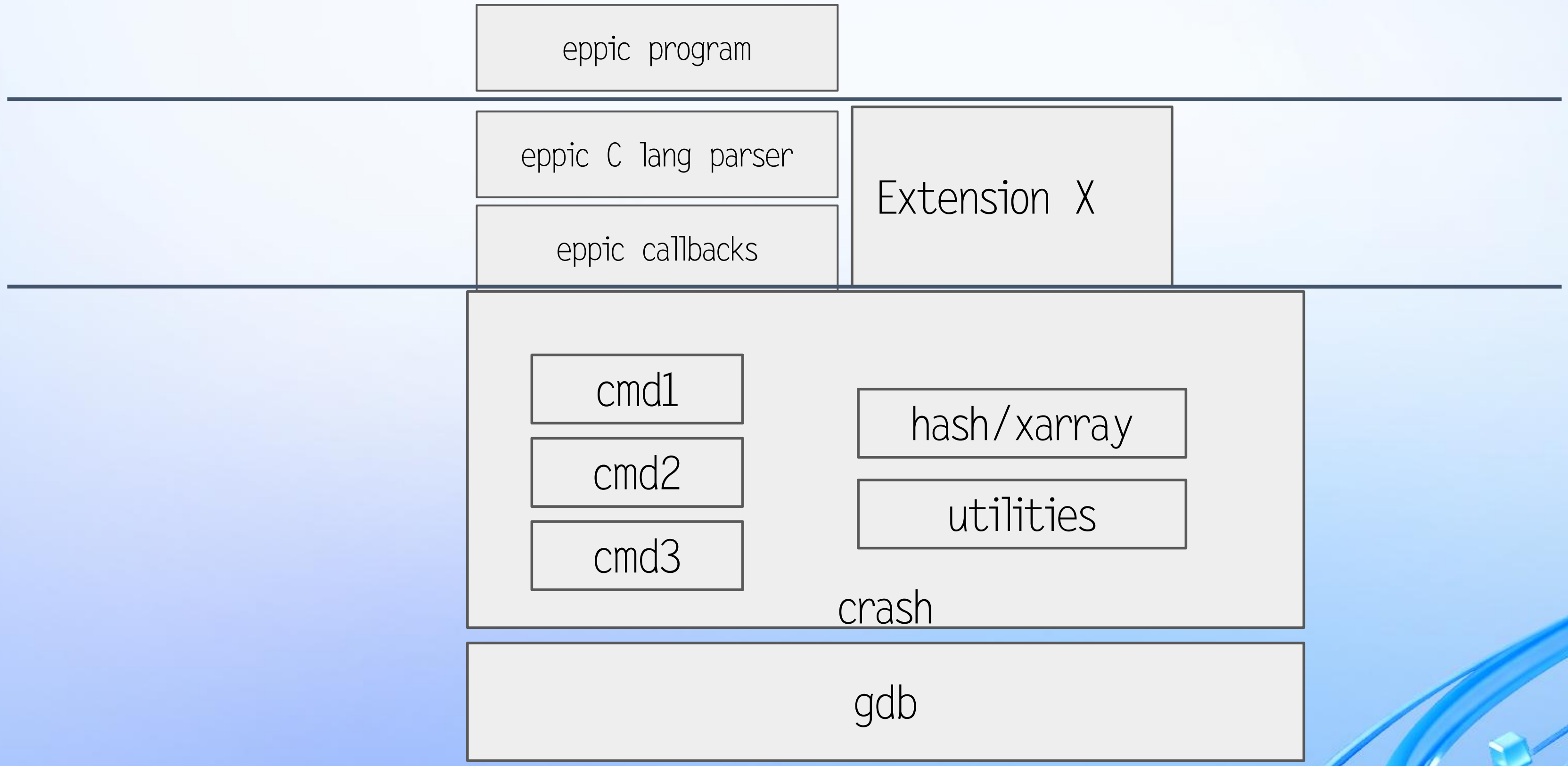
Let’ s take a deeper look at the eppic program:



```
int main()
{
        struct task_struct *p = &init_task;
        unsigned long offset = (unsigned long)&(p->tasks) - (unsigned long)p;

        do {
                printf("%d %s\n", (int)(p->pid), getstr(&(p->comm)));
                p = (struct task_struct *)((unsigned long)(p->tasks.next) - offset);
        } while (p != &init_task);

        return 0;
}
```

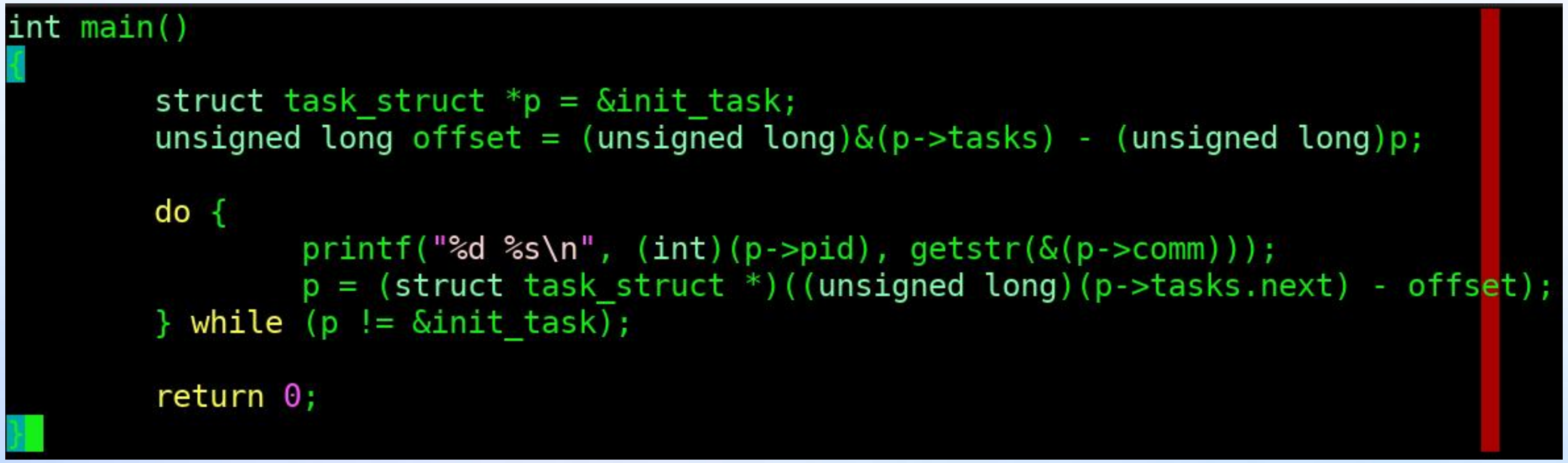
CHINA LINUX KERNEL
中国Linux内核开发者大会

What’ s the relationship between crash and eppic



Why improve eppic

Let’ s look again at the program, doesn’ t it look similar to a kernel program?



```
int main()
{
        struct task_struct *p = &init_task;
        unsigned long offset = (unsigned long)&(p->tasks) - (unsigned long)p;

        do {
                printf("%d %s\n", (int)(p->pid), getstr(&(p->comm)));
                p = (struct task_struct *)((unsigned long)(p->tasks.next) - offset);
        } while (p != &init_task);

        return 0;
}
```

That’ s the beauty of eppic program:

1) Write kernel-c-like programs to inspect kernel, easy for kernel developers. Considering how to iterate linked-list in python?

2) Coding like within kernel, and without kernel’ s constraints, no need to worry kernel’ s crash.

How to use eppic

1) compile crash and eppic:

```
    $ make -j<thread_num> lzo zstd snappy
```

and

```
    $ make extensions
```

2) crash loading for core dump or live debugging:

```
    $ crash vmcore vmlinux
```

or

```
    $ crash /proc/kcore vmlinux
    $ extend extensions/eppic.so
    $ vim <your_program.c> or
    crash> edit -f <your_program.c> # there will be code samples/references as code comment
```

3) run eppic program:

```
    crash> eppic <your_program.c>
```

Reference:

A) Conservative improvement

B) Aggressive improvement

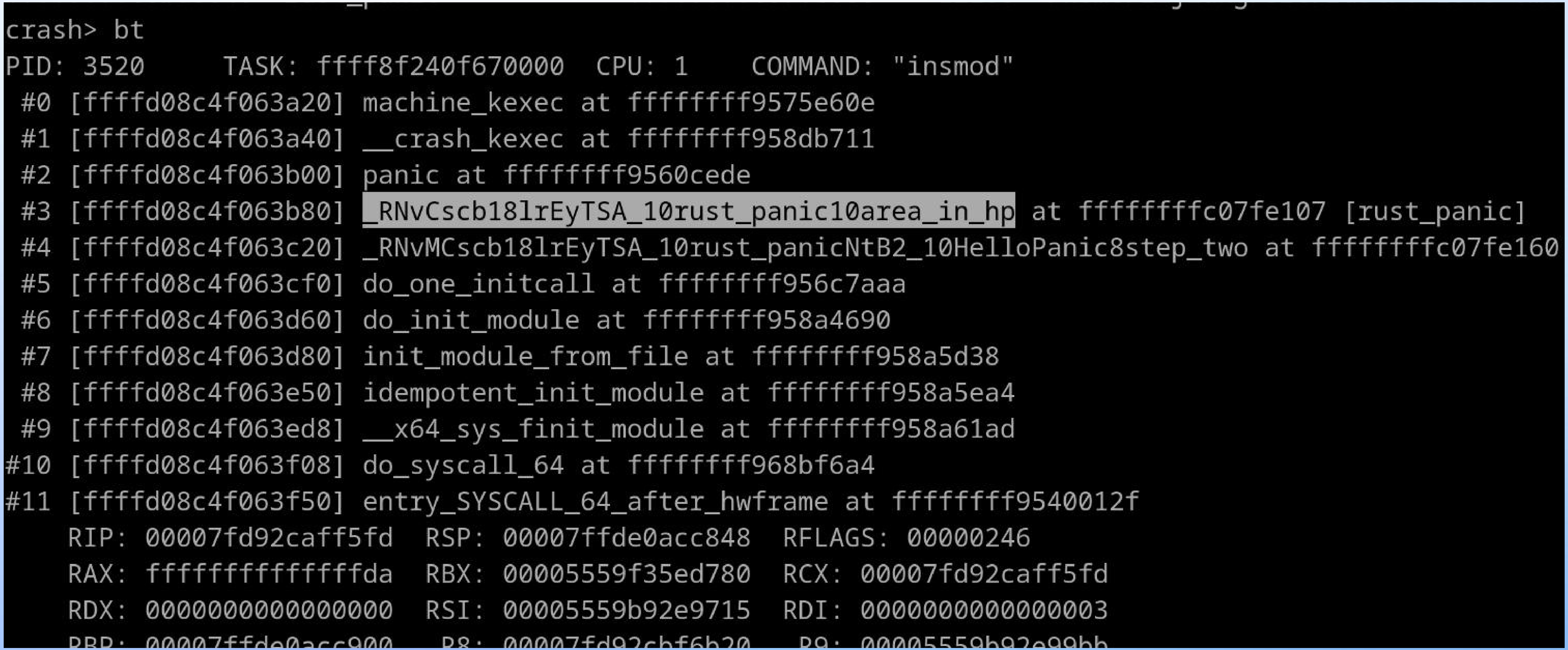
Rust support in crash-utility

– Why does the crash-utility need to support Rust?

- Linux kernel is adding Rust support, kernel modules and drivers can be written in Rust.
- Rust is becoming part of the stack we need to inspect, without Rust-aware support, crash-utility won’ t give useful information for Rust programs/drivers.
- Without demangling, you will see long mangled symbols(not human-readable symbols).
- As Rust adoption grows in kernel, not supporting it makes crash tool less useful for a growing class of incidents and users.
- Better diagnostics, human-readable symbols and backtraces make debugging faster and more accurate. Let debugging focus on the bug itself.

Rust support in crash-utility

- What would you see if Rust is not supported?

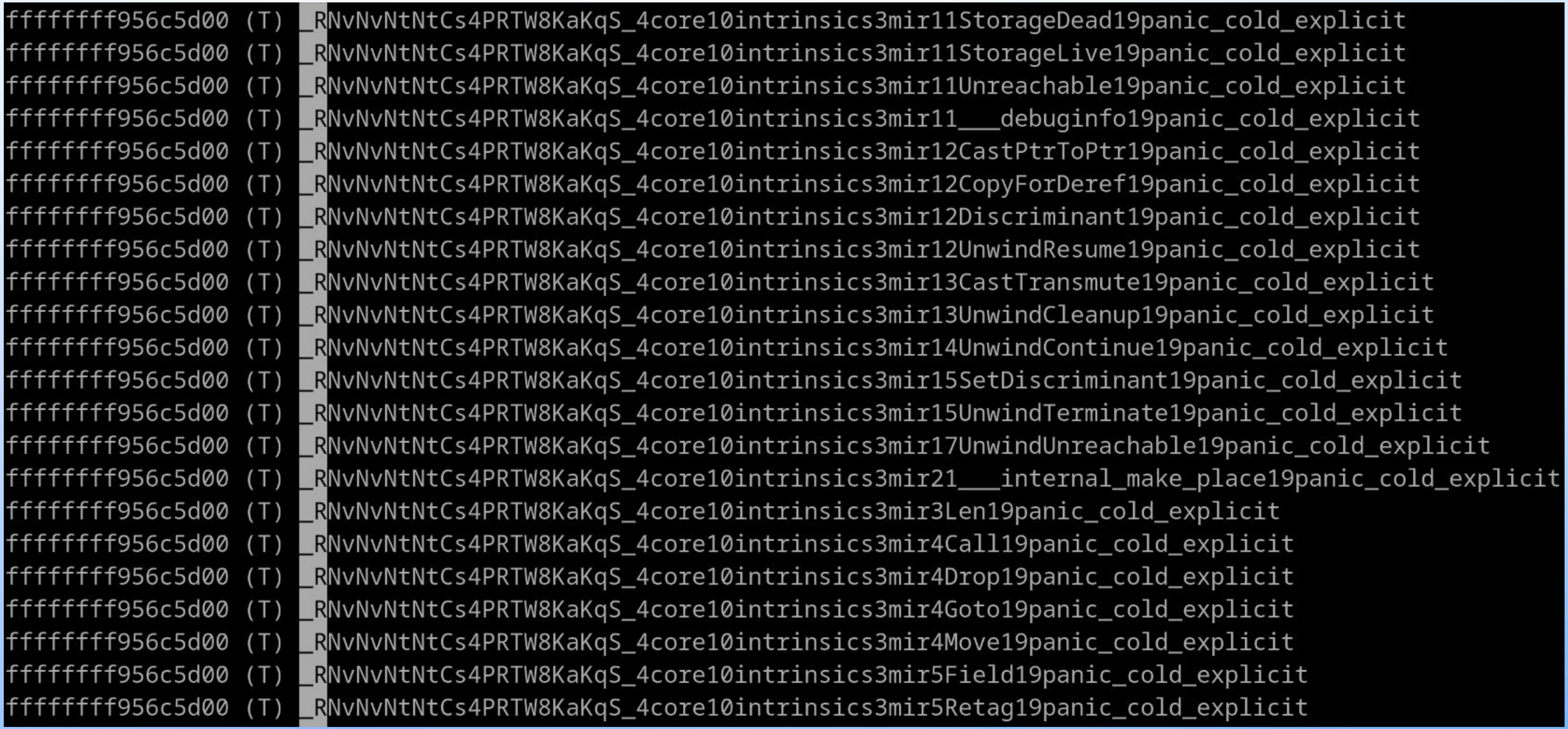


```
crash> bt
PID: 3520     TASK: ffff8f240f670000  CPU: 1    COMMAND: "insmod"
 #0 [ffffd08c4f063a20] machine_kexec at ffffffff9575e60e
 #1 [ffffd08c4f063a40] __crash_kexec at ffffffff958db711
 #2 [ffffd08c4f063b00] panic at ffffffff9560cede
 #3 [ffffd08c4f063b80] _RNvCscb18lrEyTSA_10rust_panic10area_in_hp at ffffffffc07fe107 [rust_panic]
 #4 [ffffd08c4f063c20] _RNvMCscb18lrEyTSA_10rust_panicNtB2_10HelloPanic8step_two at ffffffffc07fe160
 #5 [ffffd08c4f063cf0] do_one_initcall at ffffffff956c7aaa
 #6 [ffffd08c4f063d60] do_init_module at ffffffff958a4690
 #7 [ffffd08c4f063d80] init_module_from_file at ffffffff958a5d38
 #8 [ffffd08c4f063e50] idempotent_init_module at ffffffff958a5ea4
 #9 [ffffd08c4f063ed8] __x64_sys_finit_module at ffffffff958a61ad
#10 [ffffd08c4f063f08] do_syscall_64 at ffffffff968bf6a4
#11 [ffffd08c4f063f50] entry_SYSCALL_64_after_hwframe at ffffffff9540012f
    RIP: 00007fd92caff5fd  RSP: 00007ffde0acc848  RFLAGS: 00000246
    RAX: ffffffffffffffda  RBX: 00005559f35ed780  RCX: 00007fd92caff5fd
    RDX: 0000000000000000  RSI: 00005559b92e9715  RDI: 0000000000000003
    RBP: 00007ffde0acc900   R8: 00007fd92cbf6b20   R9: 00005559b92e99bb
```

CHINA LINUX KERNEL
中国Linux内核开发者大会

Rust support in crash-utility

- What would you see if Rust is not supported?

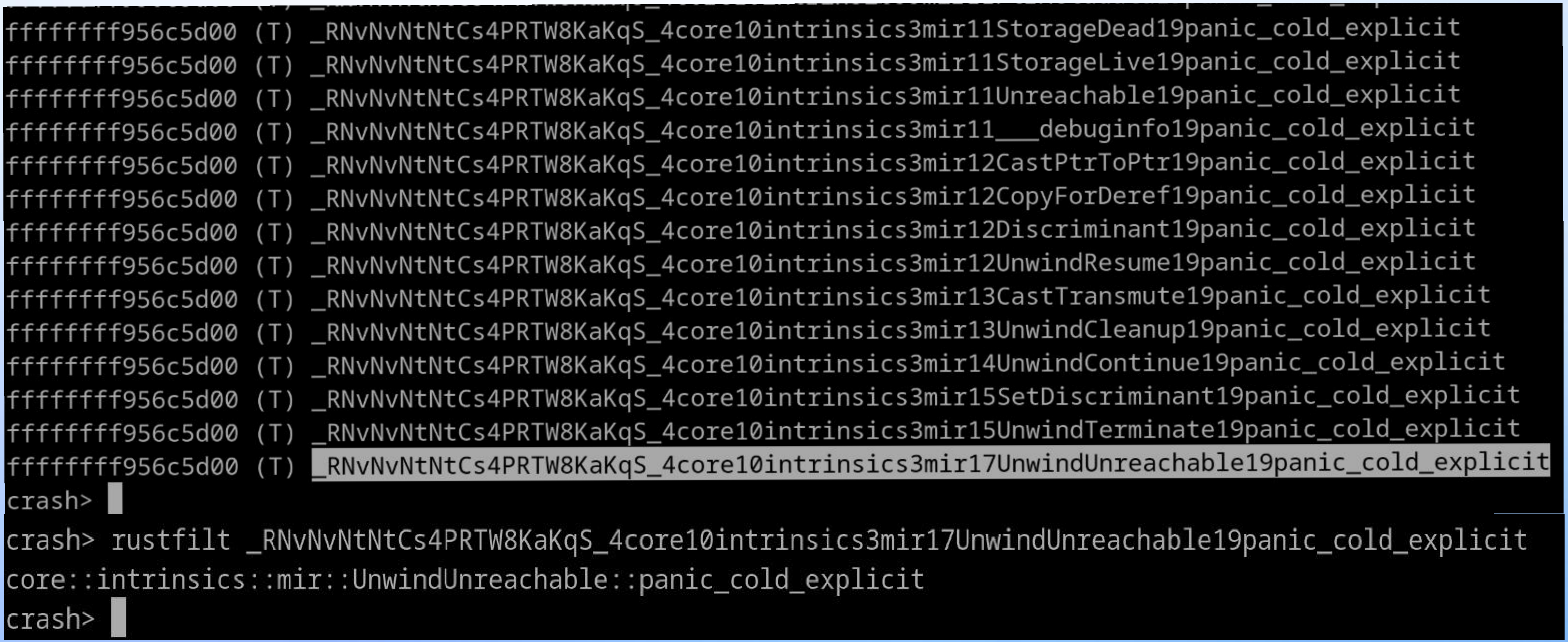


```
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir11StorageDead19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir11StorageLive19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir11Unreachable19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir11___debuginfo19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir12CastPtrToPtr19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir12CopyForDeref19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir12Discriminant19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir12UnwindResume19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir13CastTransmute19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir13UnwindCleanup19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir14UnwindContinue19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir15SetDiscriminant19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir15UnwindTerminate19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir17UnwindUnreachable19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir21___internal_make_place19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir3Len19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir4Call19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir4Drop19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir4Goto19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir4Move19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir5Field19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir5Retag19panic_cold_explicit
```

Rust support in crash-utility

– Implementation details

● Add rustfilt command to demangle Rust symbol names

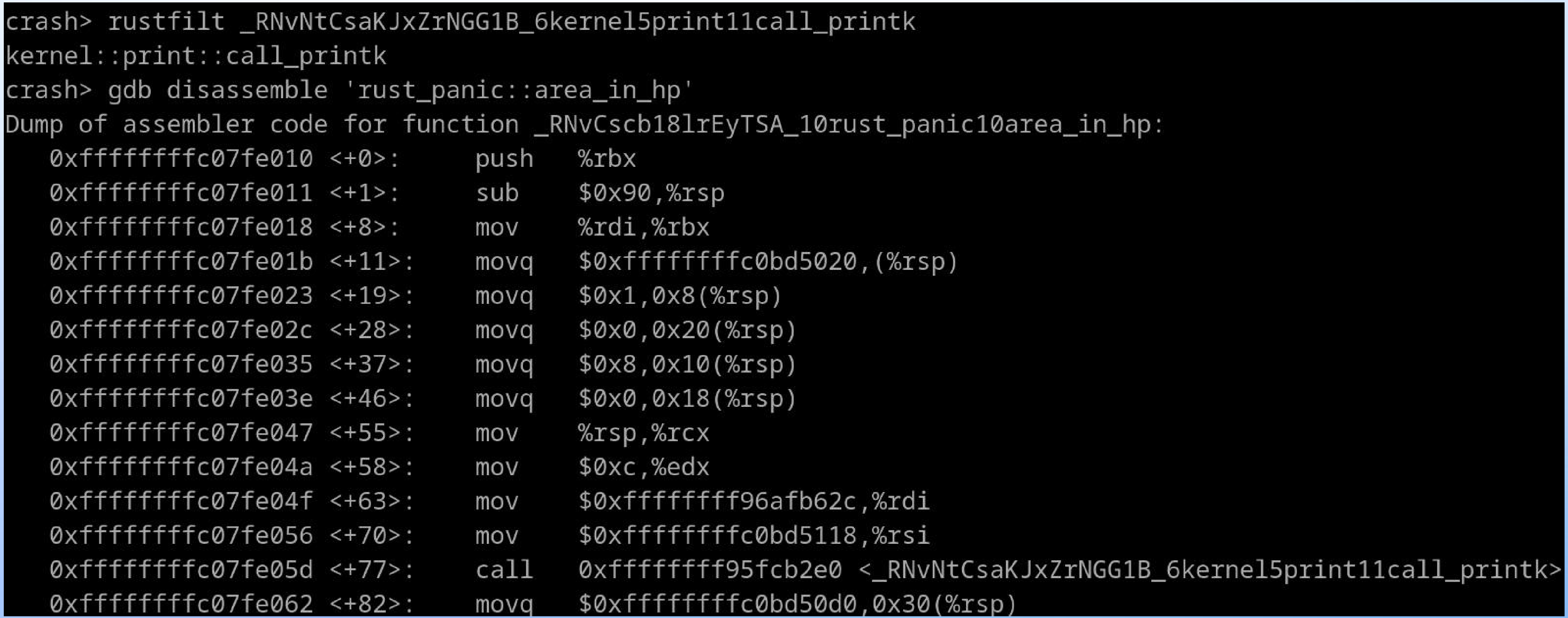


```
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir11StorageDead19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir11StorageLive19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir11Unreachable19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir11___debuginfo19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir12CastPtrToPtr19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir12CopyForDeref19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir12Discriminant19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir12UnwindResume19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir13CastTransmute19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir13UnwindCleanup19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir14UnwindContinue19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir15SetDiscriminant19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir15UnwindTerminate19panic_cold_explicit
ffffffff956c5d00 (T) _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir17UnwindUnreachable19panic_cold_explicit
crash>
crash> rustfilt _RNvNvNtNtCs4PRTW8KaKqS_4core10intrinsics3mir17UnwindUnreachable19panic_cold_explicit
core::intrinsics::mir::UnwindUnreachable::panic_cold_explicit
crash>
```

Rust support in crash-utility

- Implementation details

● Add rustfilt command to demangle Rust symbol names

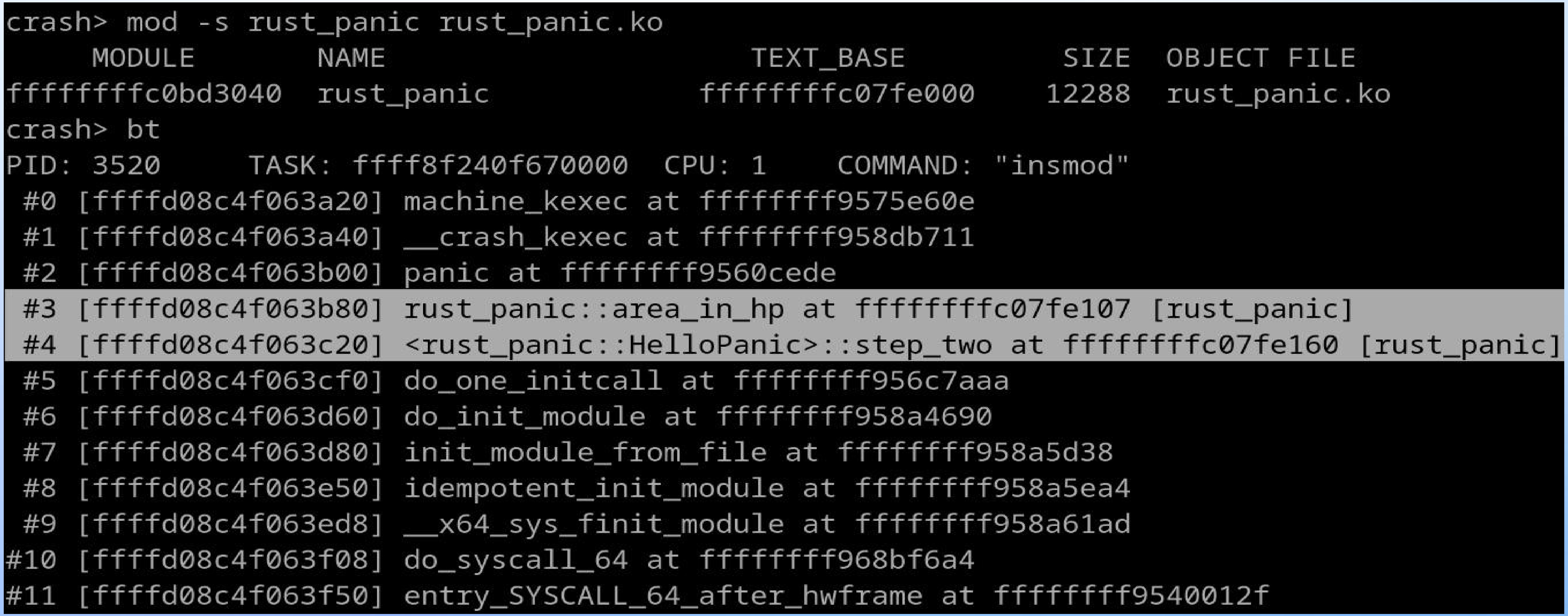


```
crash> rustfilt _RNvNtCsaKJxZrNGG1B_6kernel5print11call_printk
kernel::print::call_printk
crash> gdb disassemble 'rust_panic::area_in_hp'
Dump of assembler code for function _RNvCscb18lrEyTSA_10rust_panic10area_in_hp:
   0xffffffffc07fe010 <+0>:     push   %rbx
   0xffffffffc07fe011 <+1>:     sub    $0x90,%rsp
   0xffffffffc07fe018 <+8>:     mov    %rdi,%rbx
   0xffffffffc07fe01b <+11>:    movq   $0xffffffffc0bd5020,(%rsp)
   0xffffffffc07fe023 <+19>:    movq   $0x1,0x8(%rsp)
   0xffffffffc07fe02c <+28>:    movq   $0x0,0x20(%rsp)
   0xffffffffc07fe035 <+37>:    movq   $0x8,0x10(%rsp)
   0xffffffffc07fe03e <+46>:    movq   $0x0,0x18(%rsp)
   0xffffffffc07fe047 <+55>:    mov    %rsp,%rcx
   0xffffffffc07fe04a <+58>:    mov    $0xc,%edx
   0xffffffffc07fe04f <+63>:    mov    $0xffffffff96afb62c,%rdi
   0xffffffffc07fe056 <+70>:    mov    $0xffffffffc0bd5118,%rsi
   0xffffffffc07fe05d <+77>:    call   0xffffffff95fcb2e0 <_RNvNtCsaKJxZrNGG1B_6kernel5print11call_printk>
   0xffffffffc07fe062 <+82>:    movq   $0xffffffffc0bd50d0,0x30(%rsp)
```

Rust support in crash-utility

– Implementation details

● Demangle Rust symbol names in all outputs

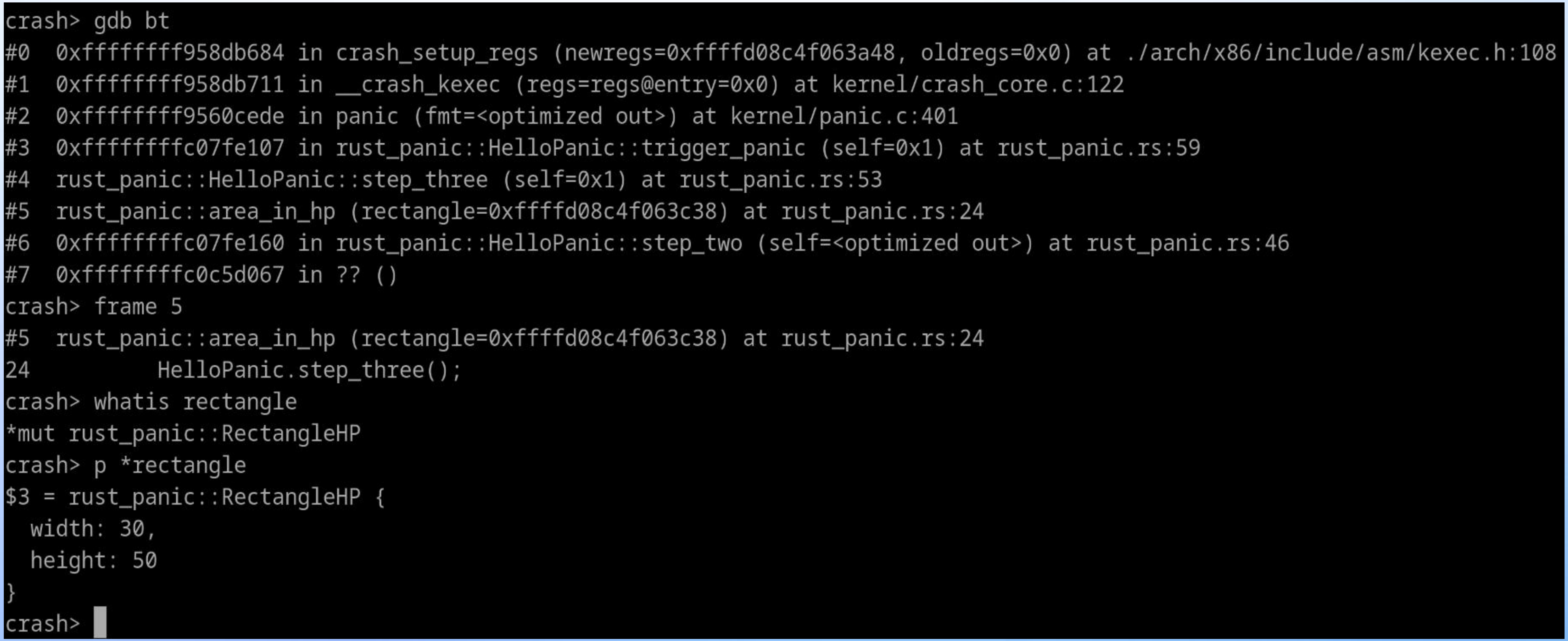


```
crash> mod -s rust_panic rust_panic.ko
     MODULE       NAME                    TEXT_BASE         SIZE  OBJECT FILE
ffffffffc0bd3040  rust_panic            ffffffffc07fe000    12288  rust_panic.ko
crash> bt
PID: 3520     TASK: ffff8f240f670000  CPU: 1    COMMAND: "insmod"
 #0 [ffffd08c4f063a20] machine_kexec at ffffffff9575e60e
 #1 [ffffd08c4f063a40] __crash_kexec at ffffffff958db711
 #2 [ffffd08c4f063b00] panic at ffffffff9560cede
 #3 [ffffd08c4f063b80] rust_panic::area_in_hp at ffffffffc07fe107 [rust_panic]
 #4 [ffffd08c4f063c20] <rust_panic::HelloPanic>::step_two at ffffffffc07fe160 [rust_panic]
 #5 [ffffd08c4f063cf0] do_one_initcall at ffffffff956c7aaa
 #6 [ffffd08c4f063d60] do_init_module at ffffffff958a4690
 #7 [ffffd08c4f063d80] init_module_from_file at ffffffff958a5d38
 #8 [ffffd08c4f063e50] idempotent_init_module at ffffffff958a5ea4
 #9 [ffffd08c4f063ed8] __x64_sys_finit_module at ffffffff958a61ad
#10 [ffffd08c4f063f08] do_syscall_64 at ffffffff968bf6a4
#11 [ffffd08c4f063f50] entry_SYSCALL_64_after_hwframe at ffffffff9540012f
```

Rust support in crash-utility

– Implementation details

● Demangle Rust symbol names in all outputs

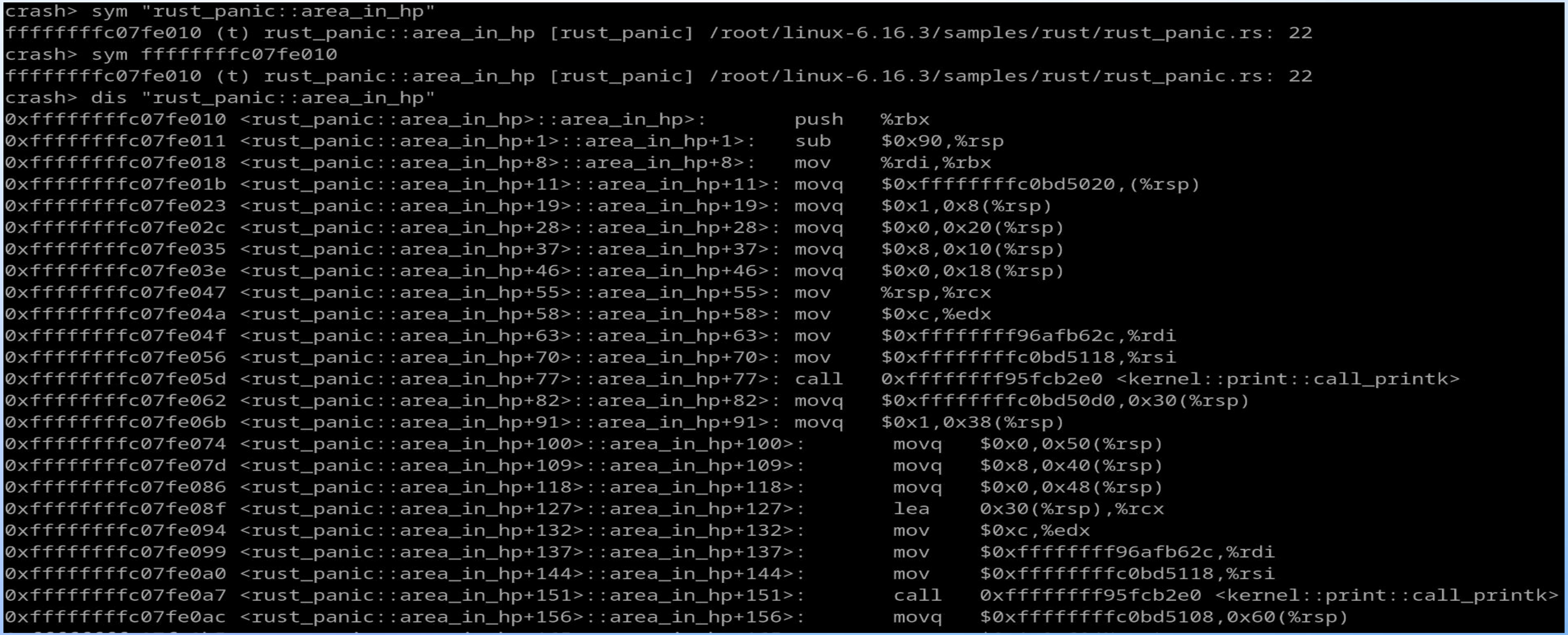


```
crash> gdb bt
#0  0xffffffff958db684 in crash_setup_regs (newregs=0xffffd08c4f063a48, oldregs=0x0) at ./arch/x86/include/asm/kexec.h:108
#1  0xffffffff958db711 in __crash_kexec (regs=regs@entry=0x0) at kernel/crash_core.c:122
#2  0xffffffff9560cede in panic (fmt=<optimized out>) at kernel/panic.c:401
#3  0xffffffffc07fe107 in rust_panic::HelloPanic::trigger_panic (self=0x1) at rust_panic.rs:59
#4  rust_panic::HelloPanic::step_three (self=0x1) at rust_panic.rs:53
#5  rust_panic::area_in_hp (rectangle=0xffffd08c4f063c38) at rust_panic.rs:24
#6  0xffffffffc07fe160 in rust_panic::HelloPanic::step_two (self=<optimized out>) at rust_panic.rs:46
#7  0xffffffffc0c5d067 in ?? ()
crash> frame 5
#5  rust_panic::area_in_hp (rectangle=0xffffd08c4f063c38) at rust_panic.rs:24
24          HelloPanic.step_three();
crash> whatis rectangle
*mut rust_panic::RectangleHP
crash> p *rectangle
$3 = rust_panic::RectangleHP {
  width: 30,
  height: 50
}
crash> 
```

Rust support in crash-utility

– Implementation details

- Demangle Rust symbol names in all outputs

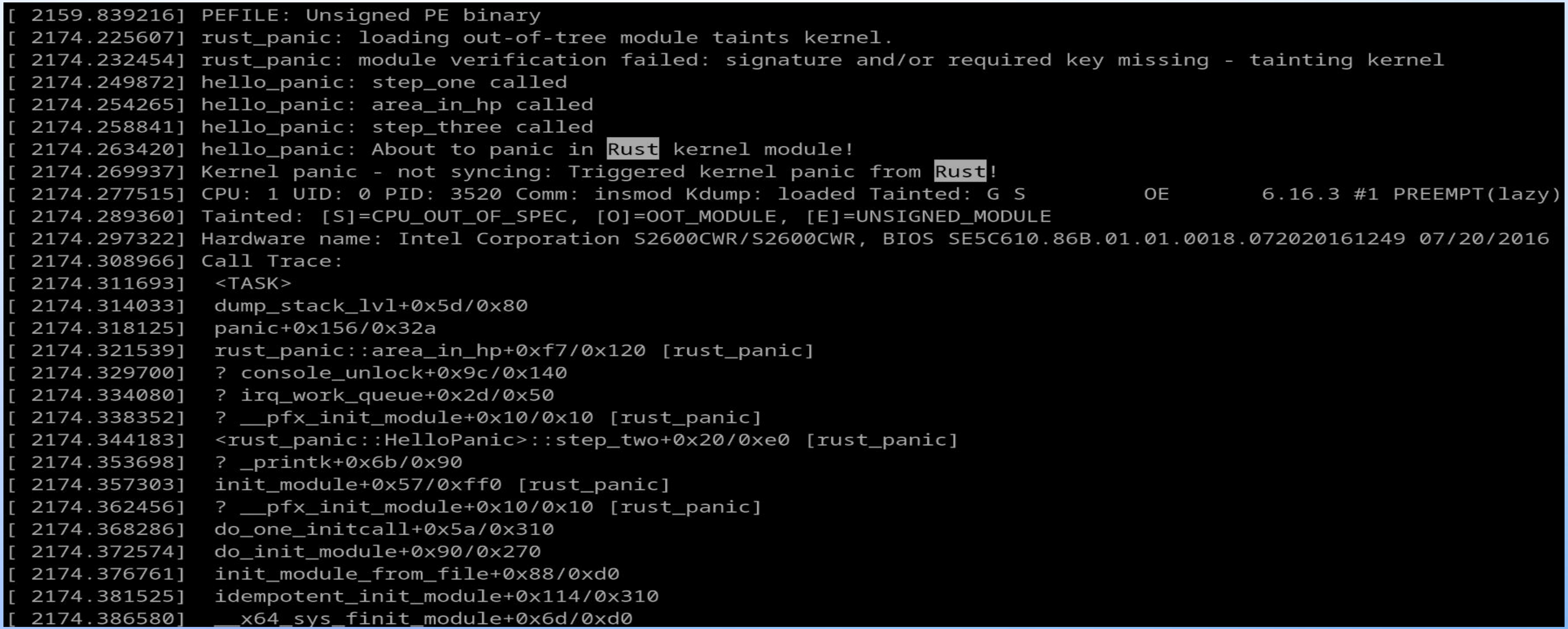


```
crash> sym "rust_panic::area_in_hp"
ffffffffc07fe010 (t) rust_panic::area_in_hp [rust_panic] /root/linux-6.16.3/samples/rust/rust_panic.rs: 22
crash> sym ffffffffc07fe010
ffffffffc07fe010 (t) rust_panic::area_in_hp [rust_panic] /root/linux-6.16.3/samples/rust/rust_panic.rs: 22
crash> dis "rust_panic::area_in_hp"
0xffffffffc07fe010 <rust_panic::area_in_hp>::area_in_hp>:       push   %rbx
0xffffffffc07fe011 <rust_panic::area_in_hp+1>::area_in_hp+1>:   sub    $0x90,%rsp
0xffffffffc07fe018 <rust_panic::area_in_hp+8>::area_in_hp+8>:   mov    %rdi,%rbx
0xffffffffc07fe01b <rust_panic::area_in_hp+11>::area_in_hp+11>: movq   $0xffffffffc0bd5020,(%rsp)
0xffffffffc07fe023 <rust_panic::area_in_hp+19>::area_in_hp+19>: movq   $0x1,0x8(%rsp)
0xffffffffc07fe02c <rust_panic::area_in_hp+28>::area_in_hp+28>: movq   $0x0,0x20(%rsp)
0xffffffffc07fe035 <rust_panic::area_in_hp+37>::area_in_hp+37>: movq   $0x8,0x10(%rsp)
0xffffffffc07fe03e <rust_panic::area_in_hp+46>::area_in_hp+46>: movq   $0x0,0x18(%rsp)
0xffffffffc07fe047 <rust_panic::area_in_hp+55>::area_in_hp+55>: mov    %rsp,%rcx
0xffffffffc07fe04a <rust_panic::area_in_hp+58>::area_in_hp+58>: mov    $0xc,%edx
0xffffffffc07fe04f <rust_panic::area_in_hp+63>::area_in_hp+63>: mov    $0xffffffff96afb62c,%rdi
0xffffffffc07fe056 <rust_panic::area_in_hp+70>::area_in_hp+70>: mov    $0xffffffffc0bd5118,%rsi
0xffffffffc07fe05d <rust_panic::area_in_hp+77>::area_in_hp+77>: call   0xffffffff95fcb2e0 <kernel::print::call_printk>
0xffffffffc07fe062 <rust_panic::area_in_hp+82>::area_in_hp+82>: movq   $0xffffffffc0bd50d0,0x30(%rsp)
0xffffffffc07fe06b <rust_panic::area_in_hp+91>::area_in_hp+91>: movq   $0x1,0x38(%rsp)
0xffffffffc07fe074 <rust_panic::area_in_hp+100>::area_in_hp+100>:       movq   $0x0,0x50(%rsp)
0xffffffffc07fe07d <rust_panic::area_in_hp+109>::area_in_hp+109>:       movq   $0x8,0x40(%rsp)
0xffffffffc07fe086 <rust_panic::area_in_hp+118>::area_in_hp+118>:       movq   $0x0,0x48(%rsp)
0xffffffffc07fe08f <rust_panic::area_in_hp+127>::area_in_hp+127>:       lea    0x30(%rsp),%rcx
0xffffffffc07fe094 <rust_panic::area_in_hp+132>::area_in_hp+132>:       mov    $0xc,%edx
0xffffffffc07fe099 <rust_panic::area_in_hp+137>::area_in_hp+137>:       mov    $0xffffffff96afb62c,%rdi
0xffffffffc07fe0a0 <rust_panic::area_in_hp+144>::area_in_hp+144>:       mov    $0xffffffffc0bd5118,%rsi
0xffffffffc07fe0a7 <rust_panic::area_in_hp+151>::area_in_hp+151>:       call   0xffffffff95fcb2e0 <kernel::print::call_printk>
0xffffffffc07fe0ac <rust_panic::area_in_hp+156>::area_in_hp+156>:       movq   $0xffffffffc0bd5108,0x60(%rsp)
```

Rust support in crash-utility

– Implementation details

● Demangle Rust symbol names in all outputs



```
[ 2159.839216] PEFILE: Unsigned PE binary
[ 2174.225607] rust_panic: loading out-of-tree module taints kernel.
[ 2174.232454] rust_panic: module verification failed: signature and/or required key missing - tainting kernel
[ 2174.249872] hello_panic: step_one called
[ 2174.254265] hello_panic: area_in_hp called
[ 2174.258841] hello_panic: step_three called
[ 2174.263420] hello_panic: About to panic in Rust kernel module!
[ 2174.269937] Kernel panic - not syncing: Triggered kernel panic from Rust!
[ 2174.277515] CPU: 1 UID: 0 PID: 3520 Comm: insmod Kdump: loaded Tainted: G S        OE      6.16.3 #1 PREEMPT(lazy)
[ 2174.289360] Tainted: [S]=CPU_OUT_OF_SPEC, [O]=OOT_MODULE, [E]=UNSIGNED_MODULE
[ 2174.297322] Hardware name: Intel Corporation S2600CWR/S2600CWR, BIOS SE5C610.86B.01.01.0018.072020161249 07/20/2016
[ 2174.308966] Call Trace:
[ 2174.311693]  <TASK>
[ 2174.314033]  dump_stack_lvl+0x5d/0x80
[ 2174.318125]  panic+0x156/0x32a
[ 2174.321539]  rust_panic::area_in_hp+0xf7/0x120 [rust_panic]
[ 2174.329700]  ? console_unlock+0x9c/0x140
[ 2174.334080]  ? irq_work_queue+0x2d/0x50
[ 2174.338352]  ? __pfx_init_module+0x10/0x10 [rust_panic]
[ 2174.344183]  <rust_panic::HelloPanic>::step_two+0x20/0xe0 [rust_panic]
[ 2174.353698]  ? _printk+0x6b/0x90
[ 2174.357303]  init_module+0x57/0xff0 [rust_panic]
[ 2174.362456]  ? __pfx_init_module+0x10/0x10 [rust_panic]
[ 2174.368286]  do_one_initcall+0x5a/0x310
[ 2174.372574]  do_init_module+0x90/0x270
[ 2174.376761]  init_module_from_file+0x88/0xd0
[ 2174.381525]  idempotent_init_module+0x114/0x310
[ 2174.386580]  __x64_sys_finit_module+0x6d/0xd0
```

Rust support in crash-utility

– challenges for Rust support

- Rust mangling changed(v0->v1),crash needs to support both demangling schemes.
- Some Rust expression does not implement in gdb
- No stable ABI for Rust-level types across compiler versions; layouts may change unless repr(C) is used.
- …

Q&A

- Upstream
  - devel@lists.crash-utility.osci.io
  - lijiang@redhat.com
  - ltao@redhat.com
- git repo:
  - https://github.com/crash-utility/crash
- Test case & Demo:
  - Please let me know if you would like to try

THANKS