DRM Panic: The BSOD is coming to Linux

Jocelyn Falempe
XDC 2025

Red Hat

DRM Panic History

- First RFC: Aug 2023
- v6.10: simpledrm, mgag200, ast, imx/ipuv3
- v6.11: tidss
- v6.13: nouveau, rcar-du, shmobile
- v6.14: amdgpu
- v6.15: virtio-gpu
- v6.17: i915, xe, bochs, hyperv

Choosing the framebuffer

Allocate a panic framebuffer at boot and use it when a panic occurs.

- Can use linear and supported format.
- May not fit the current monitor, if it has changed since boot.
- Need to reconfigure the display pipeline. Might get complex with DP hub.

Re-use the current Framebuffer.

- No need to change the monitor mode
- Need to support the current framebuffer format (tiling, colors, ...)
- Framebuffer might be in device memory, not accessible from CPU

Framebuffer access

DRM Panic provides 3 ways to draw to the framebuffer

- iosys_map (Linear, contiguous in virtual memory or io)
- Page list
- set_pixel() callback

Writing your panic handler

To add support to your GPU driver, you need to implement this callbacks for your primary plane:

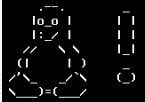
- get_scanout_buffer()
- set_pixel() (optional)
- panic_flush() (optional)

Panic screen

There are currently 3 panic screens. You can choose in the kernel configuration, or at run time with:

```
$ echo -n qr_code | sudo tee /sys/module/drm/parameters/panic_screen
```

- User (default)
- kmsg
- QR code

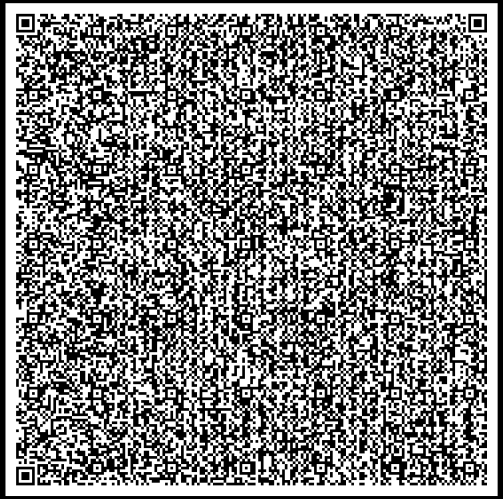


KERNEL PANIC!

Please reboot your computer.

sysrq triggered crash

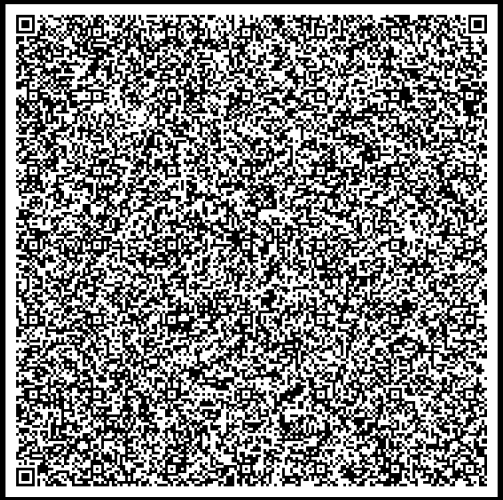
```
[    5.815581] rfkill: input handler disabled
[    7.004170] input: spice vdagent tablet as /devices/virtual/input/input7
[    9.516069] systemd-journald[480]: Time jumped backwards, rotating.
[   22.726130] sysrq: Emergency Sync
[   22.726160] sysrq: Emergency Remount R/O
[   22.726179] sysrq: Trigger a crash
[   22.726182] Kernel panic - not syncing: sysrq triggered crash
[   22.726189] CPU: 2 UID: 0 PID: 2171 Comm: panic Not tainted 6.17.0-rc2+ #343 PREEMPT(voluntary)
[   22.726192] Hardware name: QEMU Standard PC (Q35 + ICH9, 2009), BIOS edk2-20250523-16.fc42 05/23/2025
[   22.726195] Call Trace:
[   22.726197]  <TASK>
[   22.726200]  dump_stack_lvl+0x2d/0x90
[   22.726206]  vpanic+0xd3/0x320
[   22.726209]  panic+0x4f/0x50
[   22.726211]  sysrq_handle_crash+0x1a/0x20
[   22.726213]  __handle_sysrq+0x14b/0x160
[   22.726215]  write_sysrq_trigger+0x4c/0x80
[   22.726217]  proc_reg_write+0x65/0xb0
[   22.726220]  vfs_write+0x115/0x3f0
[   22.726224]  ? __se_sys_close+0x7b/0xe0
[   22.726226]  ? do_syscall_64+0xbb/0x220
[   22.726229]  ? get_close_on_exec+0x37/0x40
[   22.726230]  ksys_write+0x75/0xe0
[   22.726233]  do_syscall_64+0x8a/0x220
[   22.726236]  ? __se_sys_fcntl+0x67/0xb0
[   22.726238]  ? do_syscall_64+0xbb/0x220
[   22.726241]  ? task_work_run+0x89/0xb0
[   22.726244]  ? kmem_cache_free+0x199/0x310
[   22.726246]  ? __rseq_handle_notify_resume+0x290/0x5a0
[   22.726249]  ? get_close_on_exec+0x37/0x40
[   22.726250]  ? do_fcntl+0x81/0x8d0
[   22.726252]  ? __se_sys_fcntl+0x67/0xb0
[   22.726254]  ? do_syscall_64+0xbb/0x220
[   22.726256]  ? do_syscall_64+0xbb/0x220
[   22.726258]  ? clear_bhb_loop+0x70/0xc0
[   22.726260]  ? clear_bhb_loop+0x70/0xc0
[   22.726261]  entry_SYSCALL_64_after_hwframe+0x76/0x7e
[   22.726263] RIP: 0033:0x7fa777b6a304
[   22.726266] Code: c7 00 16 00 00 00 b8 ff ff ff ff c3 66 2e 0f 1f 84 00 00 00 00 00 f3 0f 1e fa 80 3d a5 0d 0e 00 00 74 13 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77
54 c3 0f 1f 00 55 48 89 e5 48 83 ec 20 48 89
[   22.726267] RSP: 002b:00007ffd0139afc8 EFLAGS: 00000202 ORIG_RAX: 0000000000000001
[   22.726269] RAX: ffffffffffffffda RBX: 0000000000000002 RCX: 00007fa777b6a304
[   22.726271] RDX: 0000000000000002 RSI: 00005587aa0c1b70 RDI: 0000000000000001
[   22.726272] RBP: 00007ffd0139aff0 R08: 0000000000000073 R09: 00000000ffffffff
[   22.726273] R10: 0000000000000000 R11: 0000000000000202 R12: 0000000000000002
[   22.726274] R13: 00005587aa0c1b70 R14: 00007fa777c445c0 R15: 00007fa777c41f00
[   22.726276]  </TASK>
[   22.726402] Kernel Offset: 0x25000000 from 0xffffffff81000000 (relocation range: 0xffffffff80000000-0xffffffffbfffffff)
```



KERNEL PANIC!

Please reboot your computer.

sysrq triggered crash



KERNEL PANIC!

Please reboot your computer.

sysrq triggered crash

QR Code

- Most QR code reader only support URL
- If URL is unset, data is stored as plain text (3K for V40)
- If URL is set, data is compressed and encoded (7K for V40)
- Panic data as url fragment, so stay on the client side.
  https://kdj0c.github.io/panic_report#
- Offline decoding with an Android App
  https://github.com/adryzz/drm_panic-viewer

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


facebook.com/redhatinc


youtube.com/user/RedHatVideos


twitter.com/RedHat