Project Report: Group 6

QEMU Issue - Medium

Link to issue: https://gitlab.com/qemu-project/qemu/-/issues/413

Link to forked repository: https://gitlab.com/eylamtagor1/qemu
QEMU is an open-source machine emulator and virtualizer. It is capable of emulating various architectures, including x86, ARM, MIPS, and PowerPC, and can run operating systems and applications designed for those architectures.

Issue description:

This issue revolved around solidifying checks of the return values of four methods (load_image_targphys, get_image_size, event_notifier_init, and msix_init) in order to fail gracefully upon encountering an error.

How we solved it:

We looked through every single call to these four methods and we noted those that seemed to be improperly checking for errors. After comprising a list of those calls, we made the necessary fix to each call one by one in order to ensure it fails gracefully. Here is a comprehensive list of every method call and the fixes we implemented:

get_image_size:

- hw/i386/x86.c:1145
  - Call seems more complex than within our scope, error checking would not be relevant and will likely cause more harm than good.
- hw/mips/fuloong2e.c:121
  - Appears to be checking return == -1. Fixed to check for all error values.
- hw/mips/loongson3_virt.c:353
  - Appears to be checking return == -1. Fixed to check for all error values.
- hw/mips/malta.c:913
  - Appears to be checking return == -1. Fixed to check for all error values.
- hw/mips/mipssim.c:89
  - Appears to be checking return == -1. Fixed to check for all error values.
- hw/pci-host/raven.c:350
  - Call seems more complex than within our scope, error checking would not be relevant and will likely cause more harm than good.
- hw/smbios/smbios.c:1255

  - Checks return == -1 and something else. Fixed the first check while retaining the other checks.
- hw/sparc/leon3.c:296
  - Guarded with a call to qemu_find_file, but doesn't actually check for an error from get_image_size. We found that this is sufficient since any failure of get_image_size would be due to a failure of the earlier call to qemu_find_file, which does properly check for and handle errors.

msix_init:

- hw/net/igbvf.c:264
  - Checking if(ret). Fixed to check for all error values.
- hw/net/vmxnet3.c:2124
  - This one is technically fine, but goes against QEMU style guide. Fixed to match QEMU style[1].
- hw/pci/msix.c:416
  - Checking if(ret). May be correct, but could also be against QEMU style guide. Fixed to match QEMU style.
- hw/scsi/megasas.c:2396
  - No check on return value at all; this one was complicated. We added a check for the return value and for an enum called ON_OFF_AUTO following a relevant comment's instructions.
- hw/usb/hcd-xhci-pci.c:159
  - No check on return value at all; this one was complicated. We added a check for the return value and for an enum called ON_OFF_AUTO following a relevant comment's instructions.

event_notifier_init:

- block/nvme.c:751
  - Checking if(ret). Fixed to check for all error values.
- hw/hyperv/hyperv.c:410
  - Checking if(ret). Fixed to check for all error values.
- hw/hyperv/hyperv.c:424
  - Checking if(ret). Fixed to check for all error values.
- hw/hyperv/vmbus.c:1426
  - Checking if(ret). Fixed to check for all error values.
- hw/hyperv/vmbus.c:2421
  - Checking ret != 0. Fixed to check for all error values.
- hw/remote/proxy.c:57,58
  - Not checking return. Fixed to check for all error values.
- hw/vfio/ccw.c:426
  - Checking if(ret). Fixed to check for all error values.

[1]https://www.qemu.org/docs/master/devel/style.html#:~:text=Rationale%3A%20Yoda%20conditions%20(as%20in%20%E2%

- hw/vfio/pci-quirks.c:361
  - Checking if(ret). Fixed to check for all error values.
- hw/vfio/pci.c:135,288,433,481,684,2743
  - Checking if(ret). Fixed to check for all error values.
- hw/vfio/platform.c:75,86
  - Checking if(ret). Fixed to check for all error values.
- hw/virtio/vhost-vdpa.c:863,869
  - Checking if(ret). Fixed to check for all error values.

load_image_targphys:

(Note: This function calls get_image_size, but also calls another function which seems to have more complicated reasons for errors)

- hw/alpha/dp264.c:194
  - Does not check return value for error.
  - Add check and handled like the above check, should suffice because the result is necessary to continue.
- hw/hppa/machine.c:382
  - No check.
  - Add check and handled like the above check, should suffice because the result is necessary to continue.
- hw/m68k/q800.c:672
  - No check.
  - Add check and handled like the above check, should suffice because the result is necessary to continue.
- hw/m68k/q800.c:699
  - Has a proper check on line 707 upon further inspection. No fix necessary.
- hw/m68k/virt.c:283
  - No check.
  - Add check and handled like the above check, should suffice because the result is necessary to continue.
- hw/mips/boston.c:780
  - Checks result == -1.
  - Changed to < 0 to check for all error values.
- hw/mips/fuloong2e.c:129
  - Checks return correctly upon further inspection.
- hw/mips/loongson3_virt.c:364
  - Checks return correctly upon further inspection.
- hw/mips/malta.c:928
  - Checks return correctly upon further inspection.
- hw/mips/mipssim.c:97
  - Checks return correctly upon further inspection.
- hw/riscv/boot.c:201
  - Checks return == -1.

  - Changed to < 0 to check for all error values.
- hw/s390x/ipl.c:165
  - Checks return == -1.
  - Changed to < 0 to check for all error values.
- hw/s390x/ipl.c:243
  - Checks return == -1.
  - Changed to < 0 to check fo all error values.

Challenges:

We decided to leave the following calls to load_image_targphys alone. These were in more complicated and critical sections of the codebase, so we thought it best to leave these as they were without having any additional insight into why the code was written this way in the first place:

- hw/microblaze/boot.c:175
  - No check, but the context of the call suggests that this is due to a far more complex reason that is within the scope of our task.
- hw/nios2/boot.c:191
  - No check, but the context of the call suggests that this is due to a far more complex reason that is within the scope of our task.
- hw/ppc/virtex_ml507.c:274
  - No check, but the context of the call suggests that this is due to a far more complex reason that is within the scope of our task.
- hw/m68k/mcf5208.c:313
  - Checks return value for < 8. The context of the call suggests that this is due to a far more complex reason that is within the scope of our task.
- hw/m68k/next-cube.c:1004
  - Checks return value for < 8. The context of the call suggests that this is due to a far more complex reason that is within the scope of our task.

Additionally, finding a way to contribute our changes was also more difficult than with the other repositories we worked on. We reached out to some of the developers on the email newsletter to see how we could go about merging our changes, but didn't get any response.

Firecracker Issue - Easy

Link to issue: https://github.com/firecracker-microvm/firecracker/issues/3273

Link to forked repository: https://github.com/StonewallJohnson/firecracker

Link to PR: https://github.com/firecracker-microvm/firecracker/pull/3636
Firecracker is an open-source project that provides lightweight virtualization technology for running containers and functions in a secure and isolated manner. Developed by Amazon Web Services (AWS), Firecracker uses a microVM-based architecture to provide a secure and efficient environment for running isolated workloads. The repository includes the source code for the Firecracker hypervisor, as well as tools for building and managing Firecracker-based environments.

Issue Description:

This issue is a culmination of TODO comments that are untracked, meaning they are not assigned to their own issue. Instead of adding an issue for each one, our task was to implement a reasonable fix to these TODOs and simply submit a pull request with the fixes.

How we solved it:

- src/logger/src/metrics.rs:95
  - Changed app_metrics to metrics_buf_input.
- src/dumbo/src/tcp/endpoint.rs:88
  - Added documentation comment specifying the intent behind the assert statement.

Challenges:

We listed these as potential TODO comments to complete, however, we realized that some of those were too difficult to accomplish without extensive knowledge and experience working on this project, which we are confident goes beyond the intent of this issue.

- src/devices/src/virtio/queue.rs:95
  - The TODO was stolen by some other contributor: https://github.com/firecracker-microvm/firecracker/pull/3592
- src/vmm/src/vstate/vm.rs:385, 387
  - Asks to rename this field to adopt inclusive language once Linux updates it.
  - Don't think Linux has changed the terminology.
- src/dumbo/src/pdu/mod.rs:80
  - We tried but couldn't actually verify that the checksum wouldn't overflow.
- src/dumbo/src/pdu/tcp.rs:340

OSv Issue - Medium

Link to issue: https://github.com/cloudius-systems/osv/issues/1025

Link to forked repository: https://github.com/StonewallJohnson/osv

Link to PR: https://github.com/cloudius-systems/osv/pull/1232
OSv is an open-source operating system designed for use in unikernels deployed to the cloud. It is built from the ground up to run on virtualized environments and offers a lightweight, efficient, and high-performance alternative to traditional operating systems. OSv's design is based on the principle of running a single application per virtual machine, enabling better resource utilization and improved security.

Issue description:

This issue stemmed from a user testing python3 modules and running into an error due to undefined semaphore functions. Some developers discussed potential approaches to solving this issue and reached the conclusion that named semaphores must be implemented.

How we solved it:

We first needed to find a reasonable approach and interface for implementing. Since the current unnamed semaphore implementation was already POSIX [2] (a popular standard for operating systems by IEEE), we decided to implement the corresponding POSIX named semaphores. Since this is a unikernel where only one process ever exists, some portions of the POSIX standard were not necessary to implement, like making the a corresponding file for the named semaphore. This was also corroborated by the discussion of contributors on the issue.

Following the discussion of the contributors we implemented an unordered map that stored the mappings from names to open semaphores. This required some modification and addition to the `indirect_semaphore` object, which would now need to store the number of references and link status of the semaphore.

Implementing the POSIX methods relevant to named semaphores (sem_open, sem_unlink, sem_close) was then relatively straight forward and we made sure to adhere to the description of these methods in the POSIX standard[3] where necessary.

Challenges:

The main challenge we encountered while working on this issue is understanding the POSIX standard prior to actual implementation. The provided documentation consisted of written descriptions of how a named semaphore should be implemented in a way that matches POSIX standards, which left a lot of room

[2]https://www.opengroup.org/posix-systems
[3]https://pubs.opengroup.org/onlinepubs/9699919799/

for vagueness and uncertainty. Therefore, it took more time and effort than we expected to be able to write the actual code.