Zyad A. Ali - **Student**

Rahat Mahmood - **Mentor**

Message Queues in gVisor - GSoC

Overview and Work Product

Project Description

gVisor is a secure application kernel for containers, written in Go. It implements a substantial portion of the Linux system surface and is constantly growing.

Linux provides two alternate message queue mechanisms for inter-process communication: SysV, and POSIX message queues, none of which was previously implemented in gVisor. Both mechanisms use multiple syscalls to send, receive, and manage messages.

The initial goal of the project was to implement a minimal version of the SysV variant of message queues in gVisor.

Overview of Work Done

The initial goal was achieved in week 5, and the project was extended to include both SysV and POSIX flavors. The extended goal was achieved in week 10.

As of now, the work done is

- Implementation of SysV message queues: merged.
- Implementation of POSIX message queues: to be merged / under review.

Design and Implementation

SysV Message Queues

SysV message queues add 4 new syscalls

- **`msgget`**: creates a new message queue or obtains the ID of an existing one.
- **`msgsnd`**: adds a message to the queue.
- **`msgrcv`**: receives a message from the queue.
- **`msgctl`**: performs various control operations, including deletion.

Work on SysV queues includes

- Design and implementation of package **`kernel/msgqueue`**, which serves as the main project package and contains queue logic and main structures.
- Refactoring of existing SysV IPC mechanisms into a uniform implementation residing in package **`kernel/ipc`**.
- Implementation of a test-suite to cover usage cases mentioned in the Linux manual.

POSIX Message Queues

Posix message queues introduce 6 new syscalls

- **`mq_open`**: creates a new message queue (file description) or obtains the file descriptor of an existing one.
- **`mq_unlink`**: marks an existing message queue for deletion, and deletes it after all users close it.
- **`mq_timedsend`**: sends a message through the queue.
- **`mq_timedreceive`**: receives a message from the queue.
- **`mq_getsetattr`**: retrieves or updates queue attributes.
- **`mq_notify`**: registers a process for async notification regarding the arrival of a message

Work on POSIX queues includes

- Design and implementation of **`mqfs`**, a pseudo filesystem to back message queues, and provide several control operations.
- Design and implementation of package **`kernel/mq`**, which serves as a container for most of queue operations' logic.
- Implementation of a test suite to cover both POSIX specification, and Linux-specific usage cases.

Pull Requests

The project is collected in these PRs

- gVisor#6108: Implement msgget(2) and msgctl(IPC_RMID).

- gVisor#6171: Implement msgsnd(2) and msgrcv(2).
- gVisor#6262: Implement msgctl(2).
- gVisor#6345: Implement mq_open(2) and mq_unlink(2).
- gVisor#6429: Implement mq_timedsend(2) and mq_timedreceive(2).
- gVisor#6457: Implement mq_getsetattr(2).
- gVisor#6458: Implement mq_notify(2).