Run-time verification for real-time application

Linux Plumbers Conference 2025

Nam Cao
Linutronix GmbH

Motivation

- Kernel is real-time capable
- Userspace often is not:
  - Mutex without PI
  - Not mlock memory → page faults
  - Use the RT-unsafe API (e.g. timerfd)

Run-time verification monitors can help detect common userspace mistakes

Run-time verification for real-time

The run-time monitors use trace points and verify that:

- Real-time tasks do not raise page faults
- RT tasks are not woken by lower-priority tasks or softirq
- A task going to sleep must mean:
  - clock_nanosleep (abstime, monotonic or TAI clock)
  - futex_wait
  - epoll_wait

Run-time verification for real-time

The run-time monitors use trace points and verify that:

- Real-time tasks do not raise page faults
- RT tasks are not woken by lower-priority tasks or softirq
- A task going to sleep must mean:
  - clock_nanosleep (abstime, monotonic or TAI clock)
  - futex_wait
  - epoll_wait

Are the above rules complete?

Run-time verification for real-time

The run-time monitors use trace points and verify that:

- Real-time tasks do not raise page faults
- RT tasks are not woken by lower-priority tasks or softirq
- A task going to sleep must mean:
  - clock_nanosleep (abstime, monotonic or TAI clock)
  - futex_wait ← **should we? Maybe we should change to futex_pi?**
  - epoll_wait

Are the above rules complete?

futex_wait

Should RV monitors warn users about **futex_wait**?

- Glibc’s **pthread_mutex()** without PI uses **futex_wait** → we should warn that
- Glibc’s **pthread_cond_wait()** also uses **futex_wait** → maybe we shouldn’t warn
- What about other users of **futex_wait**?

Run-time verification for real-time

The run-time monitors use trace points and verify that:

- Real-time tasks do not raise page faults
- RT tasks are not woken by lower-priority tasks or softirq
- A task going to sleep must mean:
  - clock_nanosleep (abstime, monotonic or TAI clock)
  - futex_wait
  - epoll_wait
  - **... something else? What about blocking read() and write()?**

Thank you