futex2: next steps

LPC 2021

André Almeida

Kernel Developer

andrealmeid@collabora.com

Why do we need futex2?

- Current interface will not get new features

- Futex2 interface should solve current limitations:
  - NUMA awareness operations
  - Support for various sizes (8, 16, 32, 64) bits
  - Wait on multiple futexes

Implementing futex2

- Refactor futex.c in smaller files
  - Thanks Peter!
- Reuses most of code
- No multiplexing, one syscall per operation
- Merging smaller patches

The interface: Wait on multiple

```
futex_waitv(struct futex_waitv *waiters, unsigned int nr_futexes,
            unsigned int flags, struct timespec *timo)

struct futex_waitv {
    __u64 val;
    __u64 uaddr;
    __u32 flags;
    __u32 __reserved;
};
```

The interface: Wait on multiple

```
futex_waitv(struct futex_waitv *waiters, unsigned int nr_futexes,
            unsigned int flags, ~~struct timespec *timo~~)
                                                   → __u64 time

struct futex_waitv {
    __u64 val;
    __u64 uaddr;
    __u32 flags;
    __u32 __reserved;
};
```

The interface: Wait on multiple

```
futex_waitv(struct futex_waitv *waiters, unsigned int nr_futexes,
            unsigned int flags, struct timespec *timo)
```

~~struct futex_waitv {~~
~~   __u64 val;~~
~~   __u64 uaddr;~~
~~   __u32 flags;~~
~~   __u32 __reserved;~~
~~};~~

```
struct futex_waitv {
    __u64 val;
    *void uaddr;
    __u32 flags;
};
```

The interface: Wait and wake

```
futex_wait(void *uaddr, unsigned int val, unsigned int flags,
           struct timespec *timo)

futex_wake(void *uaddr, unsigned long nr_wake, unsigned int flags)
```

The interface: Wait and wake

```
futex_requeue(struct futex_requeue *rq1, struct futex_requeue *rq2,
              unsigned int nr_wake, unsigned int nr_requeue,
              u64 cmpval, unsigned int flags)

struct futex_requeue {
     __u64 uaddr;
     __u32 flags;
     __u32 __reserved;
};
```

The interface: Flags

```
Sizes: FUTEX_8, FUTEX_16, FUTEX_32, FUTEX_64

Private: FUTEX_PRIVATE_FLAG

Clock spec: FUTEX_REALTIME_CLOCK
```

The interface: NUMA

Flag: FUTEX_NUMA_FLAG

```
void *uaddr:
struct futex32_numa {
    __u32 value;
    __s32 hint;
};
```

value → expected value

hint → [0, MAX_NUMA_NODE] for NUMA to operate, -1 to current node

Thank you

```
Message {
  config {
    priority: "high"
    body: "Collabora is hiring"  // Many open positions
    recipient: "you"             // Please join us
    calltoaction: "http://col.la/join"
  }
}
```

futex2: next steps

Backup slides

NUMA awareness

- Futex has a single global hash table
- Hurts performance for all nodes that doesn’t have the table

Variable size

- Futex can only use 32-bit integers
- Almost all uses cases are related to atomic operations
  - Userspace atomic primitives implementation
- 64-bit can be also useful to wait in a pointer value

Wait on multiple

- Wait for multiple resources is a common pattern in games

- In my use case, using futex_waitv instead of eventfd() can decrease CPU usage and enhance game performance