

MO806/MC914
Tópicos em Sistemas Operacionais
2s2008

Futex - Fast User-Mode Mutex

Prototype

```
long sys_futex (  
    void *addr1,  
    int op,  
    int val1,  
    struct timespec *timeout,  
    void *addr2,  
    int val3);  
  
int syscall(SYS_futex, addr1, FUTEX_XXXX,  
            val1, timeout, addr2, val3);
```

Operações

- FUTEX_WAIT
- FUTEX_WAKE
- FUTEX_WAKE_OP
- FUTEX_CMP_REQUEUE
- FUTEX_REQUEUE **obsoleta**
- FUTEX_FD

FUTEX_WAIT

- Bloqueio até notificação
- Não há bloqueio se `*addr1 != val1` (EWOULDBLOCK)
- Retorna ao receber um sinal (EINT)
- Pode-se estabelecer um tempo máximo (ETIMEDOUT)
- Veja o código `ex0.c`

FUTEX_WAKE

- `val1` indica o número de threads a serem acordadas
 - 1
 - `INT_MAX`
- Retorna o número de threads acordadas
- Veja os códigos `ex1.c` e `ex2.c`

Sinalizador de eventos

```
class event {
public:
    event () : val (0) { }
    void ev_signal () {
        ++val;
        futex_wake (&val, INT_MAX); }
    void ev_wait () {
        futex_wait (&val, val); }
private
    int val;
};
```

Mutex: primeira proposta

```
class mutex {  
    public:  
        mutex () : val (0) { }  
        void lock () {  
            int c;  
            while ((c = atomic_inc (val)) != 0)  
                futex_wait (&val, c + 1); }  
        void unlock () {  
            val = 0; futex_wake (&val, 1); }  
    private:  
        int val;  
};
```

Mutex: primeira proposta

- `atomic_inc` como descrito no artigo:
 - Incrementa atomicamente `val`
 - Retorna valor anterior
- `atomic_inc_and_test` como disponível:
 - Incrementa atomicamente `val`
 - Retorna `true` se `val` era 0 e `false` caso contrário

Mutex: primeira proposta

- Garante exclusão mútua
- Se a fila não estiver vazia, uma thread irá conseguir pegar o lock após um unlock.
- Se não há espera, a última chamada de sistema é desnecessária
- Livelock
- Overflow (2^{32})

Mutex: segunda proposta

- Significado para val
 - 0: unlocked
 - 1: locked, sem espera
 - 2: locked, com espera
- `cmpxchg(var, old, new)`
 - $\text{var} \leftarrow \text{new}$ se $\text{var} == \text{old}$
 - retorna valor de var antes da operação

Mutex: segunda proposta

```
class mutex {  
    public:  
        mutex () : val (0) { }  
        void lock () {  
            int c;  
            if ((c = cmpxchg (val, 0, 1)) != 0)  
                do {  
                    if (c == 2 || cmpxchg (val, 1, 2) != 0)  
                        futex_wait (&val, 2);  
                } while ((c = cmpxchg (val, 0, 2)) != 0);  
        }  
}
```

Mutex: segunda proposta

```
void unlock () {  
    if (atomic_dec (val) != 1) {  
        val = 0;  
        futex_wake (&val, 1);  
    }  
}  
  
private:  
    int val;  
};
```