

# Estudo da implementação de Mutexes na glibc

Bruno Azevedo e Wilson Higashino

Instituto de Computação / Universidade Estadual de Campinas (Unicamp)



5 de novembro de 2009

# Sumário

- Tipos de mutexes
- Atributos de mutexes
- Configurações
- Estudo do código

# Introdução

- GNU C Library 2.9
- A glibc estende o padrão POSIX.

# Fast

- Mais eficientes
- Sem verificação de erros ou consistência

Exemplo:

```
pthread_mutex_t mutex;  
pthread_mutex_init(&mutex, NULL);  
pthread_mutex_lock(&mutex);  
/* Região crítica */  
pthread_mutex_unlock(&mutex);
```

## Error-checking

- Três tipos de erros:
  - Uma thread só deve executar unlock em um mutex previamente locked. Retorna EPERM.
  - Uma thread só deve executar unlock em que ela mesma executou o lock. Retorna EPERM.
  - Uma thread não deve executar outro lock em um mutex que ela já tenha obtido o lock. Retorna EDEADLK.

Exemplo:

```
pthread_mutex_t mutex;  
pthread_mutexattr_t attr;  
pthread_mutexattr_init (&attr);  
pthread_mutexattr_settype (&attr,  
    PTHREAD_MUTEX_ERRORCHECK_NP);  
pthread_mutex_init (&mutex, &attr);
```

# Recursive

- Semelhante ao mutex error-checking.
- Pode ser obtido múltiplas vezes pelo dono.

Exemplo:

```
pthread_mutex_t mutex;  
pthread_mutexattr_t attr;  
pthread_mutexattr_init (&attr);  
pthread_mutexattr_settype (&attr,  
    PTHREAD_MUTEX_RECURSIVE_NP);  
pthread_mutex_init (&mutex, &attr);
```

# Adaptive

- Para sistemas multiprocessados.
- Se adaptam, Spin vs Block.

Exemplo:

```
pthread_mutex_t mutex;  
pthread_mutexattr_t attr;  
pthread_mutexattr_init (&attr);  
pthread_mutexattr_settype (&attr,  
    PTHREAD_MUTEX_ADAPTIVE_NP);  
pthread_mutex_init (&mutex, &attr);
```

# Robust

- Trata a situação de uma thread/processo terminar abruptamente, sem antes liberar o lock.
- Uma thread bloqueada recebe o mutex e retorna EOWNERDEAD. Esta deve garantir consistência.
- Após correção deve invocar pthread\_mutex\_consistent\_np.

Exemplo:

```
pthread_mutex_t lock;  
pthread_mutexattr_t attr;  
pthread_mutexattr_init(&attr);  
pthread_mutexattr_setrobust_np(&attr,  
    PTHREAD_MUTEX_ROBUST_NP);  
pthread_mutex_init(&lock, &attr);
```



## Protocolos de prioridade

- Buscam evitar a inversão de prioridade.
- Priority Protect e Priority Inheritance.

Exemplo:

```
pthread_mutexattr_t attr;  
pthread_mutexattr_init(&attr);  
pthread_mutexattr_setprotocol(attr, PTHREAD_PRIO_INHERIT);  
/* Ou pthread_mutexattr_setprotocol(attr,  
   PTHREAD_PRIO_PROTECT);  
pthread_mutexattr_setprioceiling(&attr, 10); */  
pthread_mutex_init(&lock, &attr);
```

# Configurações

- Os tipos são mutuamente exclusivos.
- Quase todos os tipos e atributos são ortogonais.

# Estudo do código

- Estudo de pthread mutex lock
- Estudo de pthread mutex unlock