POSIX Threads

Walter Fetter Lages

w.fetter@ieee.org

Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Departamento de Engenharia Elétrica

ENG04008 Sistemas de Tempo Real

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Introdução

- Existem diversas implementações de POSIX Threads
  - No espaço do usuário
    - FSU Pthreads
  - No espaço do kernel
    - LinuxThreads
    - RT-Linux
    - RTAI
    - NPTL
- Threads não POSIX
  - DCE threads

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Implementações de Threads

- No espaço do usuário
  - Pouco overhead
  - Não requer suporte do sistema operacional
  - Funções bloqueantes bloqueiam todos os threads
- No espaço do kernel
  - Maior overhead
  - Requer suporte do sistema operacional
  - Funções bloqueantes bloqueiam apenas um thread
  - Escalonamento mais uniforme

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Criação e Destruição de Threads

- `int pthread_create(`
  `pthread_t *thread,`
  `pthread_attr_t *attr,`
  `void *(*start_routine)(void *),`
  `void *arg)`
- `void pthread_exit(void *retval)`
- `int pthread_join(pthread_t th,`
  `void **thread_return)`
- `int pthread_detach(pthread_t th)`

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo

```
int main(void)
{
    pthread_t thread1;
    char *message1 = "Hello World";

    pthread_create(&thread1, NULL,
        (void*)&print_message_function,
        (void*) message1);
    pthread_join(&thread1, NULL);
    exit(0);
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Exemplo (Thread)

```
void print_message_function(void *ptr)
{
     char *message;
     message=(char *) ptr;
     printf("%s ",message);
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Identificação de Threads

- `pthread_t pthread_self(void)`
- `int pthread_equal(pthread_t thread1,`
  `pthread_t thread2)`

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Cancelamento de Theads

- Por default `exit()` cancela todas as threads
- `int pthread_cancel(pthread_t thread)`
- `int pthread_setcancelstate(int state, int *oldstate)`
  - `PTHREAD_CANCEL_ENABLE`
  - `PTHREAD_CANCEL_DISABLE`
- `int pthread_setcanceltype(int type, int *oldtype)`
  - `PTHREAD_CANCEL_ASYNCHRONOUS`
  - `PTHREAD_CANCEL_DEFERRED`
- `void pthread_testcancel(void)`

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Pontos de Cancelamento

- `pthread_join()`
- `pthread_cond_wait()`
- `pthread_cond_timedwait()`
- `pthread_testcancel()`
- `sem_wait()`
- `sigwait()`

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Cleanup

- `void pthread_cleanup_push(`
  `void (*routine)(void *),`
  `void *arg)`
- `void pthread_cleanup_pop(`
  `int execute)`

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Atributos

- `int pthread_attr_init(`
  `pthread_attr_t *attr)`
- `int pthread_attr_destroy(`
  `pthread_attr_t *attr)`

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Estado de Detach

- `int pthread_attr_setdetachstate(`
  `pthread_attr_t *attr,`
  `int detachstate)`
- `int pthread_attr_getdetachstate(`
  `const pthread_attr_t *attr,`
  `int *detachstate)`
- `PTHREAD_CREATE_JOINABLE`
  - Default
- `PTHREAD_CREATE_DETACHED`

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Política de Scheduling

- `int pthread_attr_setschedpolicy(thread_attr_t *attr, int policy)`
- `int pthread_attr_getschedpolicy(const pthread_attr_t *attr, int *policy)`
- `SCHED_OTHER`
  - Default, não tempo-real
- `SCHED_RR`
  - Requer privilégio de superusuário
- `SCHED_FIFO`
  - Requer privilégio de superusuário

Parâmetros de Scheduling

- `int pthread_attr_setschedparam(`
  `pthread_attr_t *attr,`
  `const struct sched_param *param)`
- `int pthread_attr_getschedparam(`
  `const pthread_attr_t *attr,`
  `struct sched_param *param)`
- Prioridade de Scheduling
  - Pode ser alterado após a criação da thread com `pthread_setschedparam()`

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Herança de Scheduling

- `int pthread_attr_setinheritsched(`
  `pthread_attr_t *attr,`
  `int *attr,int *inherit)`
- `int pthread_attr_getinheritsched(`
  `const pthread_attr_t *attr,`
  `int *inherit)`
- `PTHREAD_EXPLICIT_SCHED`
  - Política e parâmetros de scheduling são definidos pelos atributos
- `PTHREAD_INHERIT_SCHED`
  - Política e parâmetros de scheduling são herdados do thread que criou o novo thread

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Escopo de Scheduling

- `int pthread_attr_setscope(`
  `pthread_attr_t *attr,`
  `int scope)`
- `int pthread_attr_getscope(`
  `const pthread_attr_t *attr,`
  `int *scope);`
- `PTHREAD_SCOPE_SYSTEM`
  - Prioridade interpretada como relativa à todos os outros processos
- `PTHREAD_SCOPE_PROCESS`
  - Prioridade relativa apenas às outras threads do mesmo processo