Criação de threads POSIX

A criação de threads em UNIX é feita através da biblioteca padrão PThreads (*POSIX Threads*). Estes exercícios visam estudar exemplos do uso de threads e estimular o aluno a compreender sua dinâmica.

Exercício 1

Dado o programa abaixo:

thread-create.c

```
/*
Criação de threads POSIX em UNIX.

Compilar com gcc -Wall thread-create.c -o thread-create -lpthread

Carlos Maziero, DINF/UFPR 2020
*/

#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define NUM_THREADS 16

void *threadBody (void *id)
{
  long tid = (long) id ; // ID da thread

  printf ("t%02ld: Olá!\n", tid) ;
  sleep (3) ;
  printf ("t%02ld: Tchau!\n", tid) ;
  pthread_exit (NULL) ;
}

int main (int argc, char *argv[])
{
  pthread_t thread [NUM_THREADS] ;
  long i, status ;

  for (i=0; i<NUM_THREADS; i++)
  {
    printf ("Main: criando thread %02ld\n", i) ;
    status = pthread_create (&thread[i], NULL, threadBody, (void *) i) ;
    if (status)
    {
      perror ("pthread_create") ;
      exit (1) ;
    }
  }
  printf ("Main: fim\n") ;
```

```
  pthread_exit (NULL) ;
}
```

1. Analise seu código e o comente detalhadamente.
2. A ordem de criação, ativação e encerramento das threads é a mesma? Por que?
3. Desenhe o diagrama de tempo de sua execução.

Exercício 2

Dado o programa abaixo:

thread-join.c

```
/*
Criação de threads POSIX em UNIX, com operação join().

Compilar com gcc -Wall thread-join.c -o thread-join -lpthread

Carlos Maziero, DINF/UFPR 2020
*/

#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define NUM_THREADS 16

void *threadBody (void *id)
{
  long tid = (long) id ;

  printf ("t%02ld: Olá!\n", tid) ;
  sleep (3) ;
  printf ("t%02ld: Tchau!\n", tid) ;
  pthread_exit (NULL) ;
}

int main (int argc, char *argv[])
{
  pthread_t thread [NUM_THREADS] ;
  pthread_attr_t attr ;
  long i, status ;

  // para permitir a operação "join" sobre as threads
  pthread_attr_init (&attr) ;
  pthread_attr_setdetachstate (&attr, PTHREAD_CREATE_JOINABLE) ;

  for (i=0; i<NUM_THREADS; i++)
  {
    printf ("Main: criando thread %02ld\n", i) ;
    status = pthread_create (&thread[i], &attr, threadBody, (void *) i) ;
    if (status)
```

```
    {
      perror ("pthread_create") ;
      exit (1) ;
    }
  }

  for (i=0; i<NUM_THREADS; i++)
  {
    printf ("Main: aguardando thread %02ld\n", i);
    status = pthread_join (thread[i], NULL) ;
    if (status)
    {
      perror ("pthread_join") ;
      exit (1) ;
    }
  }
  printf ("Main: fim\n") ;
  pthread_attr_destroy (&attr) ;
  pthread_exit (NULL) ;
}
```

1. Analise seu código e o comente detalhadamente.
2. Explique o objetivo do parâmetro `attr` e da chamada `pthread_join`.
3. Desenhe o diagrama de tempo de sua execução.

Exercício 3

Dado o programa abaixo:

thread-print.c

```
/*
Criação de threads POSIX em UNIX, com impressão de variáveis.

Compilar com gcc -Wall thread-print.c -o thread-print -lpthread

Carlos Maziero, DINF/UFPR 2020
*/

#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define NUM_THREADS 16

int x = 0 ;  // variável global

void *threadBody (void *id)
{
  long tid = (long) id ;

  x++ ;
```

```
  printf ("t%02ld: Olá!   (x=%02d)\n", tid, x) ;
  sleep (3) ;
  x++ ;
  printf ("t%02ld: Tchau! (x=%02d)\n", tid, x) ;
  pthread_exit (NULL) ;
}

int main (int argc, char *argv[])
{
  pthread_t thread [NUM_THREADS] ;
  long i, status ;

  for (i=0; i<NUM_THREADS; i++)
  {
    printf ("Main: criando thread %02ld\n", i) ;
    status = pthread_create (&thread[i], NULL, threadBody, (void *) i) ;
    if (status)
    {
      perror ("pthread_create") ;
      exit (1) ;
    }
  }
  printf ("Main: fim\n") ;
  pthread_exit (NULL) ;
}
```

1. Analise seu código e o comente detalhadamente.
2. Compare a evolução da variável x neste programa com aquela que ocorreria em um programa equivalente usando a chamada de sistema `fork`.
3. Desenhe o diagrama de tempo de sua execução, mostrando a evolução do valor da variável x.

Observações

- Os programas devem ser compilados com a opção `-lpthread` para ligá-los à biblioteca *PThreads*.
- Informações detalhadas sobre as chamadas de sistema utilizadas podem ser encontradas nas páginas de manual do sistema UNIX e também no site LLNL POSIX Threads Programming Tutorial.
- Um relatório no formato apropriado deve ser produzido pelo aluno, contendo os resultados obtidos nos exercícios.

From:
https://wiki.inf.ufpr.br/maziero/ - **Prof. Carlos Maziero**

Permanent link:
https://wiki.inf.ufpr.br/maziero/doku.php?id=so:criacao_de_threads

Last update: **2022/11/04 09:54**