

Sistemas Operacionais - s11

Criação de Threads POSIX

Rodrigo Yudi Endo
Vítor Hiroaki Yoshioka

Exercício 1

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define NUM_THREADS 16 // define o número de threads que serão criadas

void *threadBody (void *id) //função que será chamada pela thread, recebe o
valor do id da thread
{
    long tid = (long) id ; // guarda o valor do id da thread

    printf ("t%02ld: Olá!\n", tid) ; /* imprime uma mensagem quando a thread é
chamada, imprimindo o id dela*/
    sleep (3) ; // espera 3 ms
    printf ("t%02ld: Tchau!\n", tid) ; /* imprime uma mensagem quando a thread
é chamada, imprimindo o id dela*/

    pthread_exit (NULL) ; // encerra o processo
}

int main (int argc, char *argv[])
{
    pthread_t thread [NUM_THREADS] ; // inicializa um array de threads

    long i, status ; /* define um contador (i) e uma variável que armazena o
retorno da função pthread_create() */

    for (i=0; i<NUM_THREADS; i++) /* cria um loop para inicializar todas as
threads */
    {
        printf ("Main: criando thread %02ld\n", i) ; /* avisa qual thread esta
```

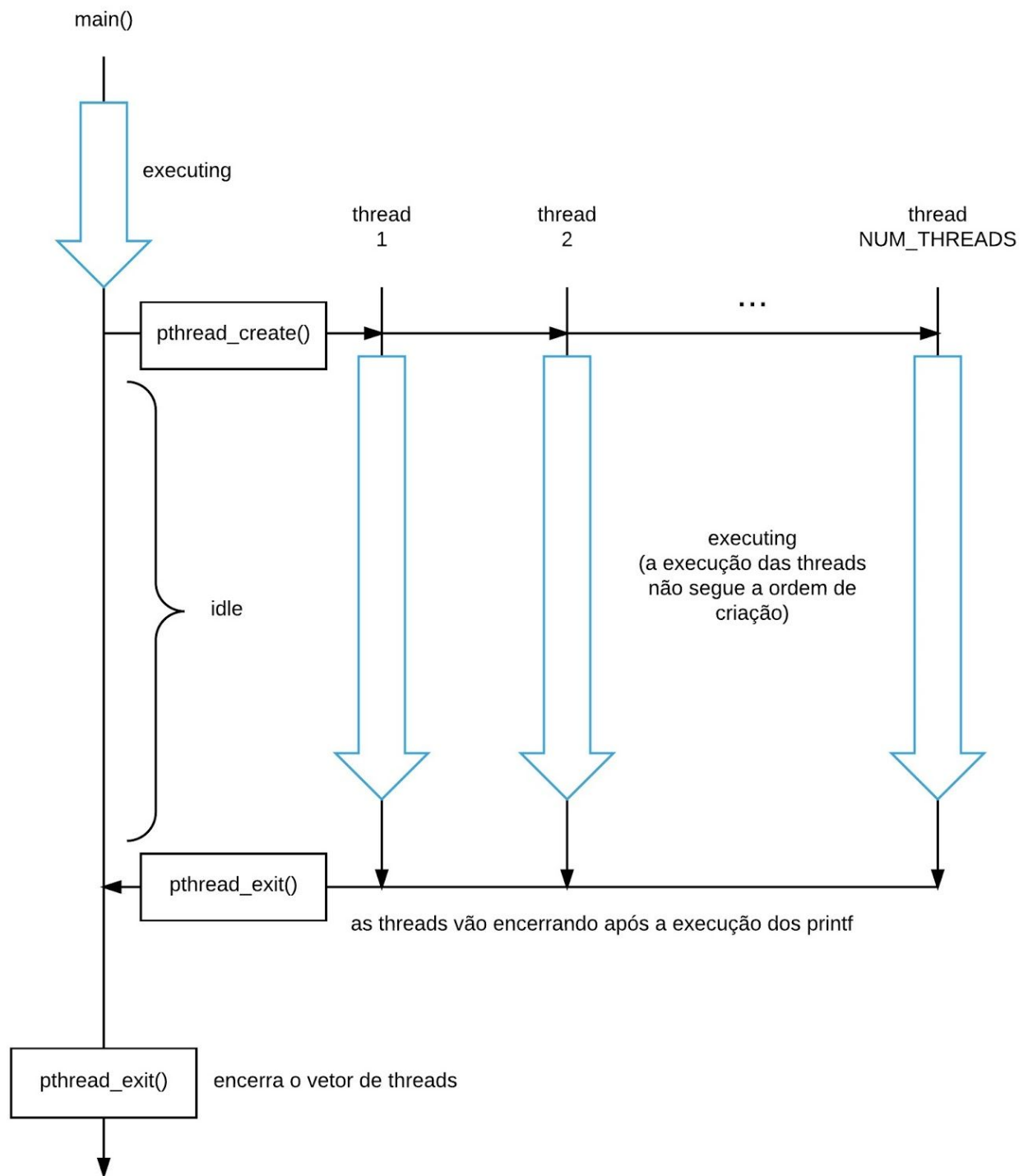
```
sendo criada */
```

```
    status = pthread_create (&thread[i], NULL, threadBody, (void *) i) ;  
/*pthread_create() é chamada recebendo o id da thread, as propriedades default  
do método, a função a ser executada e o valor do ponteiro a ser passado para a  
função. Quando essa função é chamada, ela retorna 0 se for bem sucedida ou o  
valor correspondente ao erro ocorrido. Esse valor é então guardado na variável  
status. */
```

```
    if (status) //caso haja erro, pthread_create() retorna um valor  
diferente de 0, correspondente ao código do erro  
    {  
        perror ("pthread_create") ;  
        exit (1) ;  
    }  
}  
pthread_exit (NULL) ; // encerra a thread  
}
```

B-) Não tem como saber a ordem de execução, pois não é possível controlar o escalonador do processador. Os resultados podem variar a cada execução.

C-)



Exercício 2

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

#define NUM_THREADS 16

void *threadBody (void *id) //função que será chamada pela thread, recebe o
valor do id da thread
{
    long tid = (long) id ; // guarda o valor do id da thread

    printf ("t%02ld: Olá!\n", tid) ; /* imprime uma mensagem quando a thread é
chamada, imprimindo o id dela*/
    sleep (3) ;    // espera 3 ms
    printf ("t%02ld: Tchau!\n", tid) ; /* imprime uma mensagem quando a thread
é chamada, imprimindo o id dela*/

    pthread_exit (NULL) ; // encerra o processo
}

int main (int argc, char *argv[])
{
    pthread_t thread [NUM_THREADS] ; // cria um array de threads
    pthread_attr_t attr ;    // variável que guarda os atributos
    long i, status ; /* define um contador (i) e uma variável que armazena o
retorno da função pthread_create */

    pthread_attr_init (&attr) ; /*função inicializa o objeto de atributos da
thread com valor default*/
    pthread_attr_setdetachstate (&attr, PTHREAD_CREATE_JOINABLE) ; /*permite o
uso de pthread_join na execução, que faz com que a próxima thread espere que a
anterior termine sua execução*/

    for (i=0; i<NUM_THREADS; i++) /* cria um loop para inicializar todas as
threads */
```

```

{
    printf ("Main: criando thread %02ld\n", i) ; /* avisa qual thread está
sendo executada*/

    status = pthread_create (&thread[i], &attr, threadBody, (void *) i) ;

/*pthread_create() é chamada recebendo o id da thread, as propriedades
armazenadas em &attr do método, a função a ser executada e o valor do ponteiro
a ser passado para a função. Quando essa função é chamada, ela retorna 0 se
for bem sucedida ou o valor correspondente ao erro ocorrido. Esse valor é
então guardado na variável status. */

    if (status) //caso haja erro, pthread_create() retorna um valor
diferente de 0, correspondente ao código do erro
    {
        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) ; /* utiliza o método join com
as threads criadas no loop anterior, fazendo com que cada thread suspenda sua
execução até que a anterior termine sua execução*/
    if (status) //caso haja erro, pthread_join() retorna um valor diferente
de 0, correspondente ao código do erro */

    {
        perror ("pthread_join") ;
        exit (1) ;
    }
}

pthread_attr_destroy (&attr) ; /*quando um objeto de atributos de uma
thread não é mais necessário, ele deve ser destruído utilizando essa função*/

```

```
pthread_exit (NULL) ; // encerra o processo  
}
```

B-) O parâmetro attr é o objeto de atributos da thread, utilizado na chamada da função pthread_join(), que faz com que cada thread suspenda sua execução até que a thread anterior termine a sua.

C-)

