



INSTITUTO FEDERAL
RIO GRANDE DO SUL

Programação com Objetos Distribuídos

- Programação Paralela – uso de threads.

Março / 2018

O que São Threads

- Threads são sub-processos no sistema operacional. Uma thread representa uma linha de execução das instruções do programa. É menos custoso gerenciar **threads** do que processos.
- Implementações de threads que aderem ao padrão POSIX1003.1-200X são conhecidas como POSIX threads ou simplesmente pthreads.
- C e C++ necessitam de biblioteca específica para processamento MULTITHREADING: Posix p_thread

Criando Threads

```
int pthread_create(pthread_t *tid,  
                  const pthread_attr_t *attr,  
                  void * (* start)(void *),  
                  void *arg);
```

- Cada novo thread é representado por um identificador (thread identifier ou tid) de tipo pthread_t.
- O segundo parâmetro serve para indicar uma série de atributos e propriedades que o novo thread deverá ter.
- O terceiro parâmetro indica a função de início do thread.
- Finalmente o quarto parâmetro é o valor do argumento a ser passado à função de início, como seu parâmetro.
 - **Uma thread retorna 0 no caso de sucesso e um código de erro no caso contrário.**

Finalizando Threads

- Um thread termina quando a função de início, indicada quando da criação, retornar, ou quando o próprio thread invocar o serviço de terminação:

```
pthread_exit(void *value_ptr);
```

Onde **value_ptr** é o ponteiro que o thread deve ter como resultado.

Sincronização de Threads

```
int pthread_join (pthread_t thread,  
                 void ** status);
```

- O comando `join` é executado para esperar o fim de uma Thread , de maneira semelhante a `wait()` para processos (bloqueia o thread que *chamou* esta função até que o thread com o identificador *thread_id* termine.).
 - O Identificador da thread deve ser especificado.

Exemplo 01

```
#include <stdio.h>
#include <pthread.h>
void *HelloThread (void *argumentos)    {
    pthread_t NumThread;
    NumThread = pthread_self();
    printf("\n Ola Mundo! Eu sou a thread n# %d \n", NumThread);
    pthread_exit(NULL);
}
int main ( ) {
    pthread_t thread;
    int flag, i;
    printf("Iniciando do programa de Threads. ");
    flag = pthread_create(&thread, NULL, HelloThread, NULL);
    if (flag!=0) printf("Erro na criação de thread\n");
    pthread_exit(NULL);
    return 0;
}
```

Exemplo 01

- Faça um programa de conte de 1 até 5.000.000, mostrando somente os pares, de 5.000.000 até 9.000.000, mostrando somente os impares e de um até 1 até 9.000.000 mostrando todos múltiplos de 7 e 9.
- Todos os resultados deverão ser escritos em três arquivos diferentes.
- Após criar o programa procedural, crie uma versão do mesmo programa usando threads.
- Qual foi o ganho do tempo de execução?

Exemplo 02

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

struct valor{
    int lacos;
    int id;};

void *processa(void *argumento){
    struct valor *v = (struct valor *) argumento;
    int i = 0;
    while (i < v->lacos)    {
        i++;    }
    printf("Eu sou a thread %d. Executei %d lacos antes de executar\n",v->id,v->lacos);
    pthread_exit(NULL);
}

int main(){
    pthread_t linhas[10];
    int execute,i;
    struct valor *v;
    srand(time(NULL));
    for (i=0;i<10;i++){
        v = (struct valor *) malloc(sizeof(struct valor *));
        v->lacos = (rand())+200000000;
        v->id = i;
        printf("Criando a thread <%d> com <%d> lacos\n",i,v->lacos);
        execute = pthread_create(&linhas[i],NULL,processa,(void *)v);
    }
    pthread_exit(NULL);
}
```


Atividade

1) Tendo um array de inteiros (valores) com 20000 posições (inicializado automaticamente onde a posição corresponde a valor), pretende-se calcular $\text{resultado}[i] = \text{valores}[i] * 5 + 55$.

- a) Execute de forma sequencial. Tome nota do tempo gasto na execução.
- b) Execute com 2 threads, cada uma calculando 10000 posições.
- c) Execute com 4 threads, cada uma calculando 5000 posições.
- d) Execute com 8 threads, cada uma calculando 2500 posições.

Qual teve o melhor desempenho? Por que?

Atividade

Altere o cálculo do exercício anterior para `resultado[i] = valores[i] * rand() ate 5 + rand() até 1000.`

Alguma diferença no desempenho?

Atividade Prática

a) Crie uma programa *procedural* que leia o arquivo (arquivo.txt disponível no moodle) e:

1- Conte as palavras (únicas).

2- Crie uma lista ordenada de palavras.

b) Após a implementação do programa *procedural*, alterar de modo que execute cada função paralelamente.

c) Você deverá medir o tempo de execução do programa *procedural comparando com a versão* paralela e verificar qual foi o ganho (ou perda) de desempenho.