

## Capítulo 03

# *Greetings!*

Lucas Sousa de Oliveira (10/59491)

26 de agosto de 2013

## Objetivo

Introduzir o aluno ao processamento paralelo através do estudo de um programa simples chamado “*Greetings!*”, equivalente ao famoso “*Hello World!*”.

## Resumo

```
1  #include <stdio.h>
2  #include <string.h>
3  #include "mpi.h"
4
5  int main(int argc, char* argv[]) {
6      int      my_rank;
7      int      p;
8      int      source;
9      int      dest;
10     int      tag = 0;
11     char      message[100];
12     MPI_Status status;
13
14     MPI_Init(&argc, &argv);
15     MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
16     MPI_Comm_size(MPI_COMM_WORLD, &p);
17
18     if (my_rank != 0) {
19         sprintf(message, "Greetings from process %d!", my_rank);
20         dest = 0;
21         MPI_Send(message, strlen(message)+1, MPI_CHAR, dest, tag, MPI_COMM_WORLD);
22     } else {
23         for (source = 1; source < p; source++) {
24             MPI_Recv(message, 100, MPI_CHAR, source, tag, MPI_COMM_WORLD, &status);
25             printf("%s\n", message);
26         }
27     }
28
29     MPI_Finalize();
30 }
```

## Exercícios

1. Crie um arquivo contendo o programa “*Greetings!*”. Descubra como compilar e executar em um número diferente de processadores. Qual é a saída do programa se ele for executado com apenas um processo? Quantos processadores podem ser usados?
2. Modifique o programa “*Greetings!*” de forma que ele use elementos genéricos para *source* e *tag*. Existe alguma diferença na saída do programa?
3. Tente modificar alguns parametros de MPI\_Send e MPI\_Recv (e.g., *count*, *datatype*, *source*, *dest*). O que acontece quando se executa o programa? Ele falha? Ele pára?
4. Modifique o programa “*Greetings!*” de forma que todos os processos enviam uma mensagem para o processo p-1.

## Trabalho de programação

1. Escreva um programa em que cada processo *i* envia mensagem ao processo  $(i+1)\%p$ . (Cuidado em como *i* calcula de quem deve receber a mensagem). O processo *i* deverá enviar uma mensagem para *i+1* e depois receber de *i-1*, ou o contrário? Faz diferença? O que acontece quando o programa é rodado em um processador?