

Bacharelado em Ciência da Computação - DCC/IM-UFRJ
Programação Paralela e Distribuída
Prof. Gabriel P. Silva
2ª Lista de Exercícios – 25/11/2015

1. Quais são os principais objetivos do padrão MPI? (0,5 ponto)
2. O que são comunicadores? Qual o comunicador padrão no MPI? (0,5 ponto)
3. Qual a diferença entre MPI_Init e MPI_Initialized? E entre MPI_Finalize e MPI_Finalized? (0,5 ponto)
4. Dado o trecho de código a seguir, complete o código declarando corretamente os parâmetros das funções e todos os demais trechos necessários ao correto funcionamento do programa. O programa deve imprimir adequadamente todas as informações coletadas com as funções utilizadas. (0,5 ponto)

```
MPI_Comm_size(aux, &aux1);  
MPI_Comm_rank(aux, &aux2);  
MPI_Get_processor_name(aux3, &aux4);  
MPI_Get_version (&aux5, &aux6);
```

5. Para que servem os tipos padrão MPI_BYTE e MPI_PACKED? (0,5 ponto)
6. Qual a diferença entre os tipos MPI_INT e MPI_INT_32_T? (0,5 ponto)
7. Qual o resultado da execução do trecho de código a seguir? (0,5 ponto)

```
MPI_Reduce(&aux1, &aux2, 3, MPI_FLOAT, MPI_MAX, 2, MPI_COMM_WORLD);
```

8. Escreva um trecho de código para determinar o número total de elementos efetivamente recebido pela rotina abaixo. (0,5 ponto)

```
MPI_Recv(&vetor, 100, MPI_FLOAT, MPI_ANY_SOURCE, MPI_ANY_TAG, MPI_COMM_WORLD,  
&estado);
```

9. Escreva um trecho de código com MPI_Probe, MPI_Count e MPI_Recv para receber uma mensagem alocando um vetor com o tamanho exato da mesma. (0,5 ponto)
10. Considere o seguinte trecho de código: (0,5 ponto)

```
numt = MPI_Group_size(com);  
do {  
    MPI_Send(mensagem, strlen(mensagem)+1, MPI_CHAR, numt-1, 1, com);  
} while (--numt);
```

Escreva quatro versões diferentes de trechos de código usando a rotina MPI_Recv que poderiam ser utilizados para a recepção desta mensagem com sucesso. Explique o porquê.

11. Quais as restrições existentes no uso das operações de comunicação coletivas? (0,5 ponto)
12. Qual a diferença entre MPI_Bcast e MPI_Scatter? (0,5 ponto)
13. Enumere e descreva sucintamente o funcionamento de quatro operações de comunicação coletivas no MPI. (0,5 ponto)
14. Enumere e descreva quatro tipos de operações de redução utilizadas na função MPI_Reduce. (0,5 ponto)

15. Em um programa MPI correto, o comprimento do buffer de recepção em uma função MPI_Recv: (0,5 ponto)

- (a) deve ser exatamente igual ao comprimento da mensagem enviada.
- (b) deve ser maior ou igual ao tamanho da mensagem enviada .
- (c) pode ser qualquer valor; se o comprimento do buffer for menor que o comprimento da mensagem enviada, a mensagem será truncada.
- (d) deve ser pelo menos uma posição maior que a mensagem enviada .
- (e) deve ser menor ou igual ao comprimento da mensagem enviada.

16. Um vetor de comprimento 10 é armazenado na memória local de um processo MPI. O programador deseja enviar os primeiros 3 elementos do vetor para o processo 0, os próximos 3 elementos para o processo 1 e os últimos 4 elementos para o processo 2. Qual a função MPI que você utilizaria para realizar essa função? (0,5 ponto)

17. Modifique o programa do Exemplo 5.1 do livro para que utilize rotinas MPI_Sendrecv para envio e/ou recepção das mensagens. (0,5 ponto)

18. Escreva uma rotina para calcular o produto escalar de dois vetores utilizando rotinas do MPI. Utilize a função de geração de números aleatórios no intervalo [1 , 10) para atribuir valores iniciais aos vetores. Considere cada vetor com N posições e divida a operação entre P processos distintos. Considere que a divisão de N por P não tem resto. Utilize rotinas de comunicação coletivas para envio do tamanho e conteúdo do vetor para os processos e recepção dos valores parciais calculados. (0,5 ponto)

19. Use o modelo “bolsa de tarefas” para resolver o problema de encontrar todos os números primos no intervalo de 3 a 100000, usando N processos. Atribua a um dos processos o papel de “mestre” (gerencia a distribuição de tarefas) e aos demais processos o papel de “escravos” (executam a tarefa de avaliar a primalidade de um número). (0,5 ponto)

```
int primo (int n) {  
    int i;  
    for (i = 3; i < sqrt(num) + 1 ; i += 2) {  
        if (n%i == 0)  
            return 0;  
    }  
    return 1;  
}
```

20. Um programa contém uma chamada para a função MPI_Bcast com o tipo de dados MPI_DOUBLE usado em cada processo. Qual das seguintes afirmativas deve ser correta se o programa está correto? (0,5 ponto)

- (a) O argumento count usado em um processo que não seja a raiz deve ser exatamente igual ao argumento count no processo raiz.
- (b) O argumento count usado em um processo que não seja a raiz pode ser diferente, desde que sejam maiores ou iguais ao argumento count no processo raiz.
- (c) O argumento count usado em um processo que não seja a raiz devem ser o mesmo número, mas este número deve ser maior que o argumento count no processo raiz.
- (d) O argumento count usado em um processo que não seja a raiz devem ser o mesmo número, mas este número deve ser maior pelo menos uma posição que o argumento count no processo raiz.
- (e) O argumento count usado em um processo que não seja a raiz podem ser qualquer número, se eles forem menores que o argumento count no processo raiz, a mensagem será apenas truncada.