

Trabalho 2: Programação Paralela para Multiprocessadores com Memória Distribuída Usando MPI

- Remoção dos 0s de um vetor:

- **Entrada:** Vetor vIn de n elementos inteiros
- **Saída:** Vetor $vOut$ de m elementos inteiros, $m \leq n$
 - $vOut$ é uma cópia de vIn com os elementos em 0 removidos
 - Elementos restantes **contíguos** em $vOut$

- Exemplo:

- **Entrada:**

- $n = 10$

- $vIn =$

10	77	0	24	0	0	31	58	9	2
0	1	2	3	4	5	6	7	8	9

- **Saída:**

- $m = 7$

- $vOut =$

10	77	24	31	58	9	2
0	1	2	3	4	5	6

Algoritmo Sequencial para Remoção dos 0s de um Vetor

```
void remove0(int n, int *vIn, int *m, int *vOut)
{
    int i,
        c = 0;

    for (i = 0; i < n; i++)
    {
        if (vIn[i] != 0)
        {
            vOut[c] = vIn[i];
            c++;
        }
    }
    *m = c;
}
```

Ideia da Paralelização da Remoção dos 0s de um Vetor

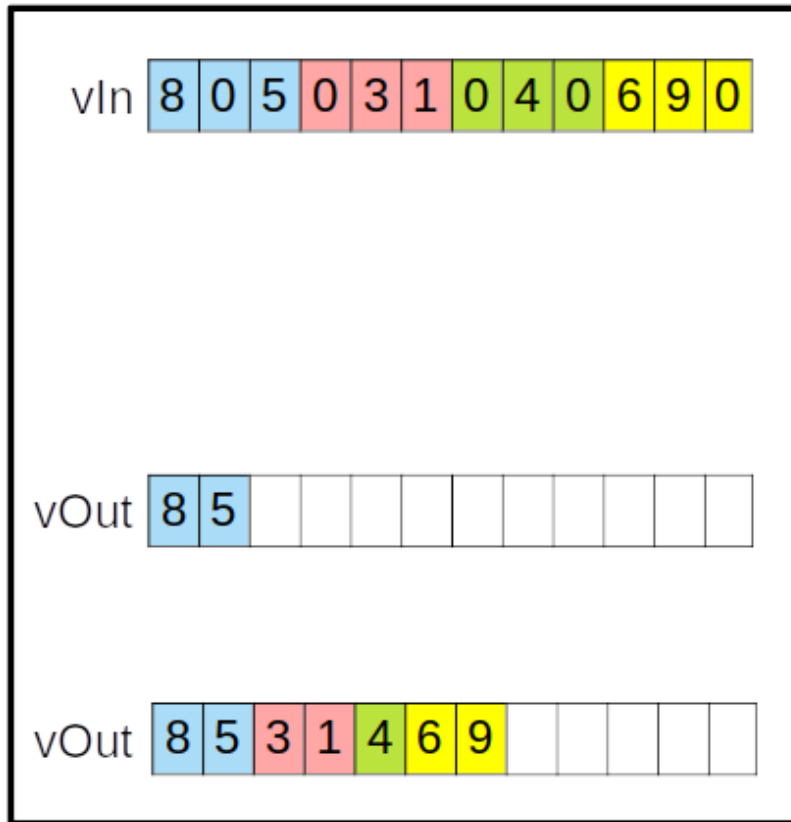
- Para p = número de processos (n múltiplo de p)
- **Processo 0:**
 - Lê vIn do arquivo de entrada
 - comunicação
 - Remove 0s de um pedaço de vIn com $\sim n/p$ elementos
 - comunicação
 - Escreve $vOut$ no arquivo de saída
- **Demais processos:**
 - comunicação
 - Remove 0s de um pedaço de vIn com $\sim n/p$ elementos
 - comunicação
- **ATENÇÃO:**
 - Processo 0 é o único que possuirá os vetores vIn e $vOut$ inteiros
 - Processamento da remoção de 0s em cada processo deve ser $O(n/p)$

Medição de tempo envolve tudo, excluindo leitura do arquivo de entrada e escrita do arquivo de saída

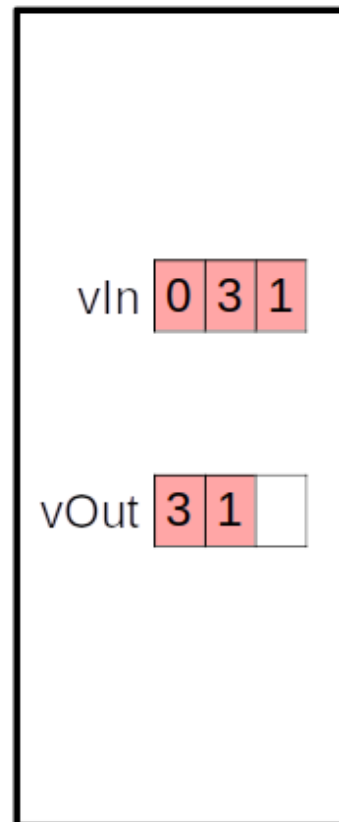
Ideia da Paralelização da Remoção dos 0s de um Vetor

- **Exemplo:** $n = 12$ e $p = 4$

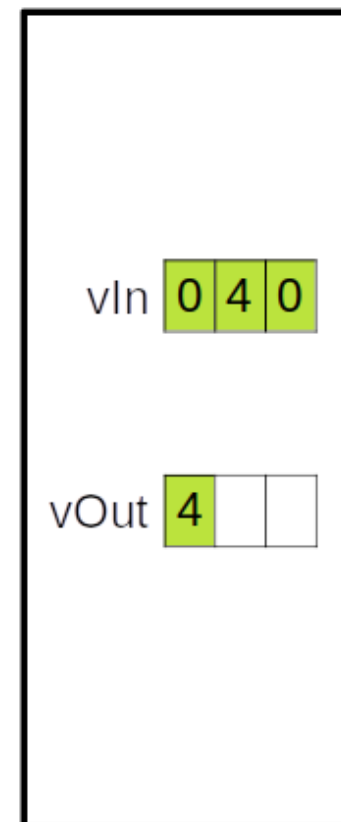
Processo 0



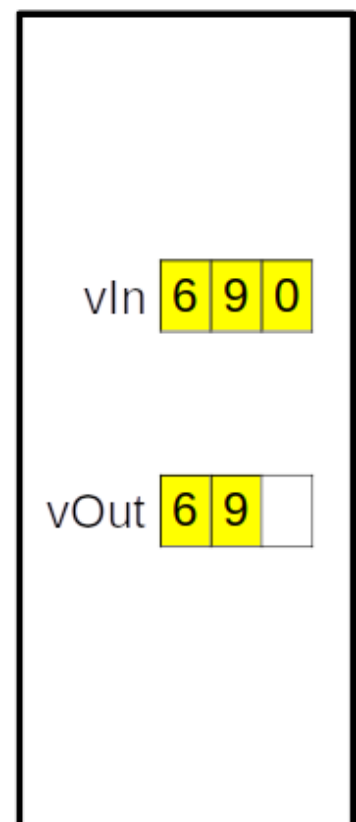
Processo 1



Processo 2



Processo 3



Programa a ser Desenvolvido

- Programa deve ser em C ou C++
- Programa sequencial fornecido
- Desenvolver programa paralelo `remove0_par`, usando MPI
- Programa será compilado com:

```
mpicc  remove0_par.c      -o remove0_par -Wall           ou
mpic++ remove0_par.cpp -o remove0_par -Wall
```

- Programa será executado com argumentos por linha de comando:

```
mpirun -oversubscribe -np 10 remove0_par entrada.txt saida.txt
```

(Obs.: Também serão usados outros valores para `-np`)

- **Submissão no AVA:** um único arquivo .zip com programa fonte paralelo

- **Programa deve ter no cabeçalho:**

Data de entrega: 10/dez, 5a. feira

- Nome dos alunos do grupo (máximo 3 alunos)
- Outras informações, apenas se necessário:
 - Comando diferente de compilação, ...
- **Um único aluno do grupo deve submeter**

Entrada e Saída do Programa

- **Entrada:**
 - **Em um arquivo texto:**
 - n
 - Vetor vIn
- **Saída:**
 - **Na tela:**
 - Tempo de execução (em segundos, medido com `MPI_Wtime`)
 - **Não deve haver nenhuma outra impressão na tela**
 - **Em um arquivo texto:**
 - m
 - Vetor $vOut$
 - **Não deve haver nenhuma outra impressão no arquivo de saída**
- **ATENÇÃO:**
 - **Arquivo de saída produzido será comparado com gabarito da saída, usando: `diff`**

Entrada e Saída do Programa

- Arquivos de entrada e saída fornecidos:

Arquivo	n	Arquivo	m
entrada1.txt	100	saida1.txt	72
entrada2.txt	1000	saida2.txt	745
entrada3.txt	10000	saida3.txt	7574
entrada4.txt	100000	saida4.txt	76138
entrada5.txt	1000000	saida5.txt	760366
entrada6.txt	10000000	saida6.txt	7595231