UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

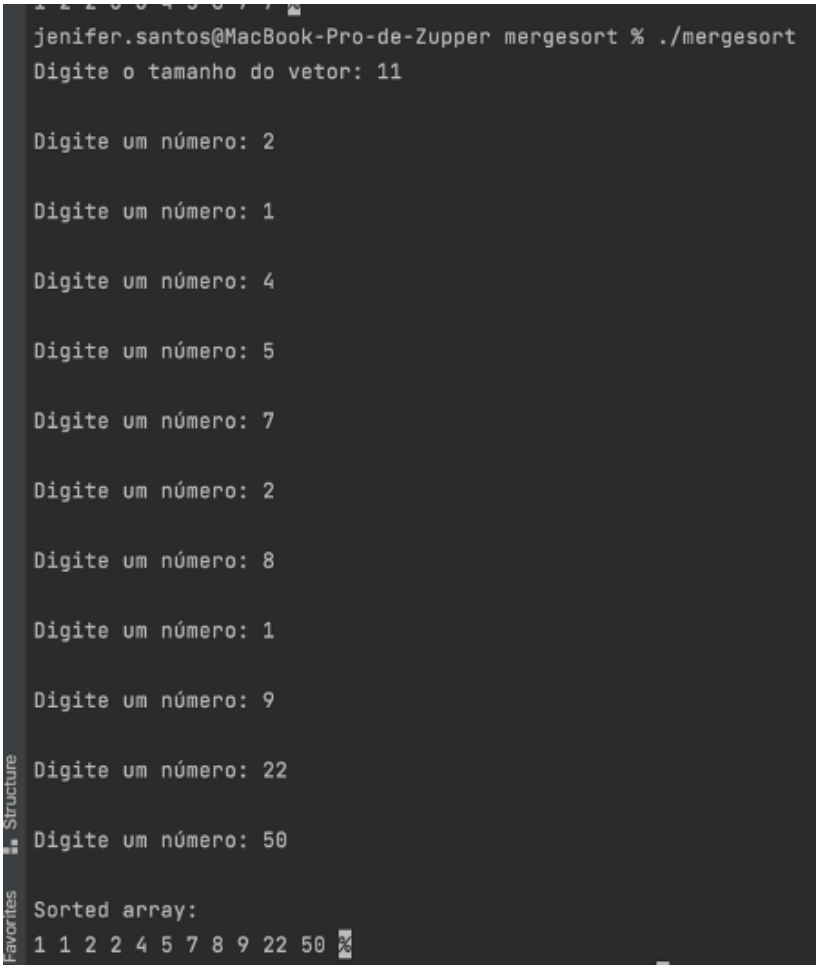
Computação Paralela
Laboratório de Busca e Ordenação

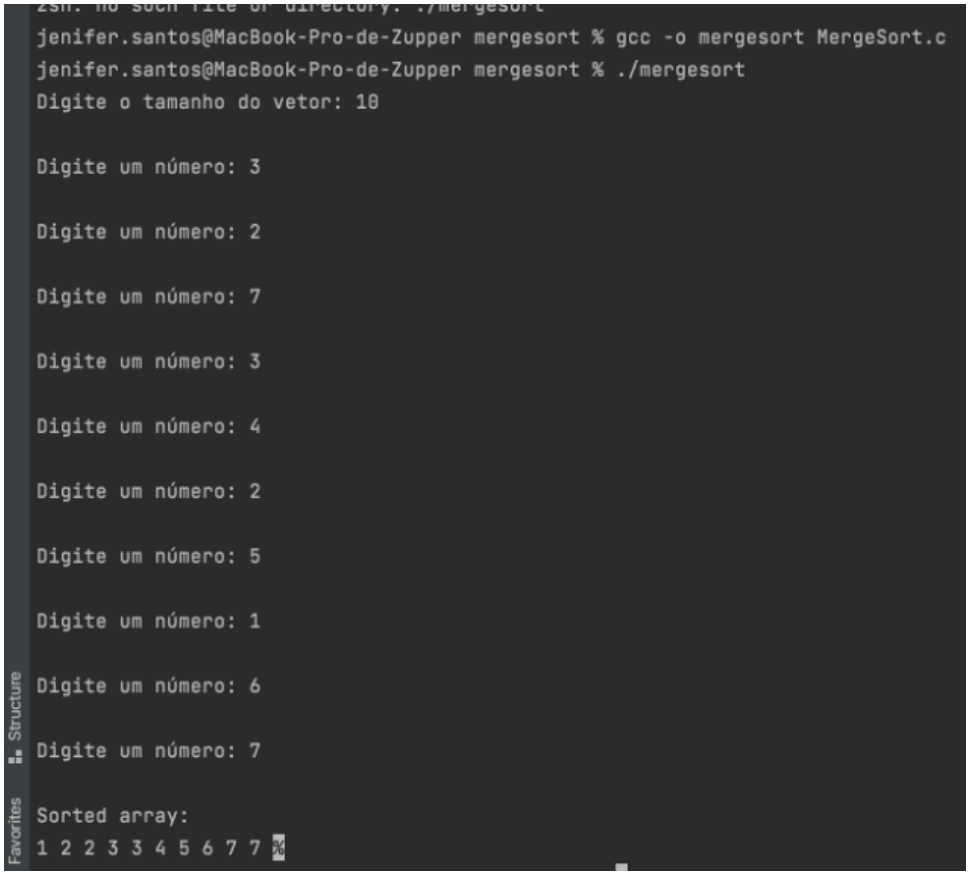
Nome: Giulia Barbieri Quagliano
TIA: 42013070

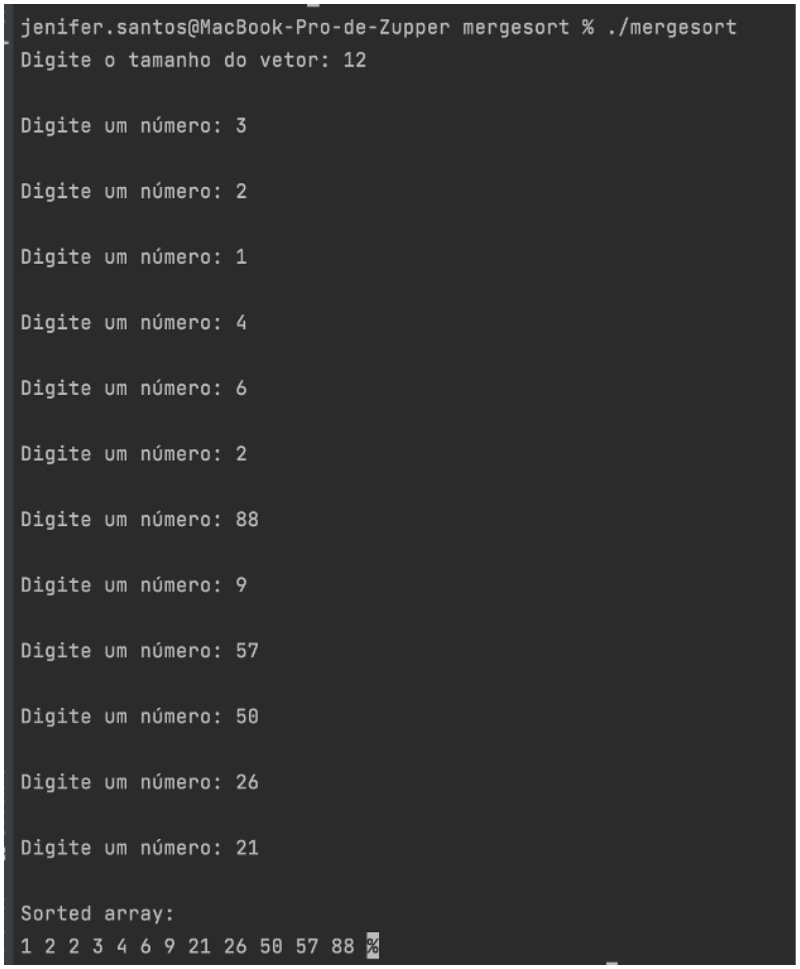
Nome: Jenifer Mathias dos Santos
TIA: 32092946

Merge Sort

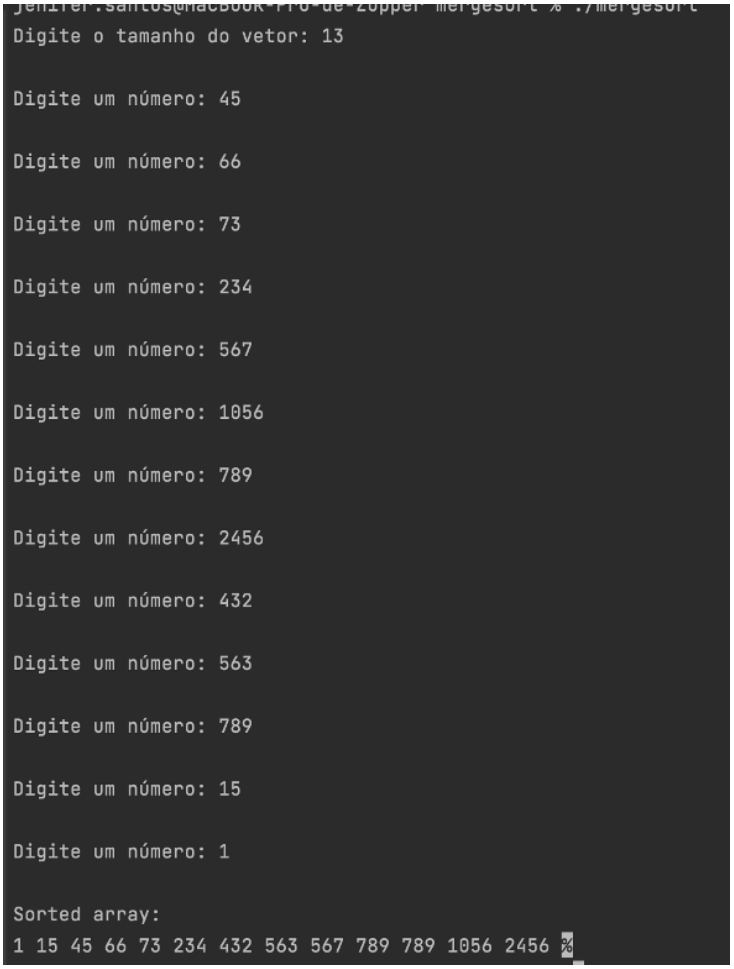
Para implementar o MergeSort de forma paralela, foi criada uma área paralela utilizando "#pragma omp parallel". O código dentro dessa área será executado pelos threads, criadas conforme os núcleos disponíveis ou alguma configuração de ambiente. A variável "meio" será compartilhada entre os threads, o que é permitido através de "shared(int meio)". Dentro da área paralela também temos a área de código que terá conjuntos de seções independentes, definida por "#pragma omp sections". Dentro dela definimos cada conjunto com "#pragma omp section". Para resolver a condição de corrida, utilizamos "#pragma omp barrier" para definir um ponto de sincronização entre os threads, onde a área paralela aguarda a finalização dos threads antes de continuar.


jenifer.santos@MacBook-Pro-de-Zupper mergesort % ./mergesort
Digite o tamanho do vetor: 11
Digite um número: 2
Digite um número: 1
Digite um número: 4
Digite um número: 5
Digite um número: 7
Digite um número: 2
Digite um número: 8
Digite um número: 1
Digite um número: 9
Digite um número: 22
Digite um número: 50
Sorted array:
1 1 2 2 4 5 7 8 9 22 50
Structure
Favorites


zsh: no such file or directory: ./mergesort
jenifer.santos@MacBook-Pro-de-Zupper mergesort % gcc -o mergesort MergeSort.c
jenifer.santos@MacBook-Pro-de-Zupper mergesort % ./mergesort
Digite o tamanho do vetor: 10
Digite um número: 3
Digite um número: 2
Digite um número: 7
Digite um número: 3
Digite um número: 4
Digite um número: 2
Digite um número: 5
Digite um número: 1
Digite um número: 6
Digite um número: 7
Sorted array:
1 2 2 3 3 4 5 6 7 7
Structure
Favorites

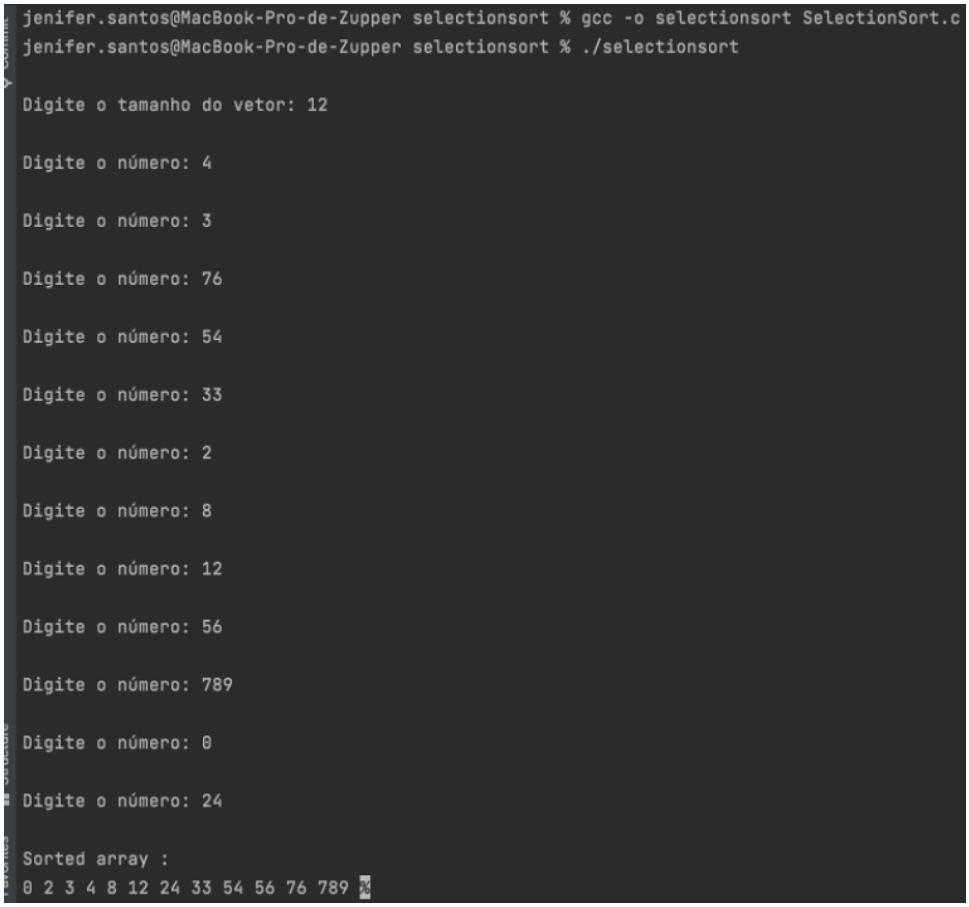

jenifer.santos@MacBook-Pro-de-Zupper mergesort % ./mergesort
Digite o tamanho do vetor: 12
Digite um número: 3
Digite um número: 2
Digite um número: 1
Digite um número: 4
Digite um número: 6
Digite um número: 2
Digite um número: 88
Digite um número: 9
Digite um número: 57
Digite um número: 50
Digite um número: 26
Digite um número: 21
Sorted array:
1 2 2 3 4 6 9 21 26 50 57 88 %

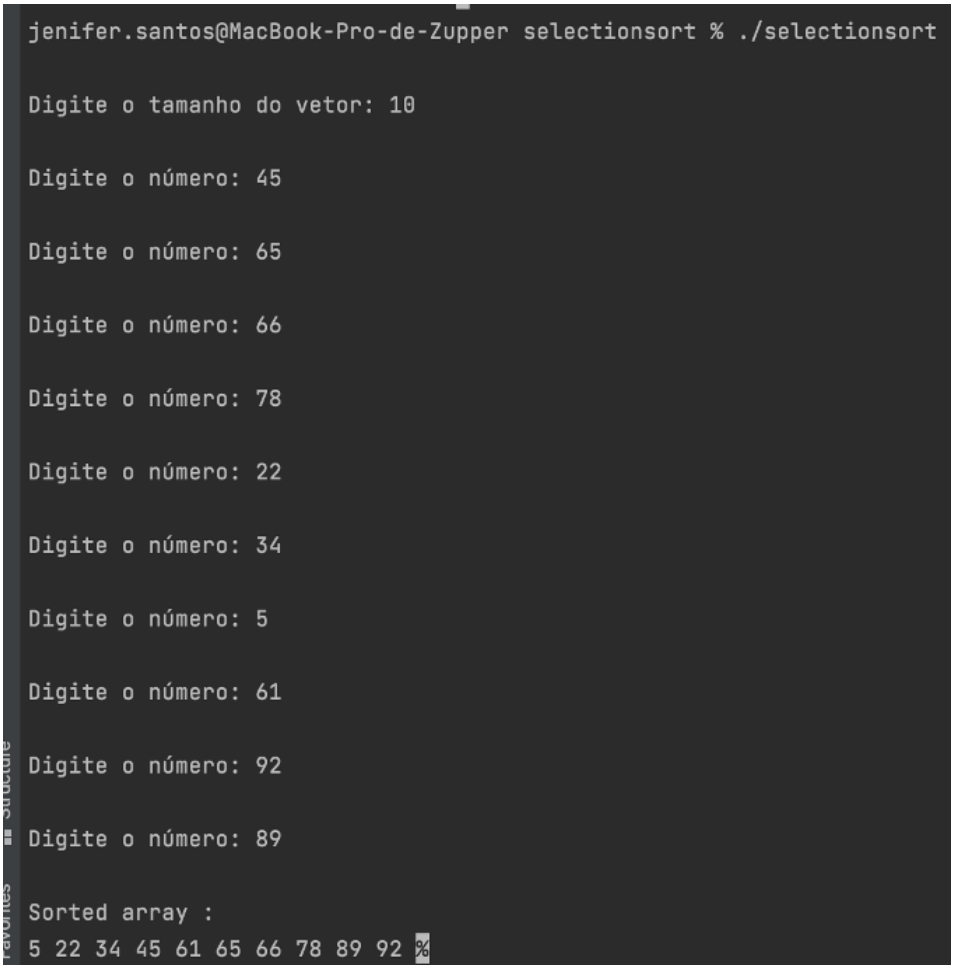
```
jenifer.santos@MacBook-Pro-de-Zupper mergesort % ./mergesort
Digite o tamanho do vetor: 9

Digite um número: 3

Digite um número: 4

Digite um número: 2

Digite um número: 7

Digite um número: 78

Digite um número: 104

Digite um número: 650

Digite um número: 986

Digite um número: 5

Sorted array:
2 3 4 5 7 78 104 650 986 %
```

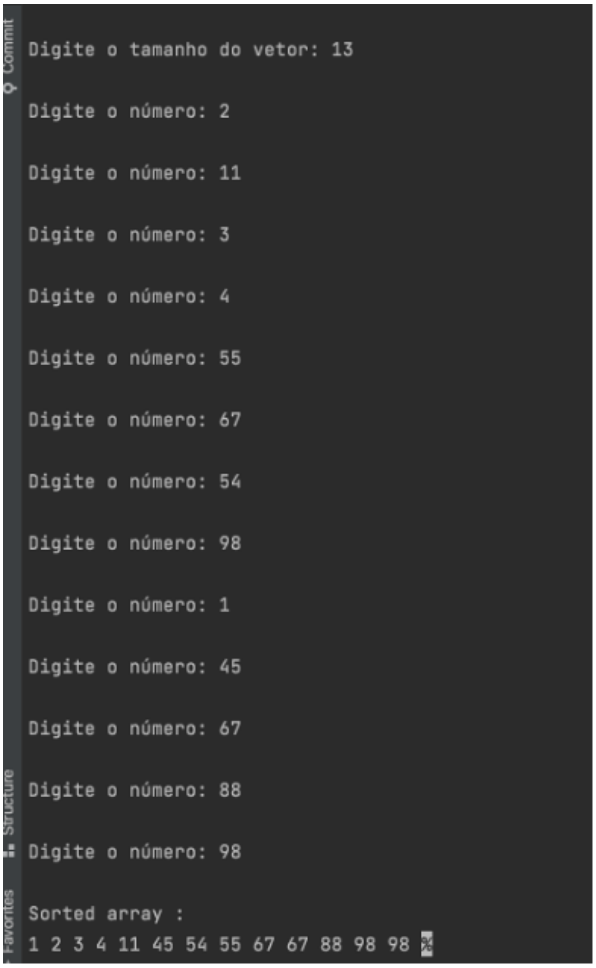


Selection Sort

Para implementar o SelectionSort de modo paralelo, foi criado uma Struct para armazenar o índice do vetor e o menor elemento. Depois foi usado o reduction customizado para manter uma cópia privada do resultado do menor valor para cada execução. A variável privada que que faz com que isso seja possível foi declarada na função ‘selectionSort’. Dessa forma, foi evitado que o código funcionasse de modo sequencial.

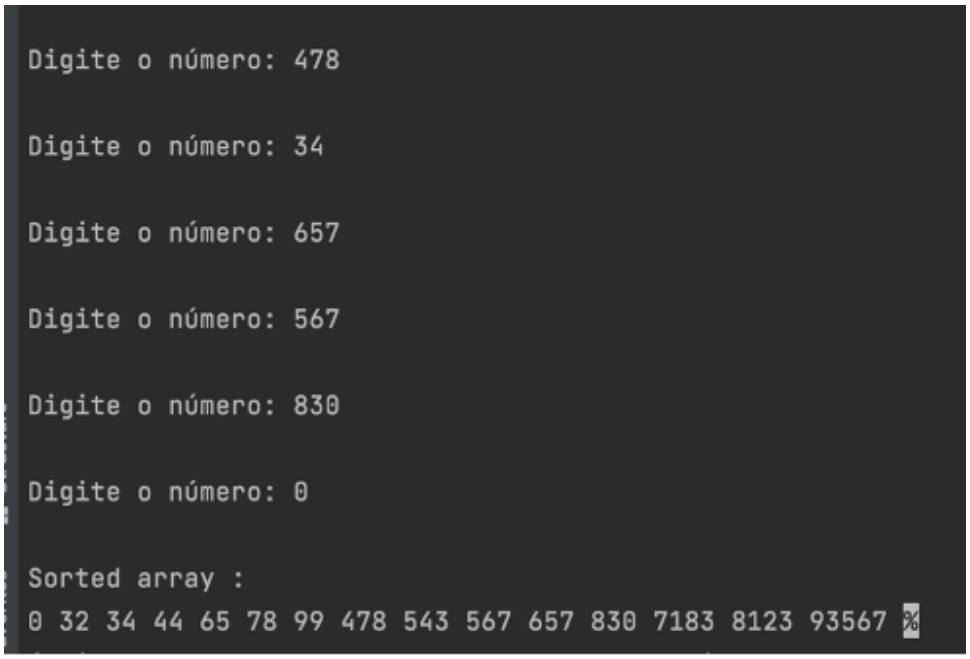

jenifer.santos@MacBook-Pro-de-Zupper selectionsort % gcc -o selectionsort SelectionSort.c
jenifer.santos@MacBook-Pro-de-Zupper selectionsort % ./selectionsort
Digite o tamanho do vetor: 12
Digite o número: 4
Digite o número: 3
Digite o número: 76
Digite o número: 54
Digite o número: 33
Digite o número: 2
Digite o número: 8
Digite o número: 12
Digite o número: 56
Digite o número: 789
Digite o número: 0
Digite o número: 24
Sorted array :
0 2 3 4 8 12 24 33 54 56 76 789


jenifer.santos@MacBook-Pro-de-Zupper selectionsort % ./selectionsort
Digite o tamanho do vetor: 10
Digite o número: 45
Digite o número: 65
Digite o número: 66
Digite o número: 78
Digite o número: 22
Digite o número: 34
Digite o número: 5
Digite o número: 61
Digite o número: 92
Digite o número: 89
Sorted array :
5 22 34 45 61 65 66 78 89 92 %


Commit
Structure
Favorites
Digite o tamanho do vetor: 13
Digite o número: 2
Digite o número: 11
Digite o número: 3
Digite o número: 4
Digite o número: 55
Digite o número: 67
Digite o número: 54
Digite o número: 98
Digite o número: 1
Digite o número: 45
Digite o número: 67
Digite o número: 88
Digite o número: 98
Sorted array :
1 2 3 4 11 45 54 55 67 67 88 98 98

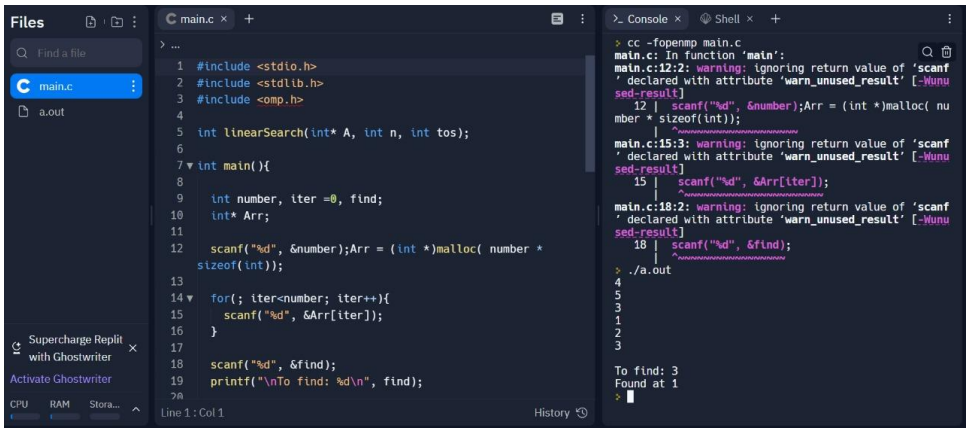
```
Digite o tamanho do vetor: 9

Digite o número: 35

Digite o número: 62

Digite o número: 12

Digite o número: 14

Digite o número: 45

Digite o número: 65

Digite o número: 87

Digite o número: 79

Digite o número: 90

Sorted array :
12 14 35 45 62 65 79 87 90 %
```

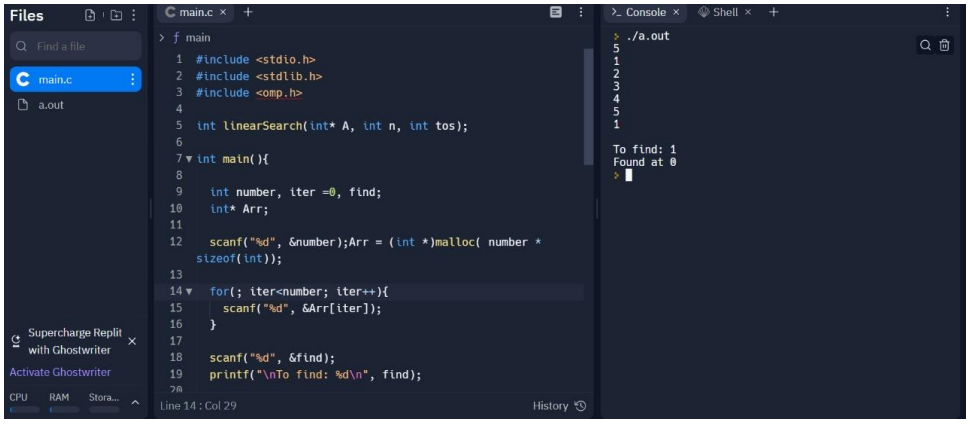
Continuação 5° vetor



Linear Search

Para implementar o Linear Search de modo paralelo, usado o openmp, dessa vez para encontrar o valor requerido, percorrendo cada item do vetor. Foi usado o for de forma paralela # pragma omp parallel for, para que o processo de busca ocorresse em um tempo mais rápido do que fazendo uma busca linear sem usar paralelismo.





UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática