

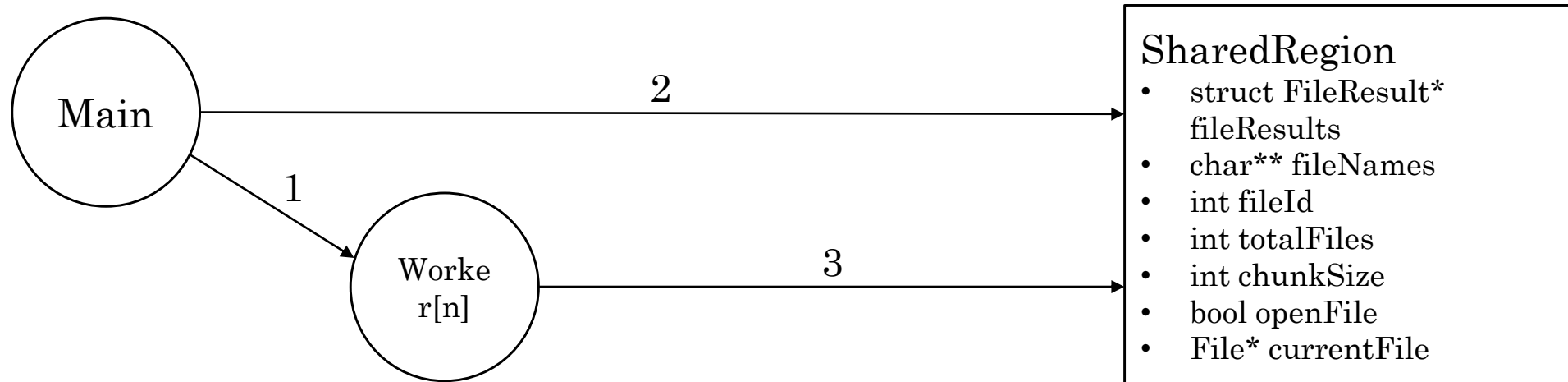
# Assignment 1

João Leite, nº 115041

Luís Batista, nº 115279

# Problem 1

## Entities diagram



### Legend

1

- init and start workers

2

- fillSharedMem – fill shared memory with file names; initialize remaining entities with default values
- printResults – print all obtained file results

3

- requestChunk – request a chunk of text to be processed
- postResults – post the obtained results in the shared region

### FileResult

- int nWords
- int vowels[6]
- char\* fileName

# Problem 1

## Results

### PC1 Specs:

Lubuntu VM  
8 cores

1 worker		2 workers		4 workers		8 workers	
Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)
0.009220	0.000396	0.007389	0.000952	0.008853	0.001383	0.008420	0.000215

### PC2 Specs:

Ubuntu VM  
4 cores

1 worker		2 workers		4 workers		8 workers	
Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)
0.026864	0.012453	0.020446	0.005223	0.024906	0.007404	0.026550	0.015925

*(both mean and standard deviation results were calculated by running the program 5 times)*

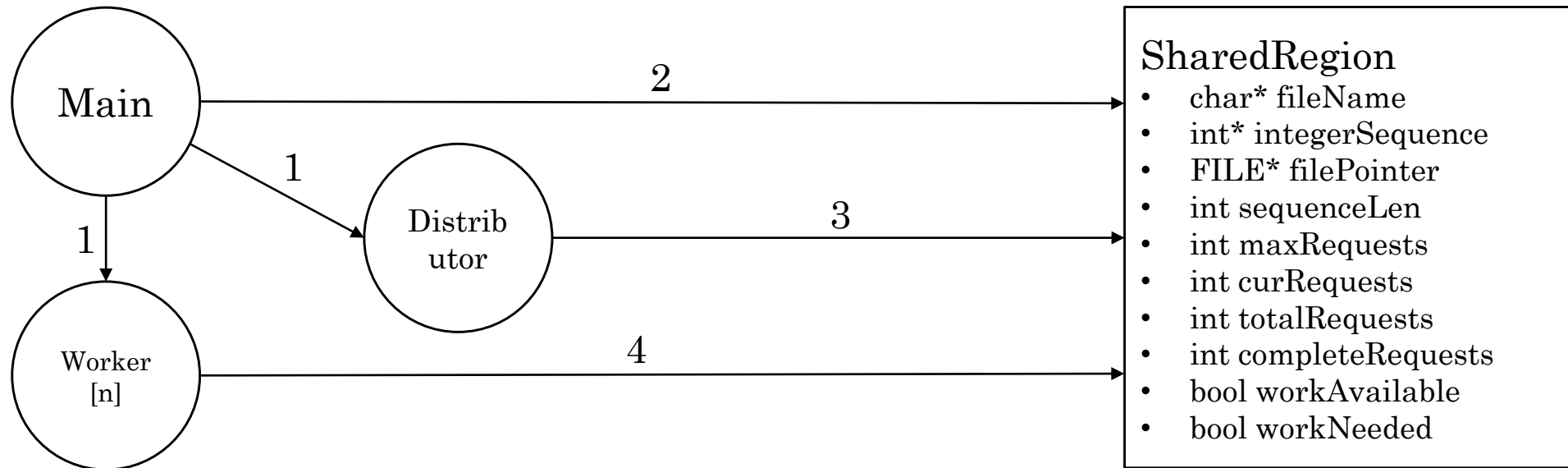
# Problem 1

## Conclusions

- By looking at the obtained results it possible to take some conclusions:
  - In both PCs, there is a small gain when using 2 threads comparing to 1 thread only.
  - From two threads upwards and **given the input files used**, the program execution time does not benefit at all of being run using more threads.
  - By the low values of the standard deviation we can perceive that all program's executions follow the same conditions (it means that there were no big changes in terms of available hardware/software). However, PC1 presents lower deviation values, which may be related to the higher specifications given to the VM when comparing with the ones given to the VM of PC2.
- When looking at these conclusions we can't forget the environment at which the program ran: a virtual machine which applies an "additional layer" between the operating system and the program execution.

# Problem 2

## Entities diagram



### Legend

- 1
  - init and start thread
- 2
  - fillFileName – fill shared memory with file name containing the integer sequence
  - validateArray – validate if the obtained array is sorted

- 3
  - readIntegerSequence – fill shared memory with the integer array
  - assignWork – assign work to the worker threads
- 4
  - requestWork – request integer sequence to sort
  - informWork – inform that the assigned sequence was sorted

# Problem 2

## Results

### PC1 Specs:

Lubuntu VM  
8 cores

Binary file	1 worker		2 workers		4 workers		8 workers	
	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)
datSeq32.bin	0.001535	0.000242	0.0018208	0.000415	0.0019512	0.000361	0.0022976	0.000405
datSeq256K.bin	0.277978	0.015954	0.2623584	0.005271	0.248080	0.005687	0.251595	0.005165
datSeq1M.bin	1.163098	0.040732	1.066309	0.020666	1.010236	0.020289	1.004948	0.026401
datSeq16M.bin	19.756249	0.235015	17.782816	0.483265	16.933987	0.135016	16.660244	0.193803

### PC2 Specs:

Ubuntu VM  
4 cores

Binary file	1 worker		2 workers		4 workers		8 workers	
	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)	Mean (s)	Standard deviation (s)
datSeq32.bin	0.002775	0.000770	0.005476	0.004615	0.008056	0.009762	0.007601	0.007706
datSeq256K.bin	0.299569	0.030652	0.270263	0.023774	0.254095	0.028820	0.287229	0.047620
datSeq1M.bin	0.994728	0.0303520	0.912594	0.0615767	0.919737	0.044286	0.943953	0.0380660
datSeq16M.bin	16.101655	0.178125	12.858210	0.186560	12.686568	0.281215	12.424148	0.316890

*(both mean and standard deviation results were calculated by running the program 5 times)*

# Problem 2

## Conclusions

- In the second problem, specially in the larger file, there is a much clear difference when running the program using a different number of threads.
- For the smaller files (*datSeq32.bin* and *datSeq256K.bin*) the program behaviour remains similar to what was observed in the first problem, as the dataset is not big enough to make a significant difference.
- The last two files (*datSeq1M.bin* and *datSeq16M.bin*), however, showed some improvement when using more threads, up to almost 4 seconds in the 16M file.
- By the low values of the standard deviation we can perceive that all program's executions follow the same conditions (it means that there were not big changes in terms of available hardware/software). However, PC1 typically presents lower deviation values which may be related to the higher specifications given to the VM when comparing with the ones given to the VM of PC2.
- When looking at these conclusions we can't forget the environment at which the program ran: a virtual machine which applies an "additional layer" between the operating system and the program execution.