Supermarket inventory

Product

barCode → int

* used as an ID *

name → String

* the name of the product *

price → double

Bill

name → String

* a string that contains the number of the bill, helps to see the number of the thread *

totalPrice → double

* the total of the bill *

inventoryChanged → Boolean

* used to mark if something changed in the inventory *

inventory → Inventory

__mutex → Lock

Inventory

products → Hashtable <Product, Integer>

* contains all the products along with the existent quantity *

A **bill** was created for each **thread**. In each **bill** a random number (between 1 and 10) of **products** of random **quantity** (between 1 and 10) were added.

An array was created containing each thread. All of the threads were started and joined together.

After finishing the threads, the Inventory must be checked to see if the number of products is not negative or to see if the products were not bought (if the program didn't change the inventory at all).

Granularity 1: everything will be **locked** until one bill finishes to take all the items needed.

(another thread starts after the one before finishes)

Granularity 2: just one product at a time is **locked** and after that it will **unlock**.

Tests granularity 1:

- 10 threads → 0.064 seconds
- 100 threads → 0.128 seconds
- 1000 threads → 0.384 seconds

Tests granularity 2:

- 10 threads → 0.64 seconds
- 100 threads → 0.128 seconds
- 1000 threads → 0.256 seconds