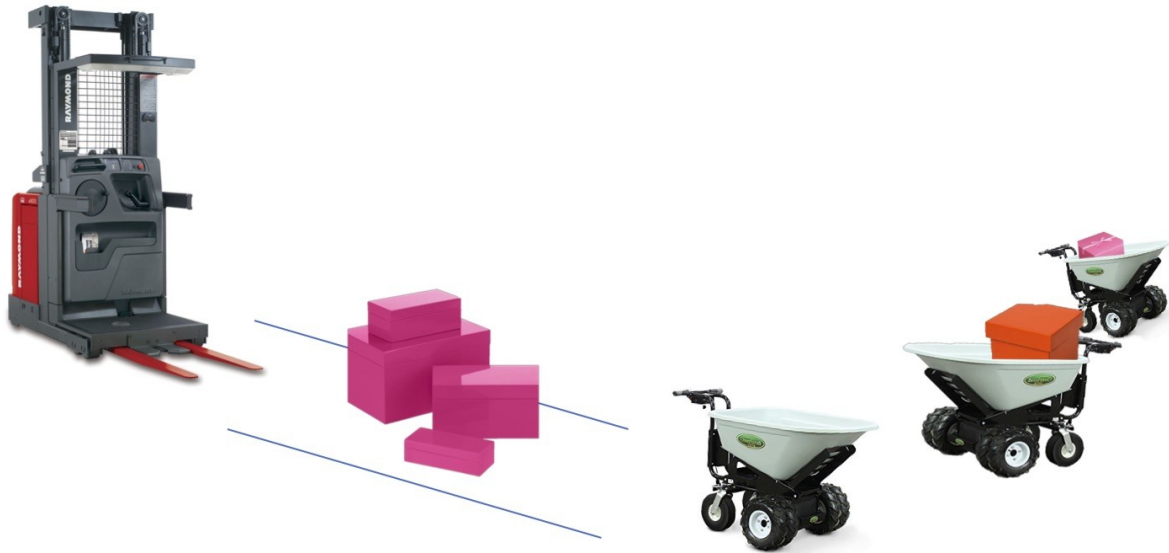


## Lab assignment #1: the forklift and the carts

Consider a box factory, where a robotic forklift waits for motorized carts to bring boxes that the forklift then has to place. Motorized carts can unload their boxes to the floor in a narrow alley near the forklift. However, the narrow alley can only accommodate 5 boxes at a time.



Assume that the alley is initially empty (so, there's room for 5 boxes). The forklift's behavior is simple: as soon as there's a box to place, the fork lift takes a box and goes to place it; after placing the box, the fork lift returns to the alley.

A motorized cart's behavior is slightly more complicated, because the cart will have to decide whether the box can be unloaded on the alley floor, or should be returned:

1. find a box to pick up and bring the box to the alley;
2. if there is room on the floor, unload the box, otherwise bring back the box to its initial location;
3. repeat

Use `pthread`s to activate one forklift thread and 10 cart threads. Ensure a proper synchronization using **semaphores**.

To simulate actions such as taking a box, placing a box, etc, use `sleep( n )` where `n` is the number of seconds the action takes. To spice things up, you can use a function such as `rand()` to generate a random number of seconds an action should take.

Simulate for a length of time, say a couple of minutes. In the end, take note of how many boxes have been placed and how many have been returned.

Provide **working code** (meaning, code that compiles) **and a text file** commenting your solution (and reporting on the numbers of boxes that have been placed or returned).

You can start from [this template](#) (but are not obliged to).

Bear in mind that:

1. even if your solution is not perfect, anything is better than nothing at all;
2. if any aspects of this assignment are unclear, you are more than welcome to ask for clarifications to the teaching staff;
3. you can use the wiki to discuss or share interesting things you discover, but not to share your solution;
4. you can either work individually, or, if you prefer, you can work in a group of 2/3 people, as long as you contribute with your share of work;
5. if your solution is the result of team work, for grading purposes, you should indicate the names of all the members of the team;
6. the point of this exercise is to give you a reason to think about synchronization problems and semaphores, as well as practice with threads and with looking for useful functions in the online manual. If you take this as a challenge, you will ultimately benefit, and chances are you will join the exam with good confidence. If you wish to master semaphore-based synchronization, there's nothing like making your hands dirty with some coding!