

HPPS assignment 5

August Pallesen, Sky Wong og Alexander Husted

Januar 2023

Indhold

1	Introduction	2
2	The Naive Solution	2
3	The Socketserver Solution	2
4	The True Solution	3

1 Introduction

The things written in the introduction holds for all of the three programs

We have removed the checkin functions after we tested our solution with the shared.py file. We see from the shared.py output that there is 3 elf's between Santa does an appearance or 8 reindeers. (If of course we start the program with the condition given in the assignment)

2 The Naive Solution

As per the hint, the solution to the elves is similar to the reindeer. From reindeer we pretty much have all we need for elves. We just changed the variable names from reindeer to elf, as well as changing the *if*-statement to equal 3 elves. The message being sent also got changed from *MSG_DELIVER_PRESENTS* to *MSG_SORT_PROBLEM*.

3 The Socketserver Solution

Deadlock

Deadlock in programming refers to a specific situation where two or more processes are unable to proceed because each is waiting for the other to release resources.

- We have potential deadlock where a process is holding at least one resource and waiting for resources: When a reindeer returns from holiday and is waiting for Santa, but Santa is waiting for all the reindeer's to return. But we ensure that the last reindeer always wakes up Santa, and thus avoiding the problem.
- When a connection is established between two sockets, no other process can use the same connection simultaneously: That is why we close the socket immediately after opening it.

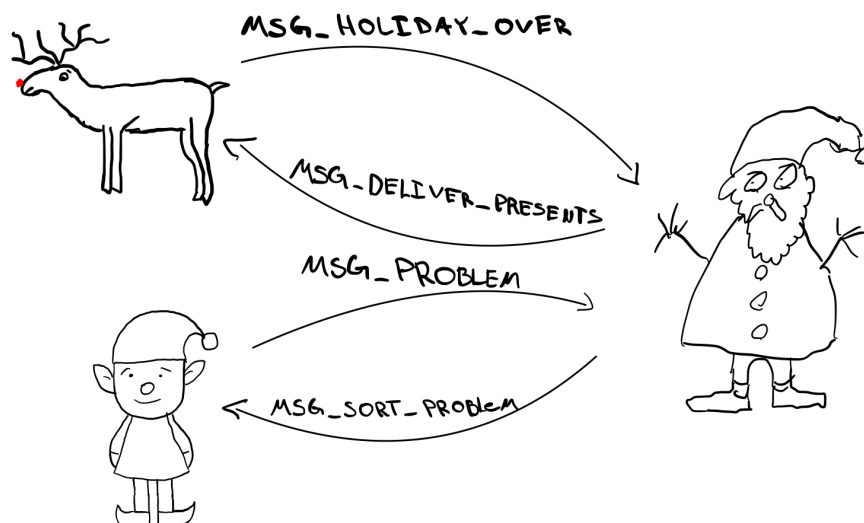
Race conditions

- The use of the lock function from the threading library ensuring that only one thread can access the handler function at a time, which makes sure that we don't have multiple threads in play which may cause race conditions.

Does it work?

We used the terminal output with temporary prints statements which includes:

- The list of elf to check that it is emptied after the elf received a message from Santa.
- The count of reindeer which should go from the number of reindeers to zero when reviving a message from Santa.



4 The True Solution

We reduce the amount of messages sent to Santa, which reduces the change of two programs messaging Santa at the same time. The deadlock and race condition are the same as the socket santa problem.

Does it work?

We started by testing the relationship between reindeer and stable, this relationship functioned correctly with the reindeers going on holiday after receiving MSG_HOLIDAY_OVER when all reindeers are in the stable. We did the same thing for elf's, and appended Santa to the list of elf/reindeer that should receive a message. As shown in the figure below:

