Universitetet i Oslo
Institutt for Informatikk
PMA
Olaf Owe, Martin Steffen

INF 4140: Models of Concurrency

Høst 2015 — **Mandatory assignment 3** — 15. Nov 2015

Issued: 15. Nov 2015

Due: 27. November 2015

1 General remarks

How to deliver

- Your solution should be delivered online (https://devilry.ifi.uio.no/).
- Program examples should be commented in order to make them understandable for the group teacher or lecturer.

Who delivers

We encourage to work together in groups of 2. However, you are not allowed to solve the oblig with more than one partner. For "technical" reasons (devilry): in a group of 2 people, *both* must upload the solution (which should be identical, just the same PDF uploaded twice).[1] The solution must be marked with name(s) and email address(es) of the contributing student(s).

If in doubt, you may also read the departmental guidelines for written assignments before you start.

Check in time that devilry works and that your status within devilry (and student web etc) is ok. Don't try your INF4240-devilry access as late as the day of the deadline. In case of doubts, for clarifications, or if having trouble with devilry etc, ask us in time.

Evaluation

This assignment is graded *pass* or *fail*. You must pass the obligs in order to take the final exam.

[1] That facilitates managing acceptance.

Dining philosophers implemented with message-passing

Consider an approach to the dining philosophers based on asynchronous message passing.

1. You can assume that there are `N` philosophers around a table, and one fork between each pair of philosophers (ref. fig. 4.6, p. 165 in Andrews). Implement a solution where both philosophers and forks are represented as processes. A philosopher picks up the forks next to himself by sending a request to the neighbor-fork-processes, asking for access to the fork. When a philosopher is finished eating, the forks are released by passing a message to the two fork-processes. Let each fork listen to a (private) single channel `request` and each philosopher listen to a (private) single channel `reply`. Is your solution "fair"? Is there a possibility for deadlock? Explain your answer.

2. An alternative solution to the dining philosophers can be acquired by letting the forks become a shared resource for all the philosophers. This can be achieved by letting the forks be handed out by a butler process. Implement a fair and deadlock-free solution where the philosophers requests the forks from a butler. You may assume that there is `N` philosophers and `M` forks. You may also assume that a philosopher who gains access to the forks always will release them. Explain why your solution is fair and deadlock free.