



Distributed Systems Part II

Exercise Sheet 2

1 Communication without Computers

Consider the actions described below, in which model (shared memory or message passing) can you describe them best? Why and how?

- Communication via postcards
- Two people speaking in a room
- Instant messages via Skype (data remains on client if partner is offline)
- Many people speaking in a room

2 Consensus with the Aid of a Wall

Alice and Bob live in the same town. Once a year they want to meet but they don't want to be seen together in public. So they want to meet at a secret place which one of them chooses. They know a wall in town which is painted white. In addition they know a painter who paints the wall in the color they wish and sends the person who gave him the order a "before and after" picture of the wall. (Of course they color-coded each possible meeting place in advance.)

- a) Design an algorithm which ensures that Alice and Bob meet at the same place
- b) Can you expand your algorithm in such a way that it still works if Charlie wants to meet them as well?
- c) How many persons can meet each other, if the wall is in front of the painters' shop and why? (You can assume that the painter immediately starts painting after receiving an order)

3 Consensus through "Fetch and Multiply"

A friend of yours is convinced to have found a great algorithm to find consensus for 13 processes. His algorithm relies on a method called "Fetch and Multiply" which is described below. What would you tell him, if he asked you for your opinion?

```
public class RMW {
    private int value ;
    public synchronized int FAM(int x) {
        int prior = this.value ;
        this.value = this.value * x ;
        return prior ;
    }
}
```

4 Consensus for two Processes

A lousy programmer wanted to solve consensus for 2 processes and came up with a sophisticated protocol. Does the protocol really solve consensus? Simplify the protocol and then prove or reason about your claim. If the protocol does not reach consensus, provide a counterexample. If it does, draw the important part of the execution tree.

```
initialize(){
    // s and i are shared
    s = '?';
    i = 0;

    // the input, an array of length 2
    input[] = [random({0,1}), random({0,1})];
}

// making the decision
decide(){
    // the id of this process, 0 or 1
    id = this.getThreadId();

    decisionMade = false;
    decision = input[id];

    while(decisionMade == false){
        value = s; // read s
        if(value == '?'){
            s = input[id]; // write s
        }
        else if(value != decision){
            decisionMade = true;
            decision = value;
        }
        else{
            if(i.fetchAndInc() == 1){
                decision = input[ 1-id ];
            }
            decisionMade = true;
        }
    }
}
```