UNIVERSITY OF TARTU
Institute of Computer Science

Distributed Systems (Spring 2020)

Task 3: Naming, election algorithms and replication

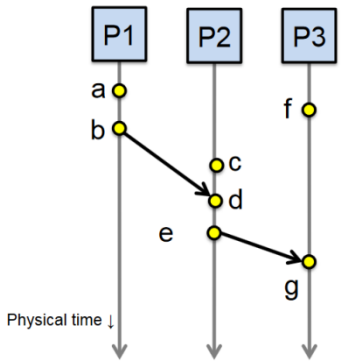
Practical information:
Due date: Monday, May 11, noon
– This task can be done in a team of max 3 persons ONLY.

Submit your solution to mohan.liyanage@ut.ee, (CC) huber.flores@ut.ee

Instructions: Please respond the following questions. Be clear and concise in your answers. Provide enough explanation to support your arguments. **Ambiguous answers to fill up space are considered wrong and no points are granted**.

1) Given the following processes, P1, P2, and P3. Complete the missing column in the table of concurrent events. Explain your reasoning (5 pts)

Operations	Concurrent?
a, b	
b, f	
c, f	
e, f	
e, g	
a, c	
a, e	



2) Assuming that all variables are initially set to 0. For the following executions, please indicate if they are sequentially consistent. Explain in detail your answer. If an execution is not sequentially consistent, then show the right execution that makes it sequentially consistent (explain in detail how you achieved that). (15 pts)

1. P1: W(x) 1
   P2: R(x) 0; R(x) 1

2. P1: W(x) 1
   P2: R(x) 1; R(x) 0

3. P1: W(x) 1
   P2: W(x) 2
   P3: R(x) 1; R(x) 2

4. P1: W(x) 1
   P2: W(x) 2
   P3: R(x) 2; R(x) 1

5. P1: W(x) 1
   P2: W(x) 2
   P3: R(x) 2; R(x) 1
   P4: R(x) 1; R(x) 2

6. P1: W(x) 1; R(x) 1; R(y) 0
   P2: W(y) 1; R(y) 1; R(x) 1
   P3: R(x) 1; R(y) 0
   P4: R(y) 0; R(x) 0

7. P1: W(x) 1; R(x) 1; R(y) 1
   P2: W(y) 1; R(y) 1; R(x) 1
   P3: R(y) 1; R(x) 0

3) In this problem, we use the useful **dig tool** available on Unix and Linux hosts to explore the hierarchy of DNS servers. Recall that in slide 51 from Naming lecture, a DNS server higher in the DNS hierarchy delegates a DNS query to a DNS server lower in the hierarchy, by sending back to the DNS client the name of that lower--level DNS server. First read the man page for dig (e.g., http://linux.die.net/man/1/dig), and then answer the following questions. (15 points)

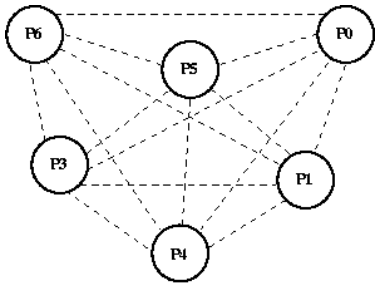
   a. Starting with a root DNS server (from one of the root servers [a--m].root--servers.net], initiate a sequence of queries for the IP address for our computer science department's Web server (www.cs.ut.ee) by using dig. Show the list of the names of DNS servers in the delegation chain in answering your query.
   b. Repeat part a) for several popular Web sites (at least 5 more), such as google.com, facebook.com, or amazon.com. Back up your answers with screen shots that show the results of your dig queries, and explain in detail the look up process.

4) Consider a network of 3 nodes (A,B and C) and a single, shared resource (e.g. a printer). Describe for each of the following algorithms how A can gain exclusive access to the resource and enumerate the necessary message exchanges (Provide draws to support your answers). In addition, state (at least) one major benefit and one major drawback of each individual algorithm. (10 pts)

   - Centralized coordinator based algorithm - B is the coordinator
   - Decentralized approach(voting)
   - Distributed algorithm(Ricart & Agrawala)
   - Token-Ring algorithm

5) Resolve using the bullying algorithm. The figure shows 6 processes, all directly connected to each other. Process 6 is the leader, as it has the highest number. Assume process 6 fails. (10 pts)
   a. What is the number of messages required to elect a new leader
   b. Assume that process 6 comes back to the system. Describe the process that leads process 6 to become the leader again

c. Assuming that the coordinator fails, generalize a formula that estimates the number of messages required in a network of N processes to select a new coordinator.



Bully Algorithm: Step 0

6. Write a program that implements the bully algorithm. (45 pts)
Specifications
Command line interface:

$ bully_program ***processes_file***

Where,

1. *bully_program* is your program
2. ***processes_file*** is the path to the file where processes are defined.

The processes_file has N number of lines, each in the following format separated by commas.
Id, name
1, A_K
2, B_K
4, D_K
3, E_K
5, F_K

Processes defined in the processes_file have unique Ids. When running the processes_file, then the processes are loaded into the program, and the bully algorithm is applied. Initially, any process can be the coordinator, but then after the bully algorithm is applied, the process with the highest ID must be elected as the coordinator. An automatic notification should announce the process that is the coordinator after the processes are loaded. When an election takes place, each process NAME is updated. This means that the K value should be increased by 1 each time and election takes place. K is equal to zero when loading the processes. So, the actual processes_file input looks like this (5 pts)

1, A_0
2, B_0
4, D_0
3, E_0
5, F_0

In addition, the program should be running continuously, such that this input commands can be introduced via command line

1- list command ($ list), this command shows the list of processes running, and indicate the process that is acting as coordinator (4 pts)
2- Killing a process ($ kill Id), this command should allow removing a process from the program (including the coordinator). If the coordinator is deleted, a new election needs to happen automatically (without requesting it). The program should notify the process that is the new coordinator, and which process started the election (8 pts)
3- Reload processes ($ reload), this command reloads the processes defined in the file (processes_file) to the running program. If a process already exists, then the process should not be reloaded. If a process was removed (killed), then the process is reloaded again into the program. Once processes are reloaded, a new election should take place. Notice also that new processes can be defined in the input file. Moreover, notice also that the K value of each process should reflect the number of elections in which a process has participated. For instance, if A_0 and B_0 were loaded initially, and participated in two elections, then their current status is A_2 and B_2, respectively. If the process A is then removed, and then there is a new election, the value of B then becomes B_3. If the process A is reloaded (after reload command is applied), and then there is a new election, A is back to the processes pool, and their current values for processes A and B should be A_0 and B_4, respectively. (20 pts)
4- Take performance measurements about the election time of the algorithm. Test your algorithm when processes are increased each time by 10 within an interval of (0,100). Meaning that initially the election will happen with 10 processes, then 20 processes, 30 processes, and so on until 100.(8 pts)

Deliverables

- Source code and instructions
- Video explaining your program and showing a simple example
- A nice bar/lines graph that shows how the election time increases as the number of processes also increase. (Y axis = election time, X axis = number of processes)