01_HW_06

March 23, 2022

1 Distributed Systems (LTAT.06.007)

1.0.1 Seminar 6: Bullying Algorithm in python

Student : ChengHan Chung

1.1 1. Task-1:

In **Seminar 4**, we have learnt how to implement Berkeley Algorithm for clock synchronization based on the pre-assigned and fixed coordinator. In this similar, you should achieve the Bully Algorithm based on `clock-sync.py` to select the coordinator dynamically. The code for the above implementation is given here for you.

1.1 For implementation `Bully Algorithm`, code and result shown as follwing below:

```python
def coordinator_election(processes):
    highest_id = max([p.id for p in processes])
    for p in processes:
        if p.id == highest_id:
            p.label = 'C'
        else:
            p.label = 'S'
```

```
utlab@DESKTOP-SO1IIEA:2022_seminar6$ python3 clock-sync.py bully.txt
[WARNING] Duplicate procees id 5 discarded
Commands: exit, list, clock, kill
list
1, A_0
3, B_0
4, D_0
7, E_0 (Coordinator)
5, F_0
clock
A_0, 23:00:12
B_0, 23:00:12
D_0, 23:00:12
E_0, 23:00:12
F_0, 23:00:12
kill 7
list
1, A_0
3, B_0
4, D_0
5, F_0 (Coordinator)
clock
A_0, 03:00:53
B_0, 03:00:53
D_0, 03:00:53
F_0, 03:00:53
exit
Program exited
```

1.2   2. Task-2:

In **Seminar 4**, we already practice to implement Berkeley algorithm by using RPyC Threaded-Server in python. For this task:

1. Integrate the bully function of **Task-1** into RPyC **server communiction**.
2. Pass **arguments** one by one ("bully.txt" and different commands) from client to server.

2.1 The result shonw as following below:

Client side

```
utlab@DESKTOP-SO1IIEA:06_Bullying_Algorithm$ python3 rpyc_client.py localhost
Input the file name: bully.txt
Input the command: list
Input the command: clock
Input the command: kill 7
Input the command: list
Input the command: clock
Input the command: exit
```

Server sdie

```
utlab@DESKTOP-SO1IIEA:06_Bullying_Algorithm$ python3 rpyc_server.py

connected on 2022-03-23 16:41:36.471778
Receive file name from client: bully.txt
[WARNING] Duplicate procees id 5 discarded
Commands: exit, list, clock, kill
Receive command from client: list
1, A_0
3, B_0
4, D_0
7, E_0, (Coordinator)
5, F_0
Receive command from client: clock
A_0, 23:00:00
B_0, 23:00:00
D_0, 23:00:00
E_0, 23:00:00
F_0, 23:00:00
Receive command from client: kill 7
Receive command from client: list
1, A_0
3, B_0
4, D_0
5, F_0, (Coordinator)
Receive command from client: clock
A_0, 23:00:00
B_0, 23:00:00
D_0, 23:00:00
F_0, 23:00:00
Receive command from client: exit
Program exited
```