

Mini project : Simple Line-like app

Group Name : Deep system

Group members

Terapap Apiparakoon 5730279821

Sasicha Poonpol 5731101921

Weeraphat Jungsomjatepaisal 573110221

Chiran Bawornkitiwong 5731018521

Pitchanai Thitipakorn 5730400321

2110318 Distributed System Essential
Faculty of Computer Engineering
Chulalongkorn University

Video-URL : <https://www.youtube.com/watch?v=1ZncfcPys8M>

ภาพรวมของระบบ (architectural model)

ภายในระบบจะประกอบไปด้วย

1. Main server จำนวน 1 ตัว
2. Backup server จำนวน 1 ตัว
3. Client จำนวน 5 ตัว (ตามจำนวนสมาชิกในกลุ่ม)

โดยที่ client จะมีความสัมพันธ์กับ server ทั้ง 2 ตัวนั่นคือ client แต่ละตัวจะทำการติดต่อสื่อสารกับผ่านทาง main server และตัว main server จะส่งข้อมูลไป update ที่ backup server ทำให้ข้อมูลที่ client ติดต่อสื่อสารกันมีการ backup เก็บไว้ที่ backup server หาก main server fail ตัว main server ก็จะส่งข้อความไปหา client เพื่อให้ client ทำการ redirect ไปที่ตัว backup server แทน แล้วตัว backup server ก็จะทำงานแทนตัว main server เดิมที่มีความสามารถในการทำงานเหมือน main server ทุกประการ

วิธีการของ chat server ในการทำ total ordering

ในการติดต่อสื่อสารกันของ client นั้นจะมีการส่งข้อความผ่านตัว server แล้ว server จะทำหน้าที่ส่งต่อข้อมูลไปให้กับตัว client ตัวอื่นๆ ภายในกลุ่มเดียวกัน สำหรับการส่งข้อมูลของ client ไปยัง server นั้น ข้อความที่ client ส่งไปให้ server จะมี pattern ของการส่งเพื่อให้ server รับรู้ว่าข้อความที่ส่งมานั้นเป็นข้อความที่จะส่งให้กับ client ตัวอื่นๆ หรือเป็นข้อความที่บอกสถานะให้กับตัว server โดยจะมี method run() ที่จะคอยรับข้อมูลที่เข้ามาของ client ตลอดเวลา

วิธีการของ client ในการแสดงผลข้อความตามลำดับ total ordering

ข้อความที่ client ส่งเข้ามาใน server จะถูกประมวลผลตามลำดับนั้น หมายความว่าข้อความใดที่ client ส่งถึง server ก่อนก็จะถูกประมวลผลก่อน จากนั้นก็จะกระจายของข้อความไปยัง client ตัวอื่นๆก่อนที่จะมีการแสดงผลข้อความ มีผลทำให้ข้อความที่ส่งมาที่หลังถูกประมวลผลหลังจากที่ข้อความก่อนหน้าแสดงผลออกที่แล้ว (ส่งก่อนแสดงผลก่อน)

วิธี update สถานะของ chat server ตัวรอง

ในขณะที่ server ตัวหลักทำงานอยู่ ข้อความที่ส่งจาก main server มายัง backup server จะถูกส่งเข้าเพื่อ update สถานะของ client เพียงอย่างเดียว แต่ถ้าหาก main server เกิด fail ขึ้น การทำงานของ backup server จะเปลี่ยนไป โดยข้อความที่ส่งมาจาก client จะเปลี่ยนจากการ update เพียงอย่างเดียวเป็นทั้งประมวลผลและupdateด้วย ซึ่งจะทำหน้าที่ได้เหมือนกับ main server

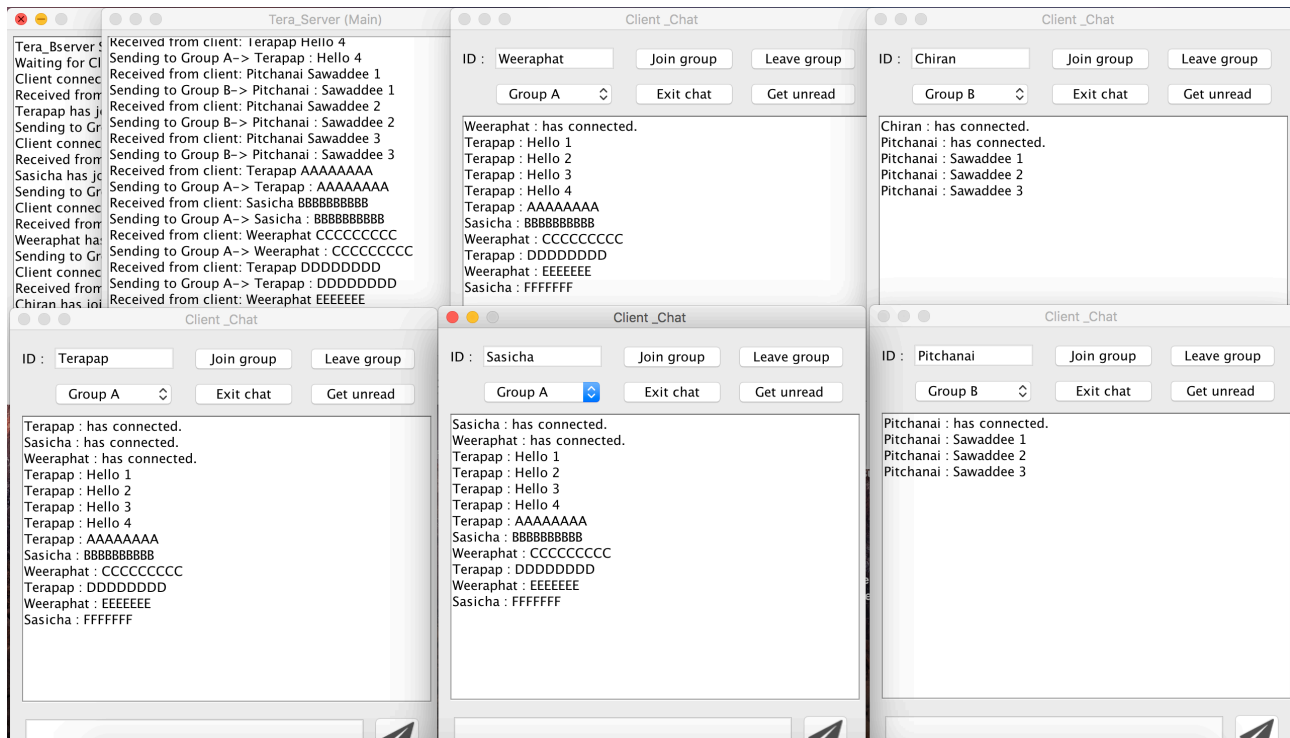
วิธีการ takeover ของ chatserver ตัวรองแทนตัวหลัก

เมื่อ main server เกิด fail ขึ้น main server จะส่งข้อความไปหา client ทุกตัวเพื่อให้เปลี่ยนการติดต่อไปยัง backup server แทน จากนั้น main server ก็จะส่งข้อความไปบอก backup server ว่าให้ทำหน้าที่แทน main server เมื่อ backup server ได้รับข้อความจาก main server แล้ว backup server ก็จะเปลี่ยนสถานะตัวเองให้ทั้งทำการประมวลผลและ update status ไปด้วย

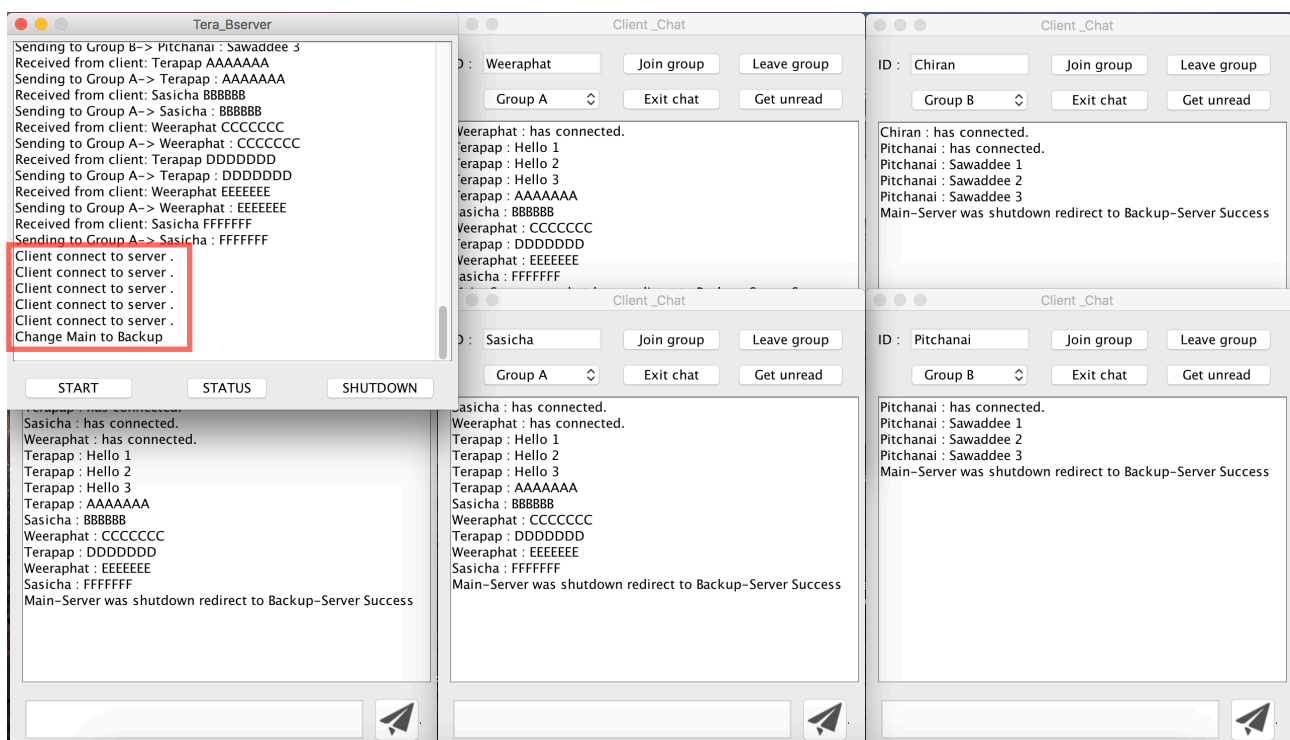
การติดต่อของ client ไปยัง server ตัวรอง

ในที่นี้เราให้ main server มี port 6969 และ backup server มี port 8888 เมื่อ client ได้รับข้อความจาก main server ตัว client ก็จะเปลี่ยน socket และ port ไปยังเครื่อง backup server แทน ทำให้ client ทุกตัวยังคงสามารถติดต่อสื่อสารกันผ่านทาง backup server ได้

ภาพหน้าจอ client ID ทั้งหมดซึ่งแสดงผลข้อความในลำดับเดียวกันก่อนจะมีการ exit

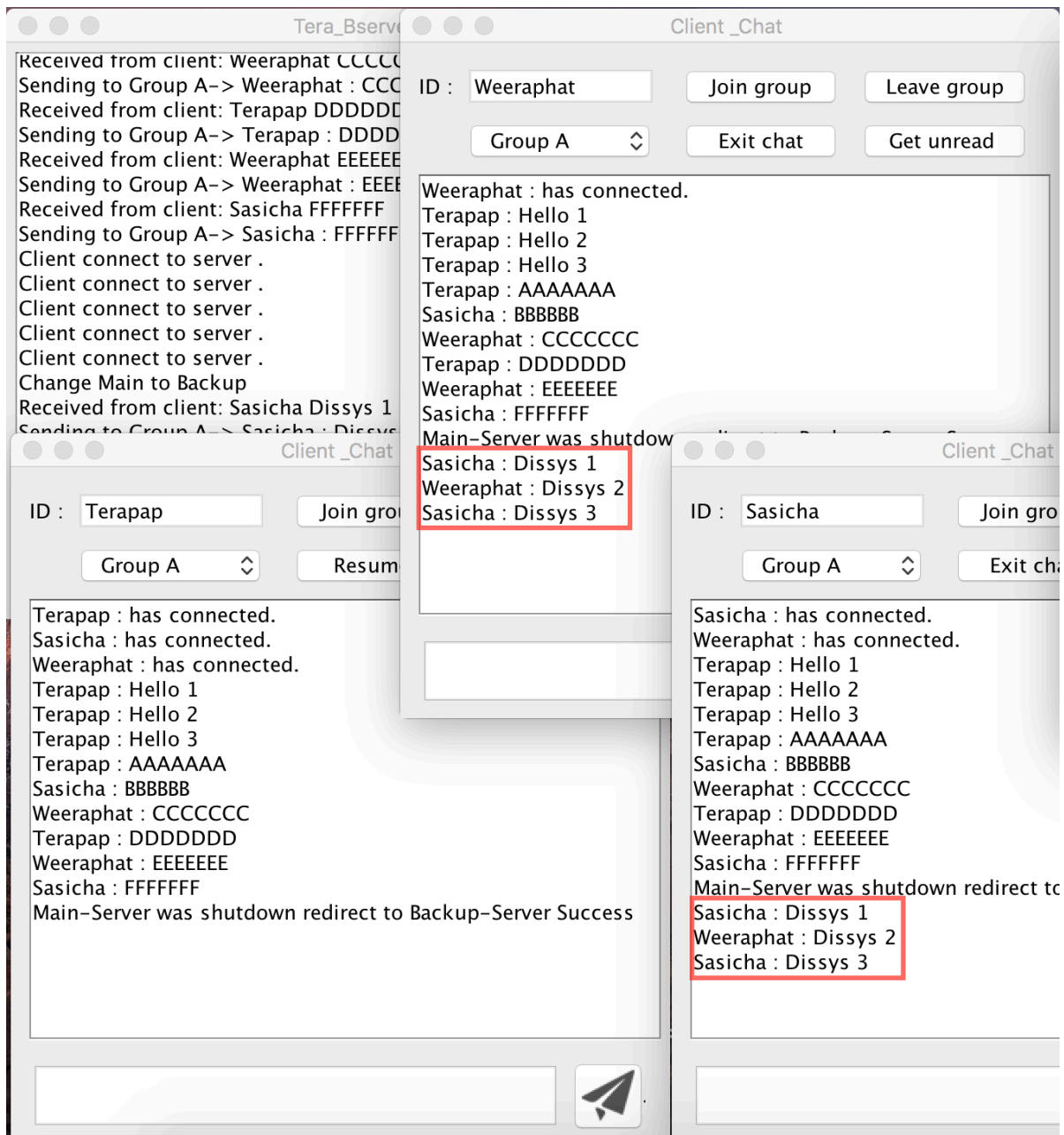


ภาพหน้าจอสถานะของตัวรองว่าพร้อมทำงานแทนหลัง takeover



ภาพหน้าจอ client ID ทั้งหมดซึ่งแสดงผลข้อความในลำดับเดียวกัน

ตอนที่ client exit ไป สถานะปุ่มจาก Exit chat เป็น Resume เพื่อรอการกลับมาของ client



โปรแกรมเราได้ตั้งเงื่อนไขไว้ว่าหาก user ต้องการอ่านข้อความ
ย้อนหลัง (Unread message) Client จะต้องกด Resume
เพื่อเป็นสถานะเป็นพร้อมรับข้อความก่อนที่จะกด Get unread
เพื่อรับข้อความย้อนหลัง

