****

**De La Salle University • College of Computer Studies**

**Synchronization Problems**

Name (last name first) : Fernandez, Ryan Austin

Poblete, Clarisse Felicia M.

San Pedro, Marc Dominic

Tan, Johansson E.

Section : G01

Date of Submission : February 16, 2016

**Ticketing Problem**

int globalturn = 1;

Int nextturn = 1;

Int ticket[2] = [-1,-1];

P1

ticket[0] = nextturn++;

while(ticket[0] != globalturn);

//CS

globalturn = ticket[0] + 1;

ticket[0] = -1;

P2

ticket[1] = nextturn++;

while(ticket[1] != globalturn || ticket[1] == ticket[0]);

//CS

globalturn = ticket[1] + 1;

ticket[1] = -1;

**Critical Section Problem**

Monitor CS {

initialization\_code() {}

void CS() {

//do CS

}

}

P1

CS.CS();

P2

CS.CS();

**Barrier Sync**

Monitor BarrierSync {

Condition allSync[3];

int procs;

initialization\_code() {

reset();

}

void reset() {

procs = 0;

}

void update(int i) {

procs++;

if( procs == 3) {

for(int i = 0; i < 3; i++) {

allSync[i].signal();

}

reset();

} else {

allSync[i].wait();

}

}

}

P1

while(true) {

//code

BarrierSync.update(0);

//code

}

P2

while(true) {

//code

BarrierSync.update(1);

//code

}

P3

while(true) {

//code

BarrierSync.update(2);

//code

}

**Reader Writer Problem**

Monitor ReadWrite {

Condition open;

boolean writing;

int read;

initialization\_code() {

read = 0;

writing = false;

}

void read() {

if( writing ) {

open.wait();

}

read++;

}

void finishRead() {

read--;

if( read == 0 ) {

open.signal();

}

}

void write() {

if( read > 0 || writing ) {

open.wait();

}

writing = true;

}

void finishWrite() {

writing = false;

open.signal();

}

}

Reader

ReadWrite.read();

//read here

ReadWrite.finishRead();

Writer

ReadWrite.write();

//write here

ReadWrite.finishWrite();