**/\* Build CO2 with lock and CV \*/**

int wC = 0; // Number of waiting C to be formed

int wO = 0; // Number of waiting O to be formed

int aC = 0; // Number of assigned C to CO2

int aO = 0; // Number of assigned O to CO2

pthread\_mutex\_t lock = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_cond\_t waitingC = PTHREAD\_COND\_INITIALIZER;

pthread\_cond\_t waitingO = PTHREAD\_COND\_INITIALIZER;

CArrives() {

pthread\_mutex\_lock(&lock);

wC = wC+1;

pthread\_mutex\_unlock(&lock);

while(aC == 0) {

if (wO >= 2 && wC >= 1) {

pthread\_mutex\_lock(&lock);

MakeCO2();

wC = wC - 1;

aC = aC + 1;

wO = wO - 2;

aO = aO + 2;

pthread\_mutex\_unlock(&lock);

pthread\_cond\_signal(&waitingO);

pthread\_cond\_signal(&waitingC);

}

else {

pthread\_cond\_wait(&waitingC,&lock);

}

}

pthread\_mutex\_lock(&lock);

aC = aC - 1;

pthread\_mutex\_unlock(&lock);

}

OArrives() {

pthread\_mutex\_lock(&lock);

wO = wO+1;

pthread\_mutex\_unlock(&lock);

while(aO == 0) {

if (wO >= 2 && wC >= 1) {

pthread\_mutex\_lock(&lock);

MakeCO2();

wC = wC - 1;

aC = aC + 1;

wO = wO - 2;

aO = aO + 2;

pthread\_mutex\_unlock(&lock);

pthread\_cond\_signal(&waitingO);

pthread\_cond\_signal(&waitingC);

}

else {

pthread\_cond\_wait(&waitingO,&lock);

}

}

pthread\_mutex\_lock(&lock);

aO = aO - 1;

pthread\_mutex\_unlock(&lock);

}

**/\* Build C2H6O with lock and CV \*/**

int wC = 0; // Number of waiting C to be formed

int wO = 0; // Number of waiting O to be formed

int wH = 0; // Number of waiting H to be formed

int aC = 0; // Number of assigned C to C2H6O

int aO = 0; // Number of assigned O to C2H6O

int aH = 0; // Number of assigned H to C2H6O

pthread\_mutex\_t lock = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_cond\_t waitingC = PTHREAD\_COND\_INITIALIZER;

pthread\_cond\_t waitingO = PTHREAD\_COND\_INITIALIZER;

pthread\_cond\_t waitingH = PTHREAD\_COND\_INITIALIZER;

CArrives() {

pthread\_mutex\_lock(&lock);

wC = wC+1;

pthread\_mutex\_unlock(&lock);

while(aC == 0) {

if (wC >= 2 && wH >= 6 && wO >= 1) {

pthread\_mutex\_lock(&lock);

MakeC2H6O();

wC = wC - 2;

aC = aC + 2;

wH = wH - 6;

aH = aH + 6;

wO = wO - 1;

aO = aO + 1;

pthread\_mutex\_unlock(&lock);

pthread\_cond\_signal(&waitingO);

pthread\_cond\_signal(&waitingC);

pthread\_cond\_signal(&waitingH);

}

else {

pthread\_cond\_wait(&waitingC,&lock);

}

}

pthread\_mutex\_lock(&lock);

aC = aC - 1;

pthread\_mutex\_unlock(&lock);

}

OArrives() {

pthread\_mutex\_lock(&lock);

wO = wO+1;

pthread\_mutex\_unlock(&lock);

while(aO == 0) {

if (wC >= 2 && wH >= 6 && wO >= 1) {

pthread\_mutex\_lock(&lock);

MakeC2H6O();

wC = wC - 2;

aC = aC + 2;

wH = wH - 6;

aH = aH + 6;

wO = wO - 1;

aO = aO + 1;

pthread\_mutex\_unlock(&lock);

pthread\_cond\_signal(&waitingO);

pthread\_cond\_signal(&waitingC);

pthread\_cond\_signal(&waitingH);

}

else {

pthread\_cond\_wait(&waitingO,&lock);

}

}

pthread\_mutex\_lock(&lock);

aO = aO - 1;

pthread\_mutex\_unlock(&lock);

}

HArrives() {

pthread\_mutex\_lock(&lock);

wH = wH+1;

pthread\_mutex\_unlock(&lock);

while(aH == 0) {

if (wC >= 2 && wH >= 6 && wO >= 1) {

pthread\_mutex\_lock(&lock);

MakeC2H6O();

wC = wC - 2;

aC = aC + 2;

wH = wH - 6;

aH = aH + 6;

wO = wO - 1;

aO = aO + 1;

pthread\_mutex\_unlock(&lock);

pthread\_cond\_signal(&waitingO);

pthread\_cond\_signal(&waitingC);

pthread\_cond\_signal(&waitingH);

}

else {

pthread\_cond\_wait(&waitingH,&lock);

}

}

pthread\_mutex\_lock(&lock);

aH = aH - 1;

pthread\_mutex\_unlock(&lock);

}