#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

int mcount,wcount;

sem\_t x,y,z,wsem,msem,cap;

void delay(void) {

int i;

int delaytime;

delaytime = random();

for (i = 0; i<delaytime; i++);

}

void \*woman(void \*param) {

sem\_wait(&z);

sem\_wait(&wsem);

sem\_wait(&y);

wcount++;

if(wcount==1)

{ sem\_wait(&msem); }

sem\_post(&y);

sem\_post(&wsem);

sem\_post(&z);

sem\_wait(&cap);

printf("woman in!\n");

delay();

printf("\twoman out!\n");

sem\_post(&cap);

sem\_wait(&y);

wcount--;

if(wcount==0)

{ sem\_post(&msem); }

sem\_post(&y);

}

void \*man(void \*param) {

sem\_wait(&z);

sem\_wait(&msem);

sem\_wait(&x);

mcount++;

if(mcount==1)

{ sem\_wait(&wsem); }

sem\_post(&x);

sem\_post(&msem);

sem\_post(&z);

sem\_wait(&cap);

printf("\t\tman in!\n");

delay();

printf("\t\t\tman out!\n");

sem\_post(&cap);

sem\_wait(&x);

mcount--;

if(mcount==0)

{sem\_post(&wsem);}

sem\_post(&x);

}

int main(void) {

int i;

srandom(60);

mcount = 0;

wcount = 0;

sem\_init(&x,0,1); // for sem\_init, initial value is 3rd argument

sem\_init(&y,0,1);

sem\_init(&z,0,1);

sem\_init(&wsem,0,1);

sem\_init(&msem,0,1);

sem\_init(&cap,0,4); // eg. cap initialized to 4

pthread\_t \*tid;

tid = malloc(80\*sizeof(pthread\_t));

// You can use your cobegin statement here, instead of pthread\_create()

// I have forgone the use of pthread barriers although I suppose they would nicely imitate the functionality of cobegin.

// This is merely to retain simplicity.

for(i=0;i<10;i++) {

pthread\_create(&tid[i],NULL,woman,NULL);

}

for(i=10;i<20;i++) {

pthread\_create(&tid[i],NULL,man,NULL);

}

for(i=0;i<20;i++) {

pthread\_join(tid[i],NULL);

}

return(0);