ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ - ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΑΥΤΟΧΡΟΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Prime Number Testing using monitors

Thread_func:

Entering monitor

Available = 1;

While(exec_sig!=1) { wait(&cond,&monitor) }

Primetest();

If(Job_finished) {wait(&cond,&monitor) }

Exiting monitor

Return;

Main:

while(!available){wait}

Entering Monitor

While(Job Exists) { exec_sig=1; signal(&cond) }

Job_Finished =1;

signal(&cond)

Exiting Monitor

return;

Bridge

Left/Right vehicle():

Entering Monitor

While(left/right==0) { wait(&cond,&monitor);}

if(car_on_bridge < N || car_on_bridge < howmany_left/right) {enter bridge; }

if(car_on_bridge == N || howmany_left/right ==0) {

if(howmany_right/left) {signal(&cond); return;}

}

Signal(howmany_left/right);

Exiting Monitor

Roller Coaster

Passengers:

Entering monitor

while(came!=1) {wait(&pass,&monitor)

if (!last) {Enter Train ; signal(&pass)}

else {signal(&train)}

Exiting Monitor

Train:

While(pass_exists) {

Entering Monitor

While(how_many!=N) { came =1; signal(&pass); wait(&train,monitor) }

TRIP;

Pass_exiting;

Exiting Monitor

Υλοποίηση CCR

```
#define CCR_DECLARE(label) \
pthread_mutex_t label ## mtx;\
pthread_cond_t label ## c;\
#define CCR_INIT(label) \
pthread_mutex_init(&label ## mtx ,NULL);\
pthread_cond_init( &label ## c,NULL);\
#define CCR_EXEC(label,cond,body)\
 pthread_mutex_lock(&label ## mtx);\
 while(!(cond)) {\
 pthread_cond_wait(&label ## c,&label ## mtx);\
pthread_cond_signal(&label ## c);\
}\ body\
pthread_cond_signal(&label ## c);\
 pthread_mutex_unlock(&label ## mtx);\
```

Bridge with CCR

Left/Right vehicle():

CCR_EXEC(label_left,left,body*)

Return(NULL);