Submitted By: -

Subham Agrawal

(U17CO110)

Philosopher Dining Problem

Code:

#define N 4

byte fork[N];

byte nr\_eat;

proctype Philosopher(byte id) {

Think:

printf("Plilosopher with id %d is thinking\n",id);

if

:: atomic { fork[id] == 0 -> fork[id] = id + 1; };

:: atomic { fork[(id + 1)%N] == 0 -> fork[(id + 1)%N] = id + 1; };

fi;

One:

if

:: atomic {

fork[id] == id + 1 -> fork[(id + 1)%N] == 0 -> fork[(id + 1)%N] = id + 1;

nr\_eat++;

}

:: atomic {

fork[id] == 0 -> fork[(id + 1)%N] == id + 1 -> fork[id] = id + 1;

nr\_eat++;

}

fi;

Eat:

printf("Plilosopher with id %d is eating\n",id);

d\_step { nr\_eat--; fork[(id + 1)%N] = 0; fork[id] = 0;}

goto Think;

}

init {

 atomic {

byte i = 0;

do

:: i < N -> run Philosopher(i); i++;

:: else -> break;

od;

}

}

