BITS, PILANI – K. K. BIRLA GOA CAMPUS

Operating Systems

by

Dr. Shubhangi

Synchronization

PROCESS SYNCHRONIZATION

Some Interesting Problems

THE READERS/WRITERS PROBLEM:

```
BINARY_SEMAPHORE    wrt          = 1;
BINARY_SEMAPHORE    mutex        = 1;
int                 readcount = 0;
```

```
Writer:
do {
        wait( wrt );
        /*   writing is performed     */
        signal(  wrt  );
} while(TRUE);
```

```
Reader:
do {
        wait( mutex );                              /* Allow 1 reader in entry*/
        readcount = readcount + 1;
        if readcount == 1  then  wait(wrt );     /* 1st reader locks writer */
        signal( mutex );
                /*   reading is performed  */
         wait( mutex );
         readcount = readcount - 1;
         if readcount == 0  then  signal(wrt );   /*last reader frees writer */
         signal( mutex );
} while(TRUE);
```

```
WAIT ( S ):
     while    ( S  <=  0 );
     S  =  S  -  1;
SIGNAL ( S ):
     S  =  S  +  1;
```

PROCESS SYNCHRONIZATION

Some Interesting Problems

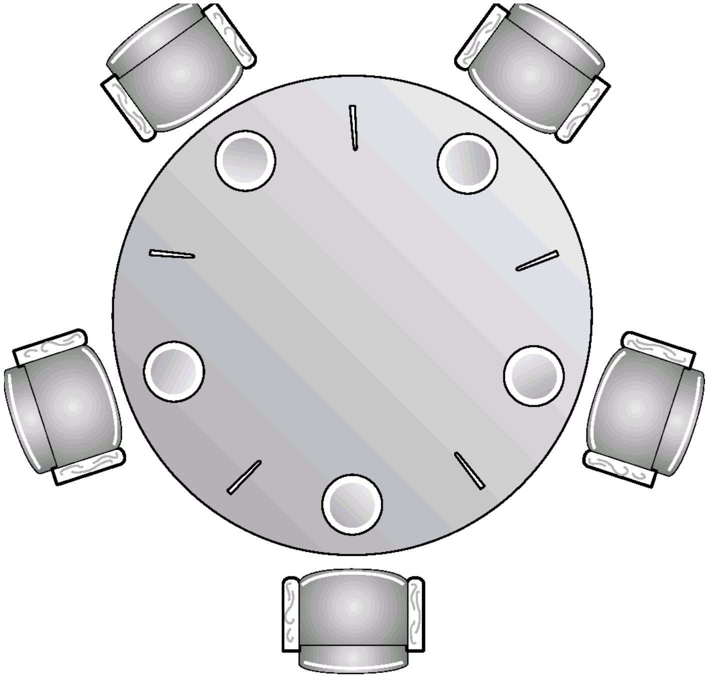
THE DINING PHILOSOPHERS PROBLEM:

5 philosophers with 5 chopsticks sit around a circular table. They each want to eat at random times and must pick up the chopsticks on their right and on their left.

Clearly deadlock is rampant (and starvation possible.)

Several solutions are possible:

- Allow only 4 philosophers to be hungry at a time.
- Allow pickup only if both chopsticks are available. (Done in critical section)
- Odd # philosopher always picks up left chopstick 1st, even # philosopher always picks up right chopstick 1st.



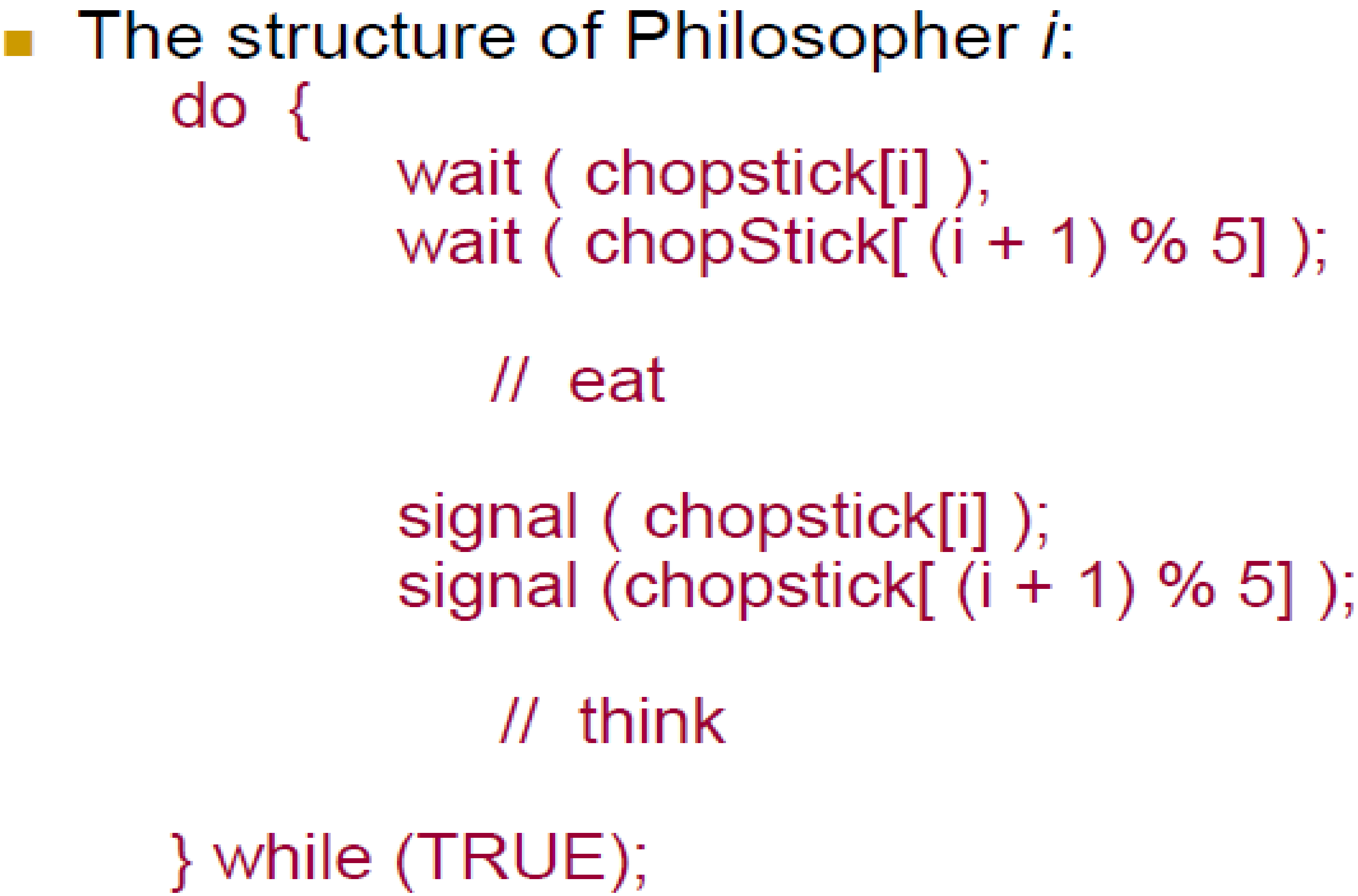
DINING PHILOSOPHER PROBLEM

- Data structure support needed
  - semaphore chopstick [N] ;

- Data structure Initialization
  - for(int i=0; i< N; i + +)

        chopstick [i] = 1;

DINING PHILOSOPHER PROBLEM



- The structure of Philosopher *i*:

```
do  {
          wait ( chopstick[i] );
          wait ( chopStick[ (i + 1) % 5] );

               //  eat

          signal ( chopstick[i] );
          signal (chopstick[ (i + 1) % 5] );

               //  think

} while (TRUE);
```

DINING PHILOSOPHER PROBLEM

- Draw back
  - Dead lock is possible → if everyone picks up left/right chopstick
- Solution
  - The philosopher can pick up a chopstick if both the left & right one are free.
  - Let the odd philosopher pick the left chopstick and even philosopher pick the right chopstick.
  - Take only even number of philosophers