

Concurrency

EECS 4315

www.cse.yorku.ca/course/4315/

The Readers-Writers Problem

The readers and writers problem, due to Courtois, Heymans and Parnas, is a classical concurrency problem. It models access to a database. There are many competing threads wishing to read from and write to the database. It is acceptable to have multiple threads reading at the same time, but if one thread is writing then no other thread may either read or write. A thread can only write if no thread is reading.

- Canadian early pioneer of software engineering
- Ph.D. from Carnegie Mellon University
- Taught at the University of North Carolina at Chapel Hill, the Technische Universität Darmstadt, the University of Victoria, Queen's University, McMaster University, and University of Limerick
- Won numerous awards including ACM SIGSOFT's "Outstanding Research" award



David Parnas

source: Hubert Baumeister

- Professor emeritus at the Catholic University of Leuven



Pierre-Jacques Courtois

source: www.info.ucl.ac.be/~courtois

The Readers-Writers Problem

```
public class Reader extends Thread {
    private Database database;

    public Reader(Database database) {
        this.database = database;
    }

    public void run() {
        while (true) {
            try {
                this.database.read();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

The Readers-Writers Problem

```
public class Database {  
    ...  
  
    public Database() { ... }  
    public void read() { ... }  
    public void write() { ... }  
}
```

The Readers-Writers Problem

Question

If we make both methods synchronized, does that solve the problem?

The Readers-Writers Problem

Question

If we make both methods synchronized, does that solve the problem?

Answer

Yes.

The Readers-Writers Problem

Question

If we make both methods synchronized, does that solve the problem?

Answer

Yes.

Question

Is it a satisfactory solution?

The Readers-Writers Problem

Question

If we make both methods synchronized, does that solve the problem?

Answer

Yes.

Question

Is it a satisfactory solution?

Answer

No.

The Readers-Writers Problem

Question

Why is it not satisfactory?

The Readers-Writers Problem

Question

Why is it not satisfactory?

Answer

It does not allow multiple readers to read at the same time.

The Readers-Writers Problem

Questions to consider when solving this problem:

- Of which information do we need to keep track?
- When does a reader have to wait?
- When does a writer have to wait?
- Who notifies a waiting writer?
- Who notifies a waiting reader?

The Dining Philosophers Problem

In the dining philosophers problem, due to Dijkstra, five philosophers are seated around a round table. Each philosopher has a plate of spaghetti. The spaghetti is so slippery that a philosopher needs two forks to eat it. The layout of the table is as follows.



The life of a philosopher consists of alternative periods of eating and thinking. When philosophers get hungry, they try to pick up their left and right fork, one at a time, in either order. If successful in picking up both forks, the philosopher eats for a while, then puts down the forks and continues to think.

The Dining Philosophers Problem

```
public class Philosopher {
    private int id;
    private Table table;

    public Philosopher(int id, Table table) {
        this.id = id;
        this.table = table;
    }
    public void run() {
        while (true) {
            this.table.pickUp(id);
            this.table.pickUp(id + 1 % 5);
            // eat
            this.table.putDown(id);
            this.table.putDown(id + 1 % 5);
        }
    }
}
```

The Dining Philosophers Problem

```
public class Table {  
    public void pickUp(int id) { ... }  
    public void putDown(int id) { ... }  
}
```


The Dining Philosophers Problem

Question

If we make both methods synchronized, does that solve the problem?

The Dining Philosophers Problem

Question

If we make both methods synchronized, does that solve the problem?

Answer

No.

The Dining Philosophers Problem

Question

If we make both methods synchronized, does that solve the problem?

Answer

No.

Question

Why not?

The Dining Philosophers Problem

Question

If we make both methods synchronized, does that solve the problem?

Answer

No.

Question

Why not?

Answer

Deadlock.

Today.