Due dates of assignments

- Assignment 1: January 27
- Assignment 2: February 24
- Assignment 3: March 24

Dates of presentations

- Presentation 1: February 5
- Presentation 2: March 5
- Presentation 3: April 1

Concurrent Programming Languages

Question

Can you name some concurrent programming languages?

Concurrent Programming Languages

Most concurrent programming languages consist of a sequential programming language plus support for

- thread creation,
- communication, and
- synchronization

Thread Creation

We distinguish between

- static thread creation
  only allowing a predefined number of threads
- dynamic thread creation
  allowing new threads to be created “on-the-fly”

Communication

We distinguish between communication using

- shared variables
- messages
  - synchronous (blocking send, blocking receive)
  - asynchronous (non-blocking send, blocking receive)

Shared variable communication

Question

What is a real life analogue for shared variable communication?

Synchronous message passing communication

Question

What is a real life analogue for synchronous message passing communication?

Asynchronous message passing communication

Question

What is a real life analogue for asynchronous message passing communication?

Synchronization

- semaphores
- locks
- monitors
- barriers
- compare-and-swap
- ...

Mergesort

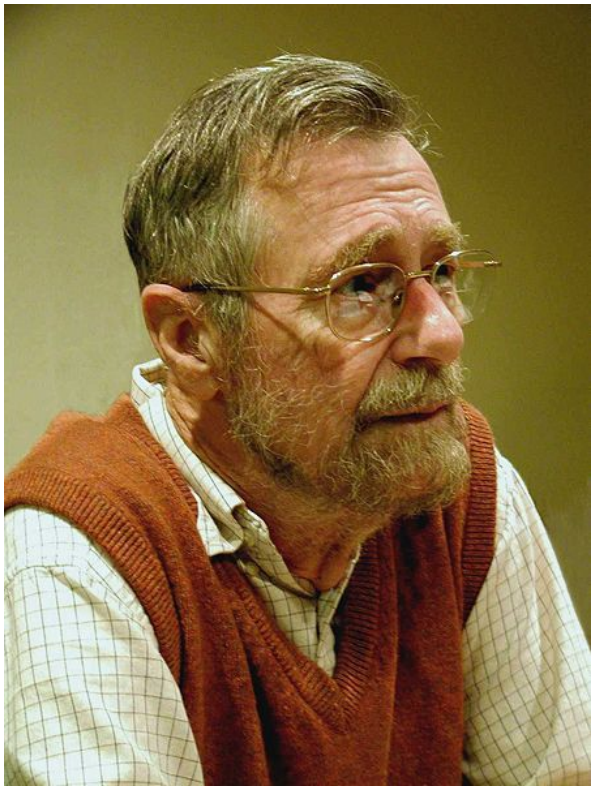
```
public void sort(int[] a)
{
   mergesort(a, 0, a.length)
}

public void mergesort(int[] a, int low, int high)
{
   // fill in the details
}

public void merge(int[] a, int low, int high)
```

Edsger Wybe Dijkstra

- Member of the Royal Netherlands Academy of Arts and Sciences (1971)
- Distinguished Fellow of the British Computer Society (1971)
- Recipient of the Turing Award (1972)
- Foreign Honorary Member of the American Academy of Arts and Sciences (1975)



Edsger Wybe Dijkstra

(1930–2002)

The Critical Section Problem

Consider two threads both defined by

```
while (true)
{
    critical section
    non-critical section
}
```

- Mutual exclusion: Make sure that at any moment at most one of the threads is in its critical section.
- Freedom from deadlock: If one of the threads is well outside its critical section, this is not allowed to lead to a potential blocking of the other thread trying to enter its critical section.
- Freedom from starvation: If both threads are about to enter their critical section, then the decision which one is to enter its critical section cannot be postponed indefinitely.

The Producer-Consumer Problem

The producer-consumer problem (also known as the bounded-buffer problem) is a classical concurrency problem.

The problem is to synchronize two threads, the producer and the consumer, who share a common, fixed-size buffer. The producer repeatedly generates a data item and puts it into the buffer. At the same time, the consumer removes data items from the buffer, one item at a time.

The problem is to make sure that the producer will not try to add data items to a full buffer and that the consumer will not try to remove data items from an empty buffer.

The Producer-Consumer Problem

We assume that the items are integers. We represent the buffer by means of an array of integers. The array has a fixed size.

```
int N = 10; // capacity of buffer
```

The producer and consumer share the following variables.

```
int[] buffer; // array representing buffer
int next = 0; // index of cell for next item
int size = 0; // number of items stored in buffer
```

The Producer-Consumer Problem

Producer:

```
while (true)
  int value = produce an item;
  buffer[next] = value;
  size++;
  next = (next + 1) mod N;
```

Consumer:

```
while (true)
  int value = buffer[(next - size) mod N];
  size--;
```

The Producer-Consumer Problem

Question

How can we make sure that the producer will not try to add data items to a full buffer?

Question

How can we make sure that the consumer will not try to remove data items from an empty buffer?

The Readers-Writers Problem

The readers and writers problem, due to Courtois, Heymans and Parnas, is another 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. The problem is how do you program the reader and writer threads?

The Readers-Writers Problem

The readers and writers share the following variable.

```
semaphore mutex = 1;
```

Reader:

```
P(mutex);
read;
V(mutex);
```

Writer:

```
P(mutex);
write;
V(mutex);
```

The Readers-Writers Problem

Question

Does it solve the readers-writers problem?

The Readers-Writers Problem

Question

Does it solve the readers-writers problem?

Answer

Yes!

The Readers-Writers Problem

Question

Does it solve the readers-writers problem?

Answer

Yes!

Question

Is it a satisfactory solution?

The Readers-Writers Problem

Question

Does it solve the readers-writers problem?

Answer

Yes!

Question

Is it a satisfactory solution?

Answer

No!

The Readers-Writers Problem

Question

Why not?

The Readers-Writers Problem

Question

Why not?

Answer

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

The Readers-Writers Problem

Options

While a writer is writing, readers and writers arrive. Once the writer is done, can the readers start reading?

Options

While readers are reading, readers and writers arrive. Can the readers start reading?

The Readers-Writers Problem

Options

While a writer is writing, readers and writers arrive. Once the writer is done, can the readers start reading?

Yes

Options

While readers are reading, readers and writers arrive. Can the readers start reading?

Yes

No reader is kept waiting unless a writer is writing.

The Readers-Writers Problem

Options

While a writer is writing, readers and writers arrive. Once the writer is done, can the readers start reading?

Options

While readers are reading, readers and writers arrive. Can the readers start reading?

The Readers-Writers Problem

Options

While a writer is writing, readers and writers arrive. Once the writer is done, can the readers start reading?

No

Options

While readers are reading, readers and writers arrive. Can the readers start reading?

No

If a writer wants to write, it writes as soon as possible.