Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Synchronous Message Passing

Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

(A)Synchronous Message Passing

https://en.wikipedia.org/wiki/Message_passing

- In concurrency theory, **synchronous message passing** means:
  - A sender and receiver must meet at the same time for a message to be transferred.
  - The sender **blocks until another thread receives the message**.

Producer thread/process		Consumer/process thread
put(item) (waiting)	----->	take() (receives item)

Java syntax

(A)Synchronous Message Passing

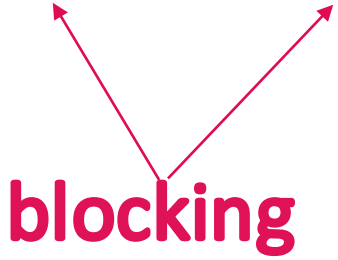
- In asynchronous message passing, channels contain unbounded num of messages.
  - If p1 sends a msg to p2 and needs to be sure p2 received it, it must wait for a reply!
  - If there is a delivery failure, p1 has no way to know the state of msg.
  - Message have to be buffered (no infinite space!). If too many msg arrive, program will crash or send() will be blocked.

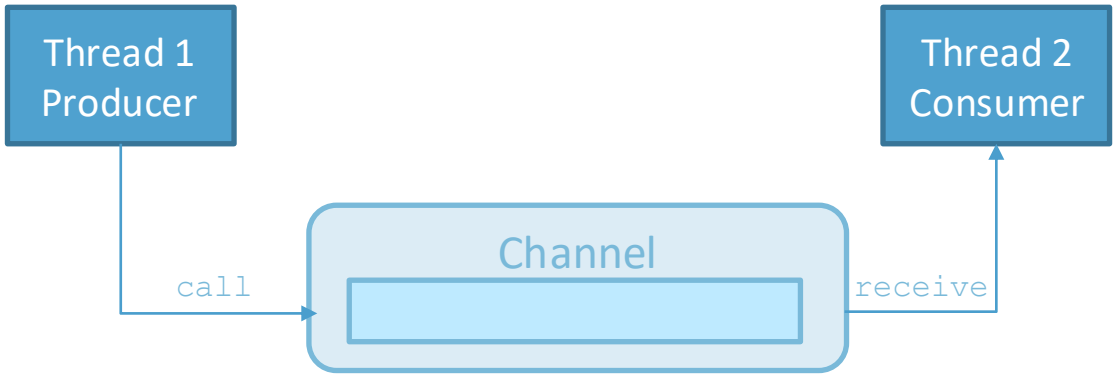
- Synchronous Message passing avoid these situations

(A)Synchronous Message Passing

https://en.wikipedia.org/wiki/Message_passing

- Processes share **channels** …
  - Abstraction of a physical communication network
- … to communicate between processes
  - `call` and `receive`

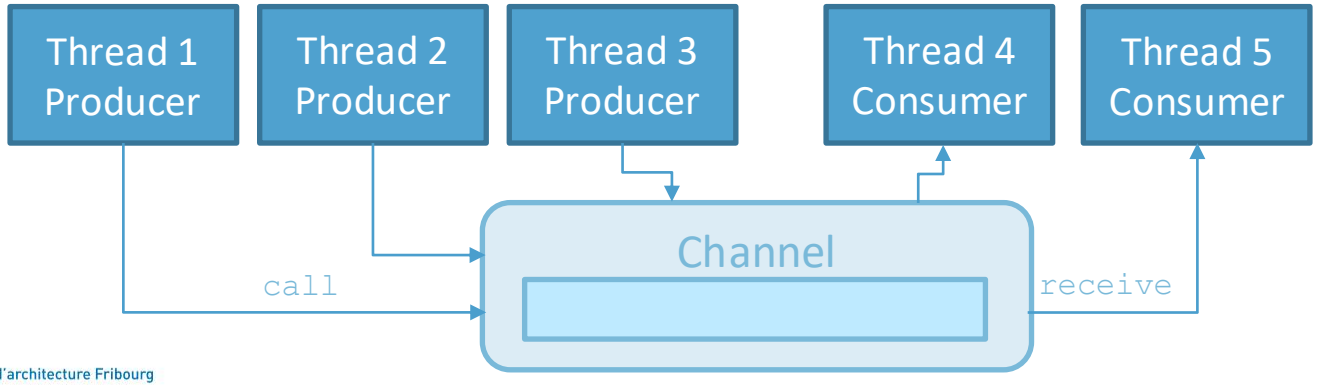




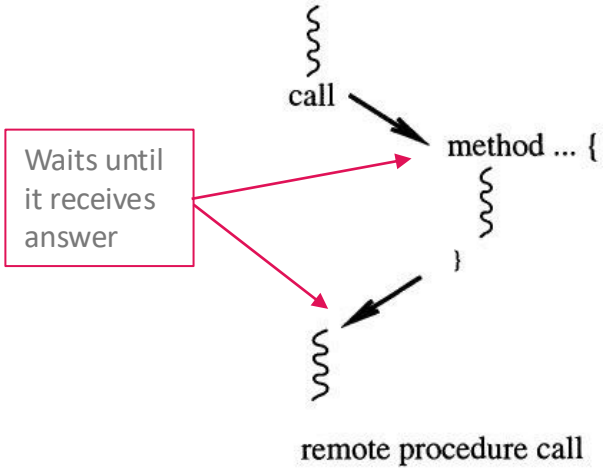
(A)Synchronous Message Passing

https://en.wikipedia.org/wiki/Message_passing

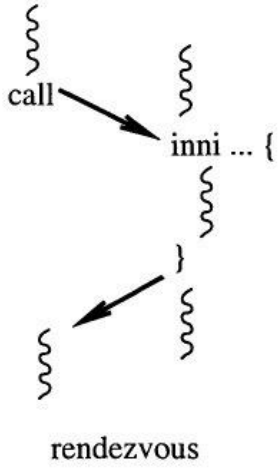
- Processes share **channels** …
  - Abstraction of a physical communication network
- … to communicate between processes
  - **`call`** and **`receive`**



Synchronous



remote procedure call



rendezvous

Serviced by: **method (`op`)** **input statement (`inni`)** (e.g. JR)

Sync msg passing – Pros / Cons

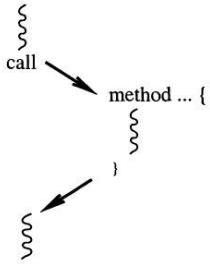
- The sending part decides if it is synchronous or asynchronous
- Synchronous - sender **blocks** until the receiver picks up the message

➔ Guarantee that the message is received by the receiver

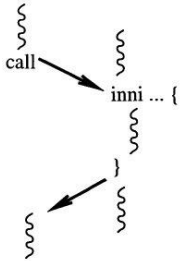
- It's like making a (remote) function call
  - RPC (remote procedure call) will be shown later
- A broken receiver breaks also the caller!

Key take away messages

- Sender puts message in channel and blocks
- Consuming the message deblocks the sender



remote procedure call



rendezvous

Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Synchronous Message Passing

Java

Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

AbstractQueues – Once again…

- Again, we can use the `AbstractQueue` interface from `java.util.concurrent`
- The Java counterpart for the synchronous message passing is

`SynchronousQueue()`

- `offer()`, `poll()`, `put()`, …
- Information on the official Oracle site: https://cutt.ly/Ft1Iw0T

SynchronousQueue – Some limitations

- Compared to the `AbstractQueues`, the **`SynchronousQueue`** has some limitations
  - Zero internal capacity!
  - `peek()` is not possible
  - Iteration over the queue is not possible... don't forget, there are no elements in the queue
  - The `SynchronousQueue` acts as an empty collection
  - *Null* elements are not permitted

Fairness

- Optional fairness policy
  - Ordering waiting producer and consumer threads

- By default, ordering is not guaranteed

Constructor	Description
`SynchronousQueue()`	Creates a `SynchronousQueue` with nonfair access policy.
`SynchronousQueue (boolean fair)`	Creates a `SynchronousQueue` with the specified fairness policy.

Key take away messages

- `SynchronousQueue`
- Usage of the well-known methods …
- … but with limitations …
- … because zero size buffer