

*Duc Anh Phi,
Michael Tabachnik,
Edgar Brotzmann*

Solutions to Problem Set 2

Due: 06.11.2018, 2pm

Exercise 1

In Python function parameters are passed using **copy-by-object-reference**. The parameter type does not matter, because (almost) everything is an object in Python, which can be referenced. Given the example code both arguments ***data*** and ***oldlist*** are references to their bound objects.

In Java and C function parameters are passed using **copy-by-value**. Objects are passed copying their addresses into the function parameter.

Exercise 2

Processes do not share the same address space/ memory, which is why pointers or references in process A cannot be interpreted by process B via remote procedure calls. Pointers are only valid within one address space.

Exercise 3

Fixed size alignments do not have to deal with maintaining message boundaries, fragmentation and reassembly. As a drawback large primitive values exceeding 4 bytes are not possible.

*Duc Anh Phi,
Michael Tabachnik,
Edgar Brotzmann*

Exercise 5

```
### Starting request with 1 parallel requests...
### Computation for 1 parallel requests took 51.88469360998715 seconds

### Starting request with 2 parallel requests...
### Computation for 2 parallel requests took 50.205210462998366 seconds

### Starting request with 3 parallel requests...
### Computation for 3 parallel requests took 50.64487799300696 seconds

### Starting request with 4 parallel requests...
### Computation for 4 parallel requests took 53.5487871519872 seconds

### Starting request with 5 parallel requests...
### Computation for 5 parallel requests took 50.29823527400731 seconds

### Starting request with 6 parallel requests...
### Computation for 6 parallel requests took 49.106504388008034 seconds

### Starting request with 7 parallel requests...
### Computation for 7 parallel requests took 51.13789443401038 seconds

### Starting request with 8 parallel requests...
### Computation for 8 parallel requests took 51.58003035999718 seconds

### Starting request with 9 parallel requests...
### Computation for 9 parallel requests took 49.048352997982875 seconds

### Starting request with 10 parallel requests...
### Computation for 10 parallel requests took 50.8850658859883 seconds
```

The elapsed execution times seem arbitrary for each number of parallel requests.

*Duc Anh Phi,
Michael Tabachnik,
Edgar Brotzmann*

Exercise 6

Sun **O**pen **N**etwork **C**omputing (ONC)

ONC RPC is transport independent - it deals only with the specification and interpretation of messages. The transport protocol can be selected at run-time.

Additionally it has developer friendly features:

- You don't have to worry about getting a unique transport address (port) but you need a unique program number per server.
- RPC applications do not have to deal with maintaining message boundaries, fragmentation, reassembly.
- simplified interface for RPC

ONC RPC can be used for projects where the transport protocol can change at runtime (for example changing UDP to TCP).

Distributed Computing Environment (DCE)

DCE RPC allows several machines (servers) to be organized into an administrative entity. Each client communicates with the administrative entity for services information.

Additionally DCE RPC has features such as:

- authenticated rpc support
- automatic generation of program ID (in contrast to ONC)

DCE RPC can be used for projects which need to scale.

*Duc Anh Phi,
Michael Tabachnik,
Edgar Brotzmann*

CORBA Common Object Request Broker Architecture

CORBA enables software components written in multiple computer languages and running on multiple computers to work together. It is reliable, comprehensive and language neutral.

As drawbacks CORBA:

- is complex, expensive to implement and often ambiguous
- has a steep learning curve
- supports only tightly coupled environments.

We would recommend CORBA for example for projects consisting microservices which are implemented in different languages.