The Solutions to Tutorial Questions and Lab Projects of Week 2

Tutorial Questions

1. A server creates a port which it uses to receive requests from clients. Discuss the design issues concerning the relationship between the name of this port and the names used by clients.

   Answer

   The main design issues for locating server ports are:

   (i) How does a client know what port and IP address to use to reach a service?

   The options are:

   - use a name server/binder to map the textual name of each service to its port;
   - each service uses well-known location-independent port id, which avoids a lookup at a name server.

   The operating system still has to look up the whereabouts of the server, but the answer may be cached locally.

   (ii) How can different servers offer the service at different times?

   Location-independent port identifiers allow the service to have the same port at different locations. If a binder is used, the client needs to reconsult the client to find the new location.

   (iii) Efficiency of access to ports and local identifiers.

   Sometimes operating systems allow processes to use efficient local names to refer to ports. This becomes an issue when a server creates a non-public port for a particular client to send messages to, because the local name is meaningless to the client and must be translated to a global identifier for use by the client.

2. Sun XDR marshals data by converting it into a standard big-endian form before transmission. Discuss the advantages and disadvantages of this method when compared with CORBA's CDR.

   Answer

   The XDR method which uses a standard form is inefficient when communication takes place between pairs of similar computers whose byte orderings differ from the standard. It is efficient in networks in which the byte-ordering used by the majority of the computers is the same as the standard form. The conversion by senders and recipients that use the standard form is in effect a null operation.

   In CORBA CDR senders include an identifier in each message and recipients to convert the bytes to their own ordering if necessary. This method eliminates all unnecessary data conversions, but adds complexity in that all computers need to deal with both variants.

3. Why is there no explicit data-typing in CORBA CDR?

   Answer

   The use of data-typing produces costs in space and time. The space costs are due to the extra type information in the marshalled form (see for example the Java serialized form). The performance cost is due to the need to interpret the type information and take appropriate action.

The RMI protocol for which CDR is designed is used in a situation in which the target and the invoker know what type to expect in the messages carrying its arguments and results. Therefore type information is redundant. It is of course possible to build type descriptors on top of CDR, for example by using simple strings.

4. Write an algorithm in pseudocode to describe the serialization procedure described in Section 4.3.2. The algorithm should show when handles are defined or substituted for classes and instances. Describe the serialized form that your algorithm would produce when serializing an instance of the following class Couple.

```
class Couple implements Serializable{
       private Person one;
       private Person two;
       public Couple(Person a, Person b) {
       one = a;
       two = b;
       }
}
```

Answer

The algorithm must describe serialization of an object as writing its class information followed by the names and types of the instance variables. Then serialize each instance variable recursively.

```
serialize(Object o) {
       c = class(o);
       class_handle = get_handle(c);
       if (class_handle==null) // write class information and define
       class_handle;
       write class_handle
       write number (n), name and class of each instance variable
       object_handle = get_handle(o);
       if (object_handle==null) {
            define object_handle;
            for (iv = 0 to n-1)
              if (primitive(iv) ) write iv
              else serialize( iv)
       }
       write object_handle
}
```

To describe the serialized form that your algorithm would produce when serializing an instance of the class Couple.

For example declare an instance of Couple as

```
Couple t1 = new Couple(new Person("Smith", "London", 1934),
new Person("Jones", "Paris", 1945));
```

The output will be:

Serialized values				*Explanation*
Couple	8 byte version number		h0	*class name, version number, handle*
2	Person one	Person two		*number, type and name of instance variables*
Person	8 byte version number		h1	*serialize instance variable one of Couple*
3	int year	java.lang.String name	java.lang.String place	
1934	5 Smith	6 London	h2	
h1				*serialize instance variable two of Couple values of instance variables*
1945	5 Jones	5 Paris	h3	

5. Outline how connectionless communication between a client and a server is established and proceeded by using sockets.

   Answer

   By using a datagram socket, it provides a connectionless communication interface. The processes of communication don't require a set-up of connection. What you need to do is to create a socket for each process of client and server, then the server binds the socket to a local endpoint. The server can subsequently do a blocking read call when it waits for incoming data from clients. Similarly, after creating the socket, the client simply does a blocking call to write data to the server. It is not necessary to close a connection whenever.

Lab Projects

Download the source code from Week 2 block of the course Moodle site for Task 1 to Task 4.