CSL458-2013-II-Assignment-1

CSL458 | Assignment#1 | 11 Mar 2014 | Due on 21 Mar 2014

Important Instructions

1. Your submission should be packaged as a ZIP file named as: CSL458-A1-XXXX.zip. Here XXXX is your roll number. For example, CSL458-A1-1023.zip
2. Plagiarism and sharing code will lead to F grade (and possibly disciplinary action for repeat cases). You are responsible for ensuring that you code is not copied by others.
3. Please document the code properly; there is partial credit for clean and well written code.
4. Your score for an assignment solution will be assessed by looking at:
    a. Implementation approach and its correctness.
    b. Readability of the code and associated manual/readme etc.
    c. Correct functioning of the code. That is, it should produce correct results for various input scenarios.
    d. Quality of your design and code. For example, if changing one input value requires you to recompile your program then it is a bad design.

We need to build a system where any machine which is connected to a network can offer certain services to local or remote machine. The services should be published on a well known port, and use a standard interface. The standard interface here means the programming API which the clients of such a service should use to call the service. The service can be invoked synchronously or asynchronously. Here, synchronous invocation means that a client invoking this service blocks until the server completes the service request and returns results to client. Asynchronous invocation of service means that the client can *submit* the service request using one function and keep checking (polling) the status of service completion via a different function, and finally obtain the results in yet another function call.

Examples:

1. A service that takes MS Word documents and converts them into PDF files.
2. A service which calculates eigenvalues of large matrices.
3. A service which scales the images.

You need to provide the following:

1. Design of the standard interface which can cater to a wide range of services. This means that the signature of the remote operations/functions should be designed such that it can address a variety of service scenarios. Special care should be taken to deal with error scenarios.
2. Implementation of a service. You should implement some service functionality and expose it via the above remote interface. For instance, you may implement a service to multiply large matrices.
3. Provide a client-side library which can be used by clients on remote (or local) machines to invoke the service.
4. Implement a client which demonstrates the working of client-server interactions.

You should use sockets APIs available in various programming languages for interactions between processes running on different machines. Do not use existing high level protocols such as HTTP/FTP/IIOP etc. Create your own standard protocol (over sockets API) for interactions between the remote processes.