# Homework 3 for CSC246

Homework 3 consists of 1 written question and 2 coding questions. Respond to each question and submit your work electronically using the link on the course webpage. Please submit your non-program answers in pdf, doc(x), rtf, or plain text. For the short programming questions, just submit the source file. The only comments that are required are comments to explain the compile command you used and the comments specified in the question. You must write the programs in one of the languages specified.

Make sure that your programs work on the school's EOS or Linux servers. **Include compile commands as comments within your program. Specify what environment you ran it on (e.g. VCL or Linux lab).** Contact the TA if you need help accessing the school's EOS or Linux environment remotely. The reason for this requirement is that it is important to make sure a program works for the environment for which it is intended (and on which it will be graded). For purposes of this assignment, the intended environment is the N.C.State's Linux environment.

This requirement is especially important if you are writing source code in Java. If you are using the latest version of Java in Eclipse, for example, there may be options that are not supported by a Java environment that may be loaded with the school's servers.

If you are using your EOS account from a remote location to do java compilations, please use the Virtual Computing Lab. More information about the VCL can be found at [http://vcl.ncsu.edu](http://vcl.ncsu.edu/).

If you need help, please contact the TA at once and arrange to get help. The TA's email address is on the syllabus.  
  
This homework is due on the date mentioned on the main website **by midnight.** This homework contains 65 points and is 6.5% of your total course grade.

*15 points*

## Question 1: Remote Procedure Call

The idea of *remote procedure calls* was launched by Birrell and Nelson in 1984. Read their paper, rpc.pdf, which is available on the course web page, answer the following questions:

Define/explain the following terms *from the paper*: (3 pts each)

* Stubs
  + - Stubs are responsible for the interpretations of arguments and results

Between the user and sever during the RPC.

* binding
  + - The way caller determines the location and identity of the callee and start the process of connecting with the information.
* call packet
  + - The call packet contains the information such identifier and table of the interface, and the entry point number of the desired procedure relative to the interface.

Use your textbook to define/explain the following terms: (2 pts each)

* marshalling
  + - Marshalling involves packaging the parameters into a form that can transfer over a network.
* Microsoft DCOM (Distributed Common Object Model) (see chapter on Windows XP)
  + - DCOM is used to developing distributed applications transparently that utilize RPC.
* Java RMI
  + - RMI stands for Remote Method Invocation, which is an API that allows an object to invoke another object in another address. This allows the communication between client and server established through simple method calls.

*25 points*

## Question 2 - Find a Word

Write a C or C++ program that executes on a Unix or Linux machine. The program should display each line a given word appears in a set of files. The program's first argument is the word to find. The remaining arguments are **1 to 5 file names** that the program should search.

A word is any sequence of characters delimited by whitespace (blank, tab, or newline).

Each file must have the specified word found and results (each line containing the word) displayed *by a separate process*. So if there were 5 command line arguments, there would be a process for *each* find operation.

**Name your source file find.c or find.cpp**.

If you need help processing command line arguments, review the sample code provided in the ‘Examples’ link of the website.

*25 points*

## Question 3 - Find Revisited

Write a c, c++, or Java program that provides the same functionality as question 2 but each file has the words found *by a separate thread*.

The c or c++ program must use the pthreads package for multi-threading.

**Name this program find2.c, find2.cpp or find2.java**.

*Read the thread examples in the textbook and the sample code on the ‘Examples’ page. See "bookthreadexample..."*