Your assignment is to write a history program that **returns a list of events that took place on a particular day in history.** The program will consist of two parts: a client and server. The server maintains the data and the client supports add and query operations.

You will also write a makefile to build your programs. Be sure to allot time for this if you have never written a makefile.

**Languages**

The point of this assignment is to use Linux RPC. Remote procedure calls for Linux distributions provide an RPC version derived from the RPC facility developed the the Open Network Computing (ONC) group at Sun Microsystems. **You will need to do this assignment in C.**

**Preface**

Do not attempt this assignment until you feel comfortable using ONC RPC. First, go through the [RPC tutorial](http://www.cs.rutgers.edu/~pxk/rutgers/notes/rpc/index.html) and the RPC Programming Guide. Don't just read the tutorial — compile the examples. You should learn to use a makefile. There are many online tutorials for makefiles if you have not used one before.

Make sure that you are doing this assignment on a system that supports Linux RPC and has the necessary support for it installed. Any Linux, SUN or OS X machine should work fine.

**Specifications**

**The server**

The server, named **hserver** should be started on some known machine without any arguments. It accepts requests from clients in the form of remote procedure calls that are *add* or *query* operations:

***add***

This enters an event. It contains three parameters: the month, the date, and a string describing the event. It returns a value of zero if the insert cannot take place for some reason (this should not generally happen). Otherwise it returns 1.

***query***

This returns the *next* event in history for a specified month and date. The procedure is called with three parameters: the month, date, and identifier. The identifier is 0 for the first call. It returns two values: a string containing a description of the event and an identifier that corresponds to that event. In order to get the next event, another procedure call is made with the identifier set to the previously returned value.

**The client**

The client, called **hist** has three modes of operation. The first argument will always be the server's machine name. The second will be a string containing the operation. The additional parameter(s) will be operation-specific.

**add**

To add an entry, the month, date, and event have to be supplied:

hist *server* add *month* *date* *event*

For example,

$ *hist multivac add 10 27 "Nylon introduced by DuPont 1938"*

record added

The result of the add operation is either a 0 for failure ot a 1 for success.

**query**

To query for events, the month and date have to be supplied.

hist *server* query *month* *date*

For example,

$ *hist multivac query 10 27*

The first of the "Federalist Papers" was published in 1787

Theodore Roosevelt, 26th president, born in New York City, 1858

radio show "You Bet Your Life" starring Groucho Marx premiered on ABC, 1947

Expo '67 closed in Montreal, 1967

James Cook is born (1466)

The first New York Subway is opened (1904)

Nylon introduced by DuPont 1938

The result of the add operation is either a failure message or a list of events.

**WHAT TO SUBMIT**

Create a plain text file named **id** that contains your name and RUID number. Please write your name in a leading comment in your code as well.

Please submit everything I need to compile your program. This will, at minimum, be three files: the client (hist.c), the server (hserver.c), and a makefile (Makefile).

Submit only what is required. Do not submit rpcgen output files, object files, executables, or temporary editor files.

To submit the assignment, you will create a tar or zip archive using 7-zip of all the files that are needed to compile the program, (including the makefile).

Hand the assignment in using the Blackboard submission system. The project name for this assignment is **History RPC**. Before sending the file to me, make sure that all the components are there. If I can't compile any part, you will get **no** credit.