Your task is to implement an "ident" server similar to what is specified in RFC 1413. You may implement this server in any of the following languages: Perl, Python, PHP, Ruby, C, C++.

Please treat this project as if it were a production-level project that will be run on a large number for years to come. You are welcome to use any freely-available open-source library to help you accomplish this task.

Instead of the RFC 1413 reply, the username that the program should reply with depends on the command line parameters passed. The "operating system" returned should always be "UNIX". The reply type must be chosen by specifying one of the following command-line parameters:

* random
  Replies with the username of a randomly-chosen Unix user on the host running your program.

* uid
  Replies with the username of the user whose UID is equal to the "client port" parameter specified by the ident client. If there is no user matching the requested UID, the program should reply with an RFC 1413 compliant "NO-USER" error.

* always <name>
  Always replies with the specified name.

The program should format the reply exactly as specified in RFC 1413, however the username returned should be as specified above. You do not need to determine the owner of the port specified by the client.

If the program is run with no parameters, it should default to "random". If an unknown parameter is specified, the -always parameter lacks a "name", or more than one of the above parameters are specified, the program should exit with an error.

Your program must also meet the following criteria:

* Runs on Linux, though portability to other operating systems is a plus.

* Accepts and services multiple concurrent connections.

* Performs basic validation of the client's request. You do not need to follow the RFC exactly, but implement the validation procedures you feel are necessary. If the client sends you an invalid request, you must disconnect the client (and only that client) immediately without sending a reply.

Your program does NOT require the following:

* "man" page or external documentation.

* Implementation of a "timeout" as specified in section 2 of RFC 1413.

* Handling of non- 7-bit ASCII characters.

You do not need to provide documentation or a "man"
page. Source code will be judged mainly on maintainability,
security, functionality (does it meet all stated requirements?),
and stability. Speed of execution is not important for this task.

This task should take no more than 48 hours or more than 200 lines of
code. Brevity will not be judged, the above is only given as a
guideline.