Working with Redis

What you need to know/learn:

- Redis commands
- Basic JavaScript for node.js
- Yargs for node.js
- Basic PHP
- An understanding of how HTML forms handle posts and gets
- phpredis
- JSON encoding in PHP

1) PubSub (using node.js)

   a) Using the examples in /home/ubuntu/redis/pubsub as a guide, create two node.js programs that will publish and subscribe using Redis. The publisher should be able to publish to any channel. The subscriber should be able to listen to any channel.

   b) Assume there is a live channel called "emergency." A subscriber listening to any channel other than "emergency" should examine the key or the value of any messages it receives. If it detects the string "urgent" anywhere in the key or the value, it should turn around and publish the message to the emergency channel.

   c) In this way, subscribers listening to the emergency channel will receive both messages intended for this channel as well as messages that probably should have been sent to this channel.

2) Storing values in Redis (using php)

   a) We are all sharing the Redis database. Any permanent storage of keys, needs to use your Rowan userid to function like a namespace, e.g., myersjac:genres

   b) You are going to manipulate account data by building a PHP backend on the database server for accountRegistration.php at: http://elvis.rowan.edu/~myersjac/db. By clicking Submit, you will see the values of the fields POSTED from the form.

   c) The PHP file you will build will be called processAccount.php and it will perform the following functions:

      i) Add any unique genres to a Redis set using a key based on your userid, e.g., `myersjac:genres`

      ii) Add the account profile information (as JSON) to a Redis key, e.g., `myersjac:10d2e5`, where the first part of the key is your userid and the second part of the key is the account number (here 10d2e5).

   You will keep track of the status of all Redis transactions by capturing and displaying Redis's response. Also, add-on commands from modules, such as JSON.SET, do not have a built-in Redis object method like "set." For these commands, you must use `$redis->rawCommand()` where the command name and all command parameters are used as parameters of the `rawCommand()` method.

```
$status  = $redis->set("tutorial-name", "Redis tutorial");
```

d) The database server is not a web server, so you will have to simulate execution by running PHP on the server with the PHP compiler. Your PHP code must include the following:

```
 // if started from commandline, wrap parameters to $_POST
if (!isset($_SERVER["HTTP_HOST"])) {
  parse_str($argv[1], $_POST);
 }
```

What this does is read PHP command line arguments and load them into the $_POST associative array if the PHP is not running on a web server (i.e., no HTTP_HOST). If the same PHP was invoked with a POST operation on a web server, the $_POST array would be automatically populated.

Then, there are two ways to run the PHP on the server. Consider this example in /home/ubuntu/php:

```
$ php redisSetup.php "account=myersjac&action=delete"
Processing account myersjac
Action: delete
```

Or, the PHP arguments could be stored in a command file **on one line** as in accountInput.txt

```
$ cat accountInput.txt
account=myersjac&action=Delete
```

Then you can run PHP from the command line this way:

```
$ cat accountInput.txt | xargs php redisSetup.php
Processing account myersjac
Action: Delete
```

3) Retrieving Redis values

a) Now that you have stored data in redis, write another PHP file called retrieveInfo.php that will work with posted data. This time the posted querystring will be the key to the Redis account information, e.g., `myersjac:10d2e5` (but use YOUR userid, not myersjac).

b) Your PHP file should retrieve and display the name and the phone number for the account.

c) Your PHP file should also display all of the genres that the users have registered for.

4) Communicating using the RESP protocol

a) Copy the file /home/ubuntu/redis/resp/RedisSocket.java to your directory

b) Modify the file so that the fourth option will do a Redis SET. It should set the name of your favorite animal using a key with your userid, e.g. myersjac:faveanimal.

The fifth option should do a Redis GET and retrieve your favorite animal.