Course Project: Robot Proxy Client and Server

Write an HTTP client that can perform HTTP get requests for 7 data sources on prescribed hostname/ports. This HTTP client must also function as an UDP server on a prescribed port. This port is identified via a command line parameter. The UDP server provides UDP clients with the result of the appropriate HTTP get request.

Write a UDP client that saves the content that it receives (images and text). The hostname and port for the UDP server are defined by command line parameters for this client program as well.

Implement the protocol that enables communication between UDP client and server. As a class, you must decide upon the communication protocol since your clients and servers will need to communicate other groups as well.

In this project, your client must perform the following task:

1. Make a robot traverse the path of a two regular polygons in sequence.
2. The polygons are length L and order N and N-1 (where L and N are integers provided by the user in addition to the other parameters).
3. 4 <= N <= 8.

This is the template for a robot proxy. To test whether this works, you will take a picture and get the position[s] of the robot. Then you will move the robot and take another picture/position.

Replace 100000 in the image URI and the id with the details I will give your group.

example HTTP GET request	**TYPE**
`http://130.127.192.62:8081/snapshot?topic=/robot_100000/image?width=600?height=500`	`image`
`http://130.127.192.62:8082/state?id=invalid`	`GPS`
`http://130.127.192.62:8084/state?id=invalid`	`dGPS`
`http://130.127.192.62:8083/state?id=invalid`	`lasers`
`http://130.127.192.62:8082/twist?id=invalid&lx=4`	`move`
`http://130.127.192.62:8082/twist?id=invalid&az=30`	`turn`
`http://130.127.192.62:8082/twist?id=invalid&lx=0`	`stop`

Your server program needs to take as a required parameter the robot id, the robot hostname, and the UDP server port. This program issues http get requests and provides the returned data to the issuing UDP client. The client program needs to be started with the hostname of the UDP server, and the UDP server port . The UDP client program is to issue the following sequence of requests.

Every time that you receive a GPS or DGPS value, you write the response to stdout.
Create a short video (3-5 minutes) that is describes the major components of your programs, and confirms that your team has successfully completed the assignment.

Submission Instructions

When your project is complete, archive your source materials, using the following command:

```
$ tar cvzf cuid1_cuid2_cuid3.tgz README Makefile {list of source files}
```

The Makefile should build your program (by running make).

The README file should include the names of all members of your group, a short description of your project, and any other comments you think are relevant to the grading. It should have five clearly labeled sections titled with KNOWN PROBLEMS, and DESIGN. The KNOWN PROBLEMS section should include a specific description of all known problems or program limitations. This will not hurt your grade, and may improve it. I am more sympathetic when students have thoroughly tested their code and clearly understand their shortcomings, and less so when I find bugs that students are either ignorant of or have tried to hide. The DESIGN section should include a short description of how you designed your layers (especially anything you thought was interesting or clever about your solution). You should also include references to any materials that you referred to while working on the project. Please do not include special instructions for compilation. The compilation and running of the tests will be automated (see details below) and your instructions will not be followed.

Please make sure you include only your source files, not the object files.

We will test on the cslinux lab machines. Make sure it runs there!

5 Grading

Your project will be graded based on the results of functional testing and the design of your code. We will run several tests to make sure it works properly and correctly handles various error conditions. Specifically, you will receive 10% credit if your code successfully compiles, and 10% for code style and readability.

Your source materials should be readable, properly documented and use good coding practices (avoid magic numbers, use appropriately-named variables, etc). You code should be -Wall clean (you should not have any warnings during compilation). Our testing of your code will be thorough. Be sure you test your application thoroughly.

6 Collaboration

You will be assigned to groups of 3-4 students for this project. Even though you work with a partner, you should submit individually (be sure to note who you worked with in your README file). You may also, of course, discuss the project with our course TA, and you may discuss conceptual issues related to the project that we have already discussed in lecture. Collaborating with anyone else on this project, in any other manner will be treated as academic misconduct.

d mclaughlin	**p clark**	n meade	**j savold**
s lemieux	**o mayar**	r baxley	**m clyburn**
a kajal	**b woo**	j dixon	**j wood**
b doher	c priester	**k martin**	r orr
j henline	s malik	**z derose**	b raulerson
d flaaten	b george	**m pfister**	j moore
m lowder	**j marquez**	w newberry	
a basha	**bgreen**	z deng	
b Mitchell	**s nix**	s platt	
	d best	g watkins	
m deangelis	r townsend	**d johnson**	a gupta
t slaton	t brinson	**c davis**	c pickard
j dawkins	w black	**s southard**	a brennan
	l davis	**b briggs**	