Programming Project II
CS475
Randall DUE Oct 12, 2017

For Programming Project II you will be writing a weather server/client application. The weather information will be taken from wunderground.com.

Weather information can be received from this website by adding:

Weather Underground API Key Sign up. FREE
https://www.wunderground.com/weather/api/

API Request Examples:
https://www.wunderground.com/weather/api/d/docs?d=data/index

Example Evansville Indiana XML

http://api.wunderground.com/api/YOUR_KEY/forecast/geolookup/conditions/q/IN/evansville.xml

Response

<response> <version>0.1</version> <termsofService>
http://www.wunderground.com/weather/api/d/terms.html
</termsofService> <features> <feature>forecast</feature>
<feature>geolookup</feature> <feature>conditions</feature>
</features> <location> <type>CITY</type> <country>US</country>
<country_iso3166>US</country_iso3166>
<country_name>USA</country_name> <state>IN</state>
<city>Evansville</city> <tz_short>CDT</tz_short>
<tz_long>America/Chicago</tz_long> <lat>37.97457886</lat> <lon>-87.57350159</lon> <zip>47701</zip> <magic>1</magic>
<wmo>99999</wmo> <l>/q/zmw:47701.1.99999</l>
<requesturl>US/IN/Evansville.html</requesturl>
<wuiurl>http://www.wunderground.com/US/IN/Evansville.html</wuiurl
> <nearby_weather_stations> <airport> <station>
<city>Evansville</city> <state>IN</state> <country>US</country>
<icao>KEVV</icao> <lat>38.04305649</lat> <lon>-87.52027893</lon>
</station> <station>

…
<degrees>289</degrees> </avewind> <avehumidity>53</avehumidity>
<maxhumidity>0</maxhumidity> <minhumidity>0</minhumidity>
</forecastday> </forecastdays> </simpleforecast> </forecast>
</response>

Your Server Program should allow a client application to make a request to receive weather data from at least 3 different cities. You can provide as much information as you would like but you must include the current conditions and 5 day forcast.

To start your client application it should take 2 arguments the server name, and server port number. Your client application should use gethostbyname to resolve the server IP.

Your Client application should provide a text based menu that will provide the user a list of available cities. The User should be prompted to make a selection, Once the selection has been made the client should contact your server to retrieve the current weather data for the city selected. The Client should then disconnect from the server. The client should then present the current weather data in a clear form to the user. At this point the user should be asked if they would like to check the weather of another city or exit.

Making the html request from c++ is simple.

You will be connected to the wunderground server just like you have connected to any other server, but you will connect via port 80. Once you are connected you will perform a Get command on the server which will return the XML code to your program you will need to programmatically retrieve the data and parse out the information that you need.

Get: The following 2 lines of code request the data from the weather server once a socket has been created with the server. The first argument "weather" below is the socket descriptor that was created with the server.

```
send(socketDesc, "GET /api/YOUR
KEY/forecast/geolookup/conditions/q/IN/evansville.xml
HTTP/1.0\r\nHost: api.wunderground.com\r\n\r\n",strlen("GET /api/YOUR
KEY/forecast/geolookup/conditions/q/IN/evansville.xml
HTTP/1.0\r\nHost: api.wunderground.com\r\n\r\n"), 0);
```

The final portion of project 2 will be to write an application protocol for your application. The Protocol should explain data transfer, and the client/server relationship.