UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

INTRODUCTION TO MODERN C++

Lecture 8

Rémi Géraud

March 31, 2016

École Normale Supérieure de Paris

Lecture 8
Networking with cURL and libxml.

NETWORK BASICS

NETWORK BASICS: OSI LAYERS

Internet works by a succession of abstraction layers :

1. (Physical) Voltage variations / EM waves / IR
2. (Data link) Bits
3. (Network) Addresses
4. (Transport) Guarantees of delivery
5. (Application) File download, Tweet sending, etc.

We need all the layers to make the cake.

NETWORK BASICS: ADDRESSES

Internet exists by solving the addressing problem:

How can I send you a message (and receive an answer) if I don't know where you are?

To that end, every user of the internet has an IP address, of the form

`192.168.0.1`

or, more recently

`2001:0db8:0000:0042:0000:8a2e:0370:7334`

NETWORK BASICS: DOMAIN NAMES

For convenience, we may sometimes use domain names, e.g.

`www.google.com`

instead of the more familiar

`216.58.218.238`

The correspondence between domain names and IP addresses is updated an maintained on a volunteer basis. Again, it's a convenience.

Remark: Domain names are read right-to-left.

THE HTTP PROTOCOL

THE HTTP PROTOCOL: URLS

Internet can be used to do a lot of things: e-mails, videos, VoIP, etc.
In what follows we'll focus just on one : websites.

Websites are served by the HTTP protocol and behave essentially like remote files

$$\underbrace{\texttt{www.reddit.com}}_{\text{domain name}}\,/\,\underbrace{\texttt{r/damnthatsinteresting}}_{\text{resource location}}$$

The complete sentence is known as an "universal resource locator" (URL).

THE HTTP PROTOCOL: FILES

Most files served over HTTP are glorified text files.

Notable exceptions: images and video content

While some websites assume the use of a web browser to be usable, this is not true and there is a trend towards simplicity again (e.g. REST APIs).

THE HTTP PROTOCOL: HTML

The *de facto* standard for webpages is the HTML file format.

It looks like this:

```
<html>
  <head>
    <title>The title of my webpage</title>
  </head>
  <body>
    <h1>Hello HTML!</h1>
    <p>This is a simple page</p>
  </body>
</html>
```

You can also try any website (Ctrl+U on Chrome).

The HTTP protocol: HTML

The problem with HTML is: How do we extract information from it?

Hint: It's not simple. We'll deal with that in a moment.

libcurl and libxml2

LIBCURL

To take care of HTTP communication, we will use `libcurl`.

(It is possible to do it manually, but why would we do that now?)

All `libcurl` commands start with `curl_` and the final program should be linked with option `-lcurl`.

Following the overall direction of this course, `libcurl` is a portable, free, cross-platform library that is widely available.

```
apt-get install libcurl4-gnutls-dev
```

(or, if you're curious, compile and install from source)

LIBXML2

To take care of XML / HTML parsing, we will use `libxml2`.

(It is possible to do it manually, but why would we do that now?)

`libxml2` uses the simple XML API (SAX) format. The final program should be linked with option `-lxml2`.

Again, `libxml2` is a portable, free, cross-platform library that is widely available.

```
apt-get install libxml2-dev
```

(or, if you're curious, compile and install from source)

Questions?

Lab : Fetching, parsing, ???, Profit!