Datanet 2015

Assignment introduction

Kenneth Skovhede

Overview

- 4 assignments
- Each assignment scores 0-10 points
- You need 24 points to qualify for the exam
- No re-submissions

Structure of assignments

1. Read/research
2. Implement
3. Write report
4. Hand in

1. Read and research

- Read the assignment
- Find required extra material
- Make a hypothesis for the assignment
  - I.e. describe what you are examining
- Write it down

+

2. Implementation

- Make an implementation that allows you test your hypothesis
- Most likely you will have to go back and research
- You do not get points for source code!
  - But try to make it readable so we can verify your claims
- The implementation is a tool to help you understand the details, complexity and imprecisions of a real-world example

3. Report

- You get points for the report!
  - Read the "your report must contain …" section!
- Your report must be in the ACM standard template
- Try to use this format:
  - Abstract
  - Introduction
  - Design
  - Implementation
  - Experiments
  - Results

4. Handing in

- Deadlines
  - 1st assignment: 2015-04-28 23:55 (11:55pm)
  - 2nd assignment: 2015-05-08 23:55 (11:55pm)
  - 3rd assignment: 2015-05-26 23:55 (11:55pm)
  - 4th assignment: 2015-06-04 23:55 (11:55pm)
- Plagiarism
  - I check if you copy stuff (i.e. from Wikipedia)
  - If you copy something, make sure you quote (or reference)
  - You may work in groups, but be sure you hand in your own report

1st assignment: HTTP Client

- Build a rough version of `wget` / `curl`
  - Learn sockets in practice
  - Learn HTTP in practice
  - Measure latency, topology and bandwidth

URLs

- Many features in a string:
  - scheme://user:password@host:port/path?query_string#fragment_id
- Most are optional:
  - scheme://host/path
- An example:
  - http://www.diku.dk/index.html

+ HTTP request message

carriage return character

line-feed character

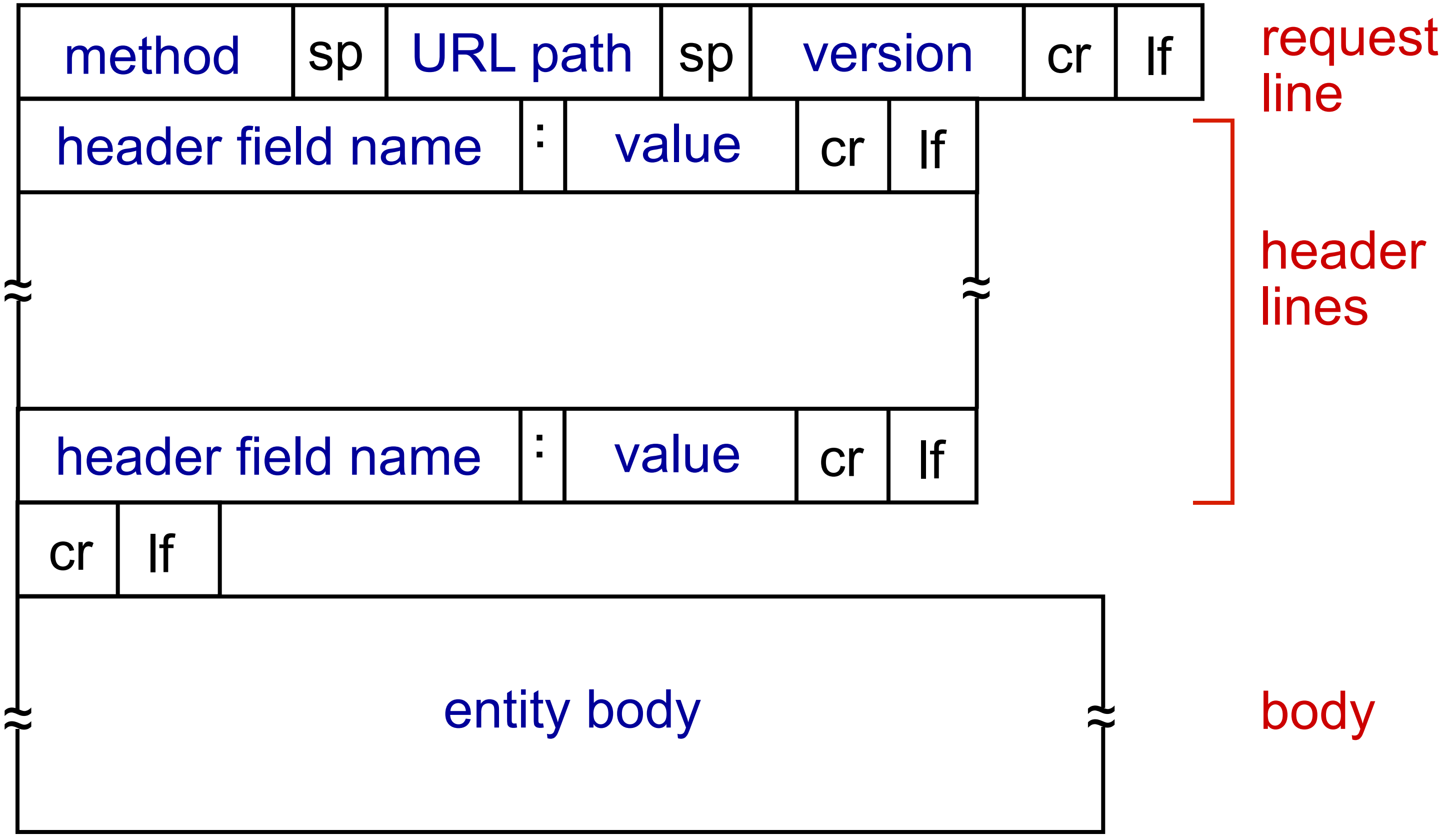
request line
(GET, POST,
HEAD commands)

header
lines

carriage return,
line feed at start
of line indicates
end of header lines

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

HTTP request message: general format



method sp URL path sp version cr lf

header field name : value cr lf

header field name : value cr lf

cr lf

entity body

request line

header lines

body

Sockets

- An example in Python:

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("www.diku.dk", 80))
s.send(
    "GET / HTTP/1.1\r\n" +
    "Host: www.diku.dk\r\n" +
    "Connection: close\r\n" +
    "\r\n")

msg = s.recv(4096)
while msg != '':
    print msg
    msg = s.recv(4096)
s.close()
```

HTTP Response Message

status line (protocol status code status phrase)

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT
   \r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html;
   charset=ISO-8859-1\r\n
\r\n
data data data data data ...
```

header lines

data, e.g., requested HTML file

Too easy?

Use the time wisely

- Time to learn LaTeX?
- Familiarize yourself with the ACM template
- Think in academic writing style
  - Introduction
  - Hypothesis
  - Experiments
  - Results
  - Conclusion
- Try out Amazon EC2

Collaboration rules

- You may work in groups of 2-4 persons
- But you must hand in an *individual* report
- If you cite/copy/reference something, you *MUST* state the source, i.e.:
  - "Goats are among the earliest animals domesticated by humans." [Wikipedia]
- Failure to follow this is considered cheating, and may have repercussions

Questions?