

Lab2: Net Ninny

Questions?

- How to handle 2-3-... word combinations? *find() in the C++ library.*
- Forking really necessary? Or threads okay? *Threads are okay.*
- State machine by character to parse the messages? *Yes.*

Teaching session

Advice

- use one port per msg on the client side (recv)
- backlog: 10
- if sending failed (check first), send again
- use the header **Connection:** `close` and enforce it on HTTP 1.1!
- look at **Content-type:** `text` in the response msg before parsing
- only filter urls and text!
- if abusive content found, send 301, then new request for the error page, then forward

Setup

- Set the browser to 127.0.0.1 + port number
- use port > 1024
- use only IPv4

Architecture

Server	Client
get GET request + forward response to client url filtering	send GET to server + get response content filtering

What needs to be done

Done

- Threads
- Server able to get requests
- Client able to get requests
- Client able to send request
- Client able to get response
- Constants file
- Constant content redirect response
- Exceptions
- Request as abstract object (get) with `to_string` method
- Response class with `to_string` method
- Separate TCP and Server files
- for the CHttpRequest: method `string getHost()`
- for the CHttpResponse: method `string getContentType(const string& header)`

Martin

- for the CHttpResponse: method `bool isTextContent(const string& header)` (optional)
- replace `Connection: keep-alive` by `Connection: close` in the request
- bind server and client in the threads

Lena

- Check for errors when the server uses the client
- Write report