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	send GET to server + get response
url filtering	content filtering

What needs to be done

Interface Client: CHTTPResponse &mainClient(CHTTPRequest &request) // CHHTPRequest always valid // we can then use request.toString().c_str() // CHTTP *response = new CHTTPResponse(header, content)

Martin

- constants
- parse the http: method, url, version, etc.
- object http_request as an abstract object + to_string method
- parse for 'Content-type' in the response
- replace 'Connection: keep-alive' by 'Connection: close' in the request
- http response class, with to-string (constructor with header and content)
  - get, set for the content type

Lena

- starting the client part, entirely separated: get IP of the current interface and send hardcoded message to real server + get response
- use String for the response message
- make exceptions
- client: gets a request object and returns a response object
- create response dynamically (new)