HTTP-protocol

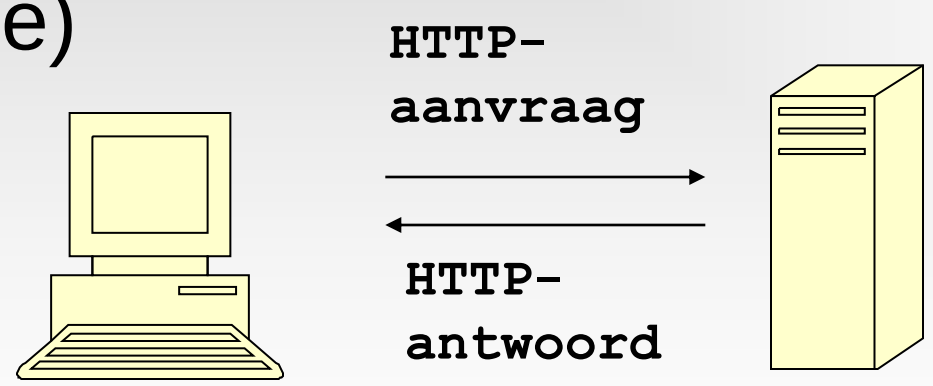
Veerle Ongenae

HTTP-protocol

- Drie officiële versies: 0.9, 1.0 en 1.1
- Hypertext Transfer Protocol
- Gebruik: WWW, …
- HTTP/1.1 beschreven in RFC 2616 (http://www.w3.org/Protocols/rfc2616/rfc2616.txt)
- Meer informatie: http://www.w3.org/Protocols/

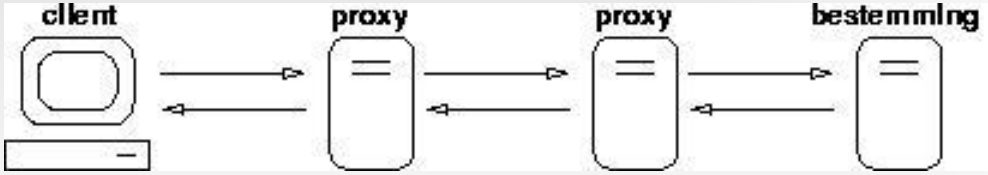
Inleiding

- Client-Server protocol
- Transportprotocol: TCP op poort 80
- Aanvraag-antwoord protocol (request-response)

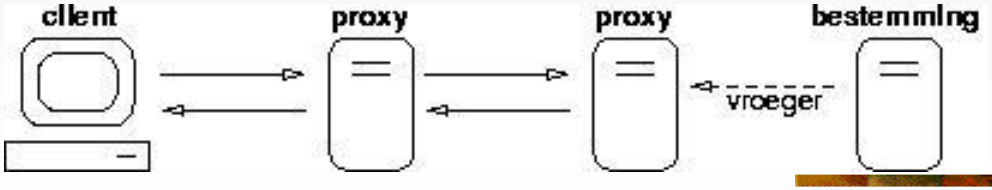


Inleiding

- Aanvraag
  - Rechtstreeks naar server met informatie
  - Via tussenliggende HTTP-servers (proxyservers)



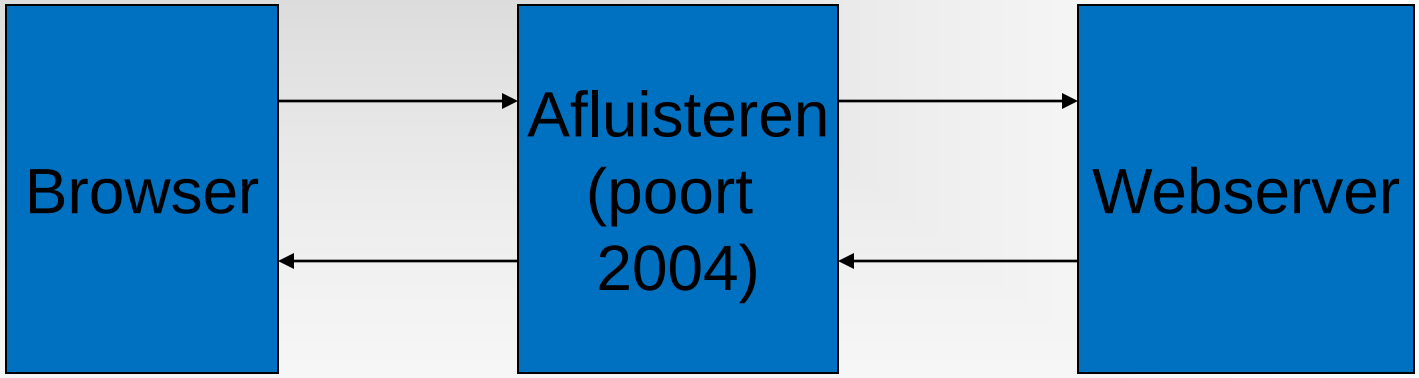
  - Eventueel gecached



Overzicht

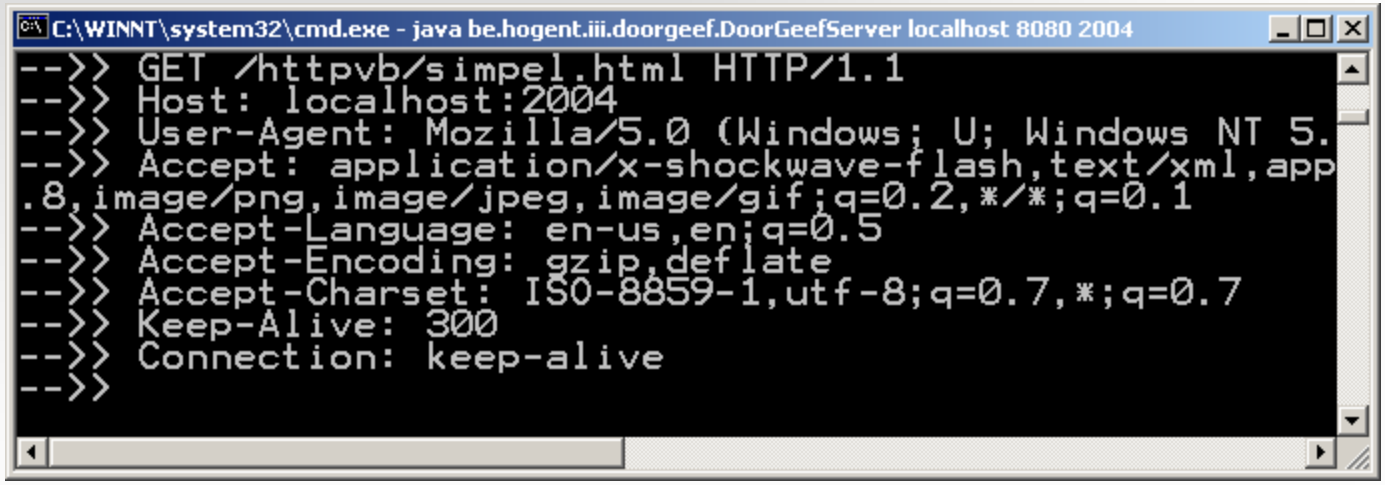
- Structuur HTTP-bericht
- Cookies
- Virtuele Servers
- Wachtwoorden
- Persistente verbindingen

Afluisteren communicatie

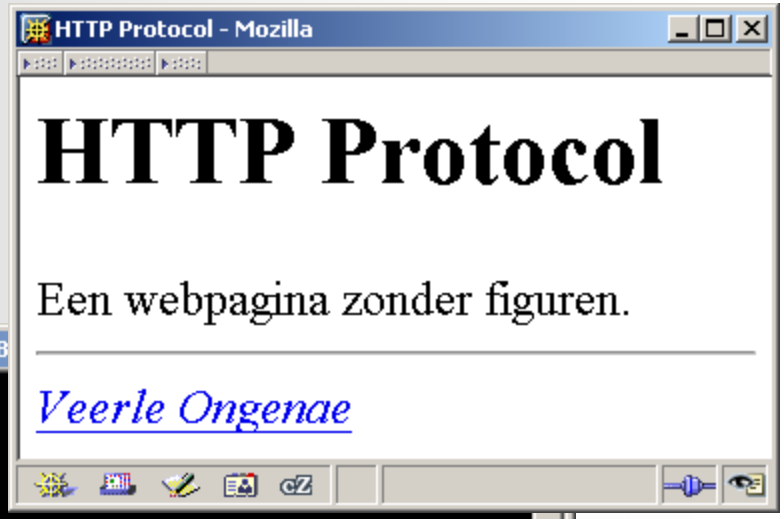


Eenvoudige webpagina: aanvraag

http://localhost:2004/httpvb/simpel.html



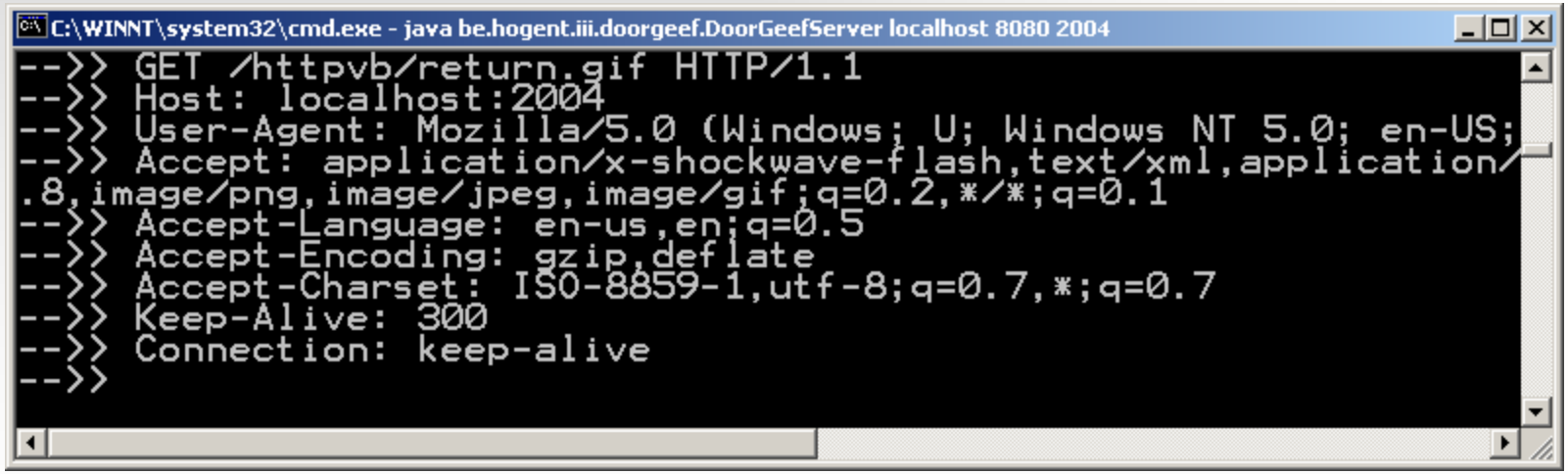
Eenvoudige webpagina: antwoord




```
C:\WINNT\system32\cmd.exe - java be.hogent.iii.doorgeef.DoorGeefServer localhost 8
<<-- HTTP/1.1 200 OK
<<-- Content-Type: text/html
<<-- Content-Length: 301
<<-- ETag: "301-1096366927155"
<<-- Date: Tue, 20 Sep 2005 08:33:25 GMT
<<-- Server: Apache Tomcat/4.0.5 (HTTP/1.1 Connector)
<<-- Last-Modified: Tue, 28 Sep 2004 10:22:07 GMT
<<--
<<-- <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<<-- <html>
<<--   <head>
<<--     <title>HTTP Protocol</title>
<<--   </head>
<<--
<<--   <body>
<<--     <h1>HTTP Protocol</h1>
<<--     Een webpagina zonder figuren.
<<--     <hr>
<<--     <address><a href="mailto:Veerle.Ongenae@hogent.be">Veer
<<--   </body>
<<-- </html>
```

Figuur: aanvraag

http://localhost:2004/httpvb/return.gif



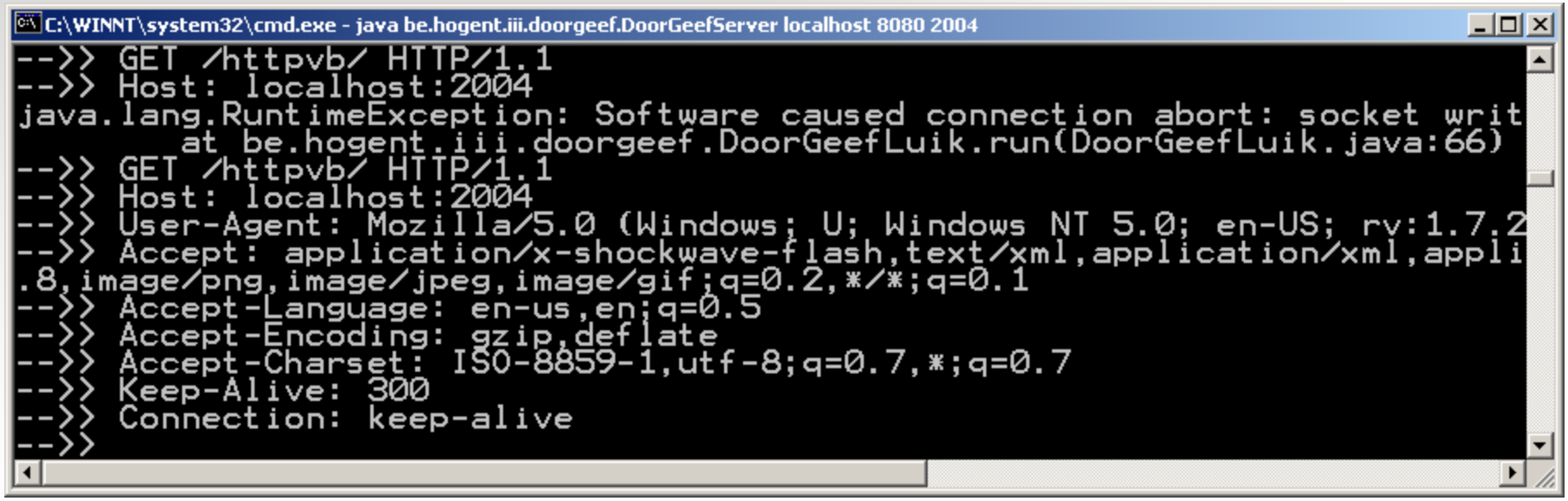
Figuur: antwoord

C:\WINNT\system32\cmd.exe - java be.hogent.iii.doorgeef.DoorGeefServer localhost 8080 2004

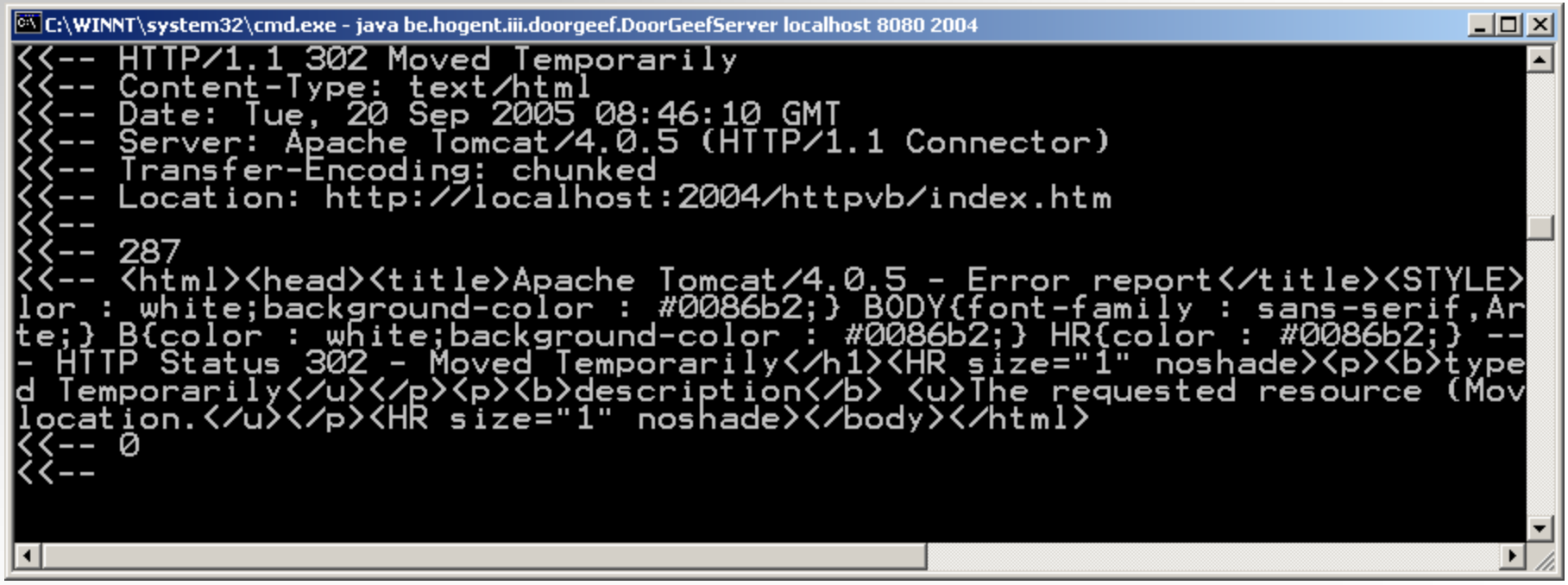
```
<<-- HTTP/1.1 200 OK
<<-- Content-Type: image/gif
<<-- Content-Length: 1231
<<-- ETag: "1231-1032771534000"
<<-- Date: Tue, 20 Sep 2005 08:42:07 GMT
<<-- Server: Apache Tomcat/4.0.5 (HTTP/1.1 Connector)
<<-- Last-Modified: Mon, 23 Sep 2002 08:58:54 GMT
<<--
<<-- GIF89a!!!)))999JJJRRRZZZccckkksss{{{JBB911)!!199!!!ZcJR199B)1s)1!){s
Z1sJB1))!)!s!Rk!!)9skRZR9J!!1B{B{1c{!)BZ)9Z{!1!1J9BJsksZRZ!!!,@HP x13RhN
$ 1Z*<Jh%R)%&@ResiF>ZC<|g|,%/^@h%S1@!44%!\Q5$'@*ea]$r!n8xndY#)V(P
<<-- <'CD&(0"2TQbQG4C!,$TPy]C-$ E<T )W'2rjFT0'G!;G)TU)aC5a}EIuG(;
```

Webpagina met figuur: aanvraag

http://localhost:2004/httpvb/

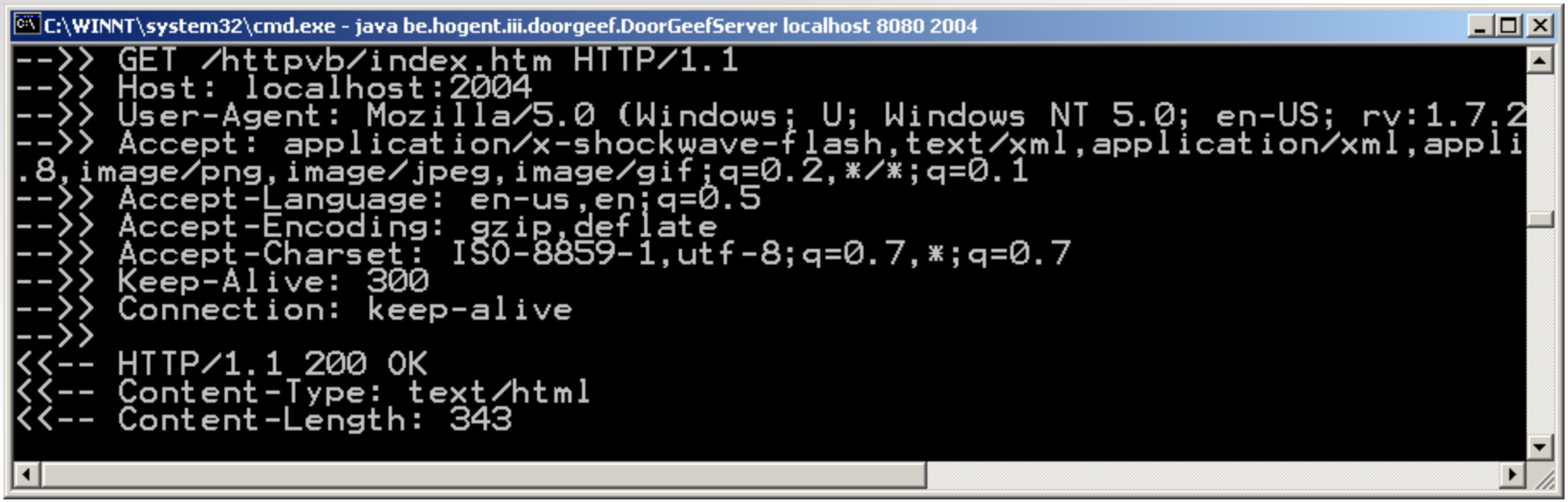


Webpagina met figuur: doorverwijzen naar index.htm

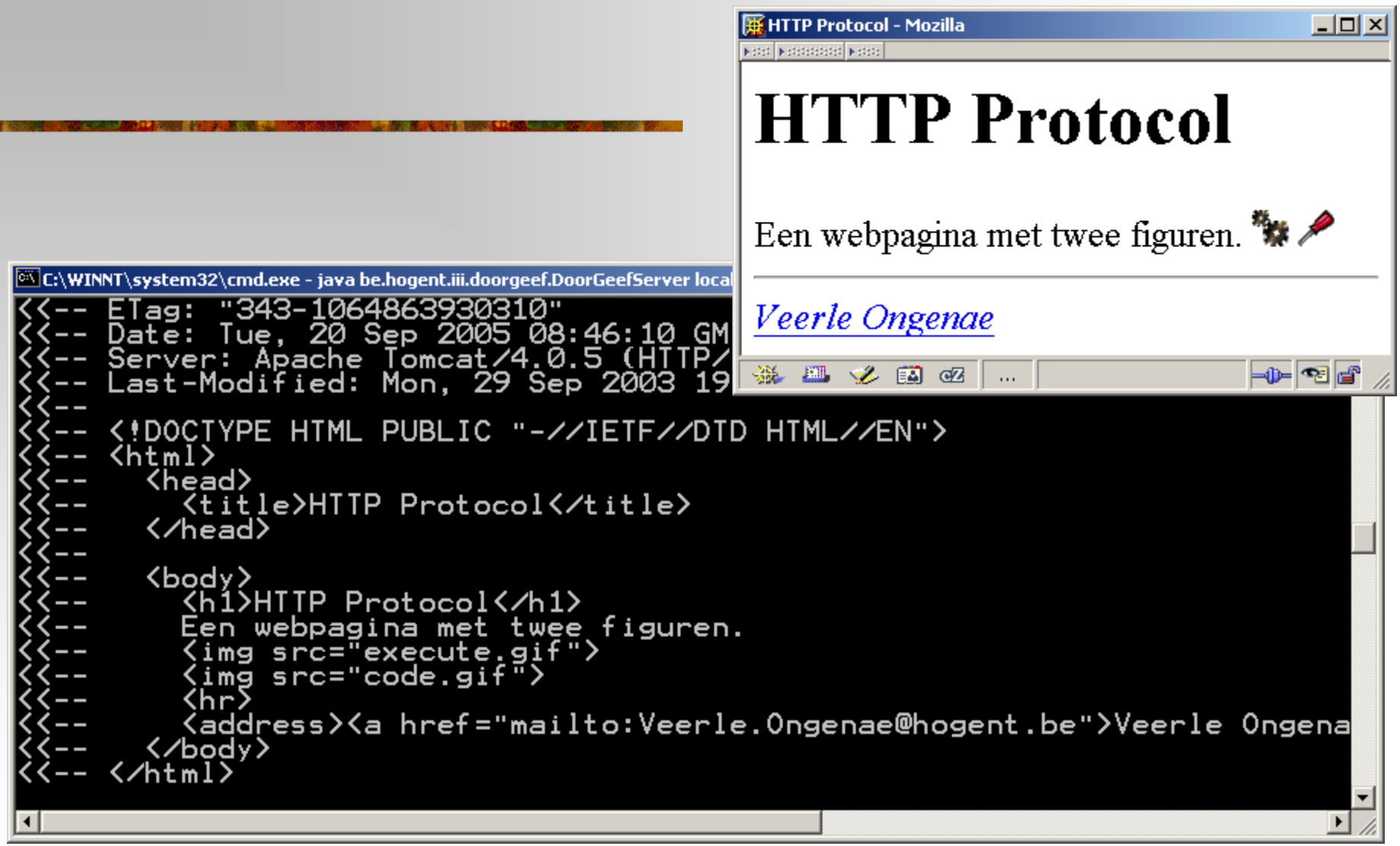


Webpagina met figuur: tweede aanvraag

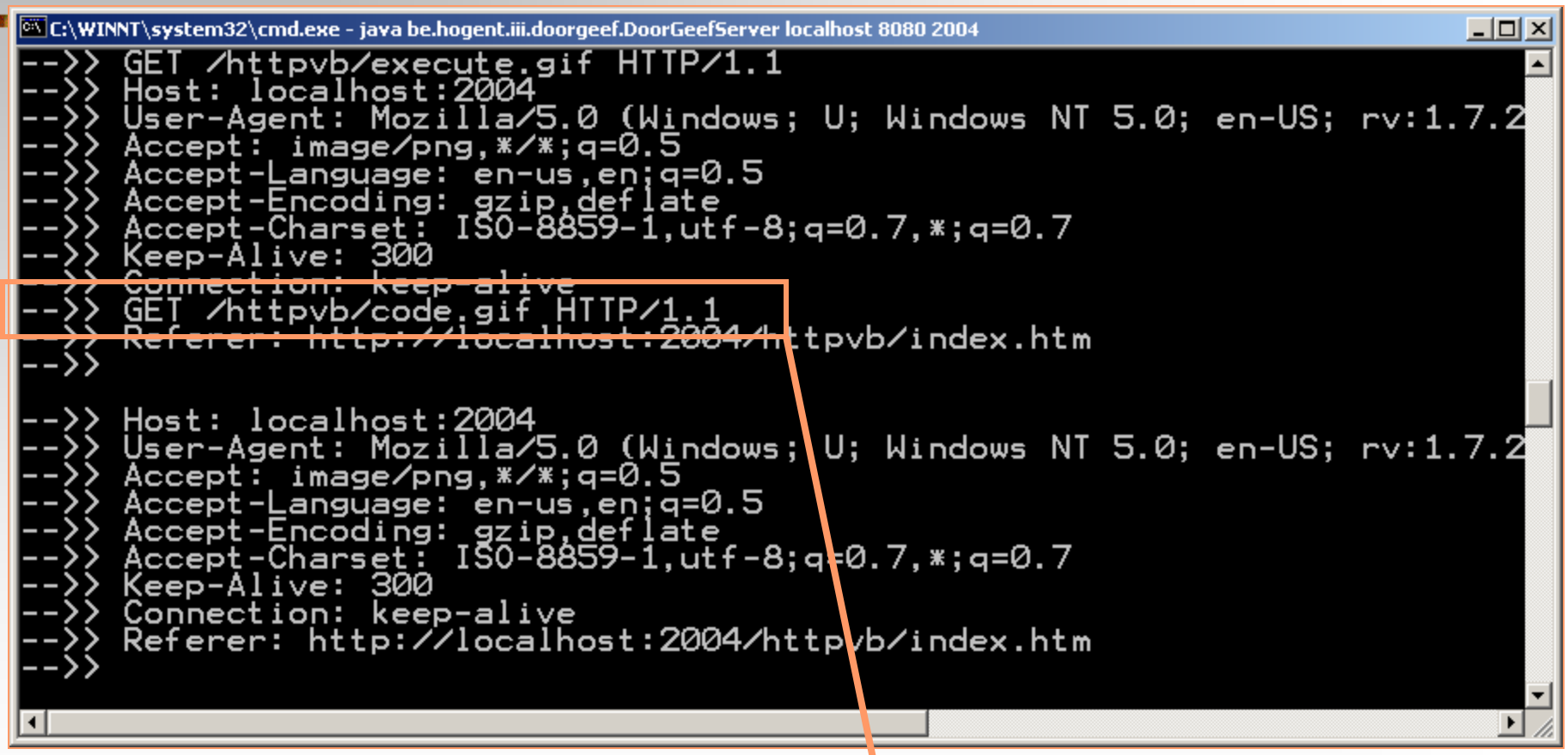
http://localhost:2004/httpvb/index.htm



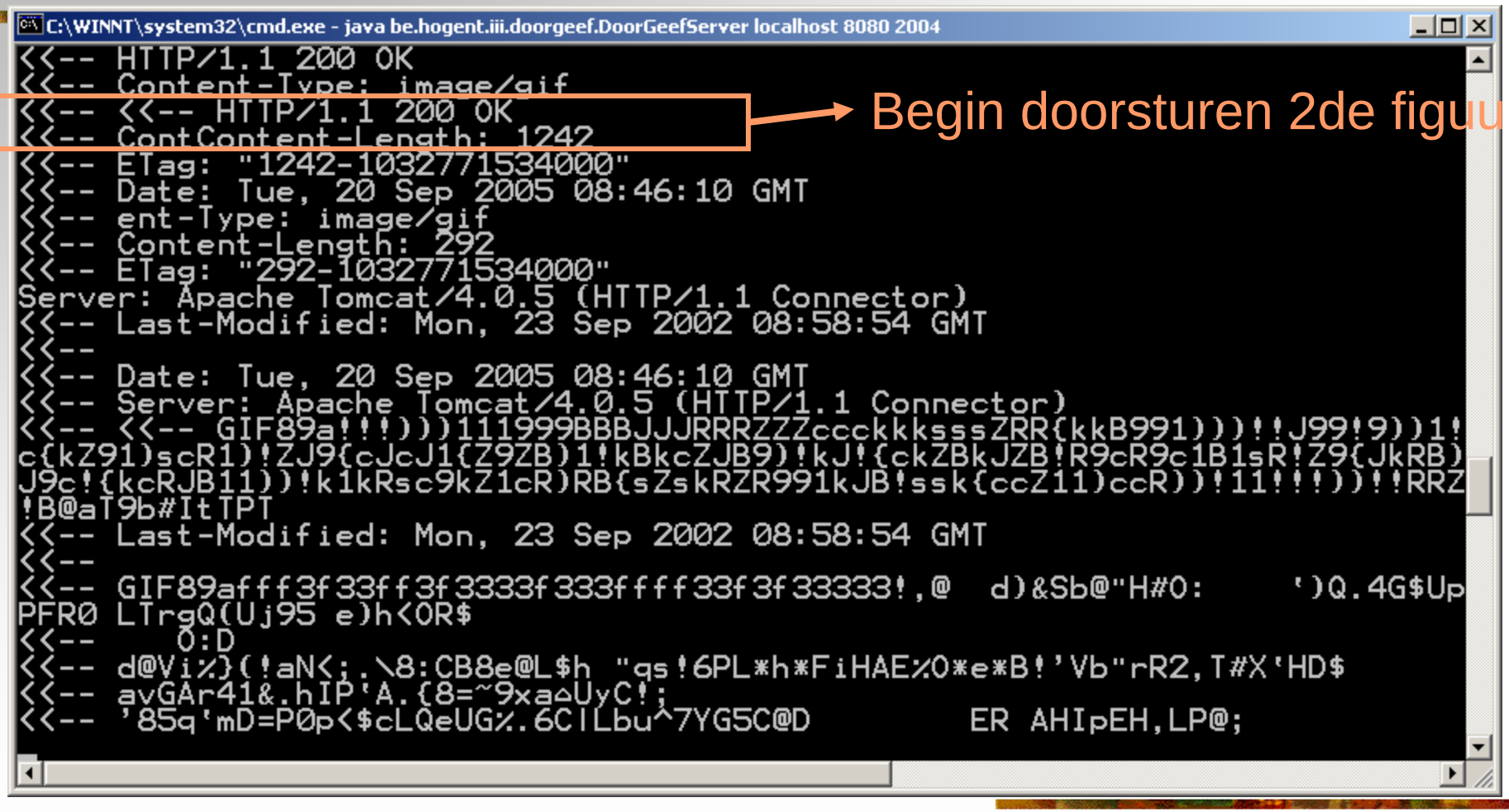
Webpagina met figuur: antwoord



Figuren opvragen



Figuren doorsturen



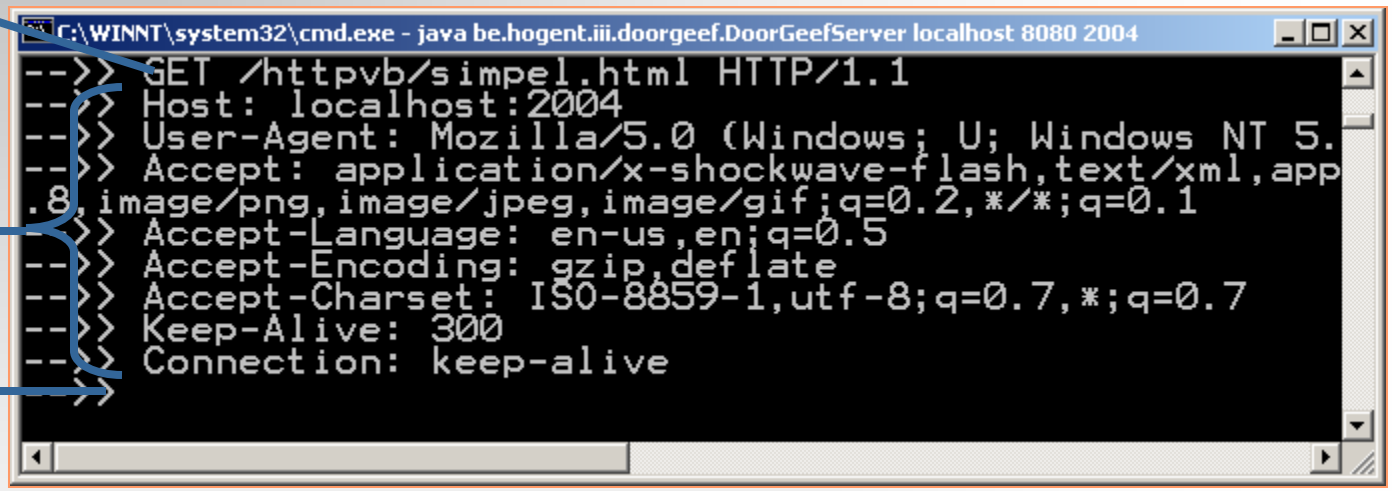
Structuur bericht

Eerste headerlijn

Volgende headerlijnen

Lege lijn

Corpus (optioneel)



Structuur bericht

- Eerste headerlijn
- Volgende headerlijnen
- Lege lijn (CRLF)
- Corpus

Eerste headerlijn

- Verschillend voor
  - HTTP-aanvraag
  - HTTP-antwoord

HTTP-aanvraag

- Eerste headerlijn bestaat uit drie delen gescheiden door een spatie
  - Aanvraagmethode
  - Aanvraag-URI
  - HTTP-versie
- Voorbeeld

```
GET http://gonzo.hogent.be/intranet HTTP/1.1
GET /intranet HTTP/1.1
GET /intranet/welkom.html HTTP/1.1
```

Aanvraagmethodes

- GET
  - Enkel headerlijnen, geen corpus
  - Vaak bestand op server
  - Geen verandering op server
  - Mag gecached
  - URL intikken, link klikken
- HEAD
  - Vraag naar informatie
  - Antwoord bestaat enkel uit headerlijnen

Aanvraagmethodes

- PUT
  - Bestand op server plaatsen
  - Bv. Uploaden eigen webpagina's (i.p.v. FTP)
  - Weinig ondersteund
- POST
  - Laat toe om gegevens aan te passen op server
  - Meestal niet gecached
  - Vaak gebruikt in combinatie met HTML-formulieren
  - Corpus = inhoud formulier

Aanvraag-URI

- Uniform Request Identifier
- Types
  - Absolute URI: vooral aanvragen aan proxyservers
  - Absoluut pad: geen protocol en geen servernaam, naam van de server in een aparte Host-headers

    Host: gonzo.hogent.be

Eerste headerlijn

- Verschillend voor
  - HTTP-aanvraag
  - HTTP-antwoord

HTTP-antwoord

- Eerste headerlijn is statuslijn
- Statuslijn
  - Protocolversie
  - Decimale foutcode
  - Kort tekstbericht
- Voorbeeld

  HTTP/1.1 200 OK

  HTTP/1.1 404 Not Found

Foutcodes

- Eerste cijfer geeft type status aan
- 1??: Informatief
- 2??: Succes
- 3??: Omleiding
- 4??: Fout bij client
- 5??: Fout bij server

Veel gebruikte statuscodes

- 200 OK: aanvraag gelukt

- 301 Moved Permanently
- 302 Found / Moved Temporarily
- 303 See Other

  Aanvraag gelukt, maar moet opnieuw

  Location-header bevat nieuwe lokatie

  Gebruiker merkt normaal niets

Veel gebruikte statuscodes

- 401 Unauthorized: geen toegang
- 403 Forbidden: niet toegankelijk
- 404 Not Found: bestaat niet (meer)

- 501 Not Implemented: aanvraag wordt niet ondersteund door serversoftware

Structuur bericht

- Eerste headerlijn
- Volgende headerlijnen
- Lege lijn (CRLF)
- Corpus

Volgende Headerlijnen

- Algemene vorm

  naam: waarde CRLF
- Waarde over verschillende lijnen:
  - Nieuwe lijn start met spatie of TAB
- Naam: geen onderscheid tussen hoofdletters en kleine letters
- Voorbeelden:

```
Accept: text/html, text/plain,
   image/jpg, image/gif
Host: gonzo.hogent.be
```

Structuur bericht

- Eerste headerlijn
- Volgende headerlijnen
- Lege lijn (CRLF)
- Corpus

Corpus

- Geen interne structuur
- Inhoud vaak gecodeerd mbv MIME (Multipurpose Internet Mail Extensions)

  `Content-type: text/html`
- Begin: blanco-lijn
- Einde:
  - Content-Length-header: lengte in bytes

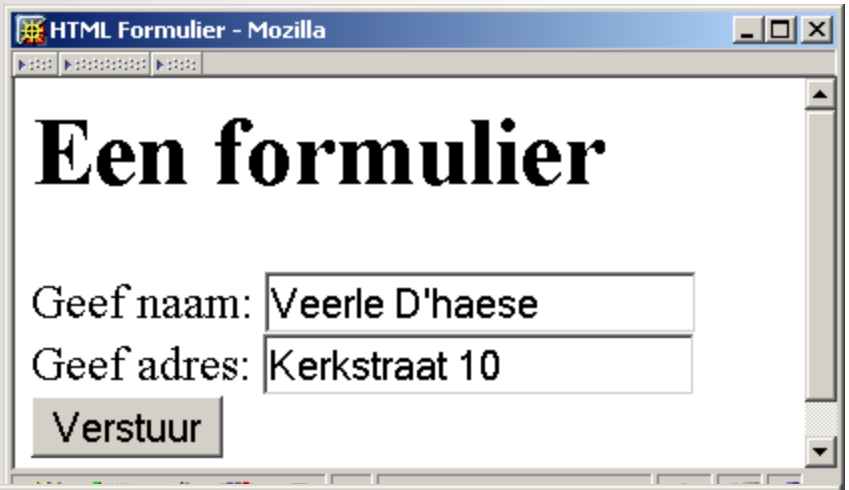
    `Content-Length: 410`
  - Chunked transfer coding

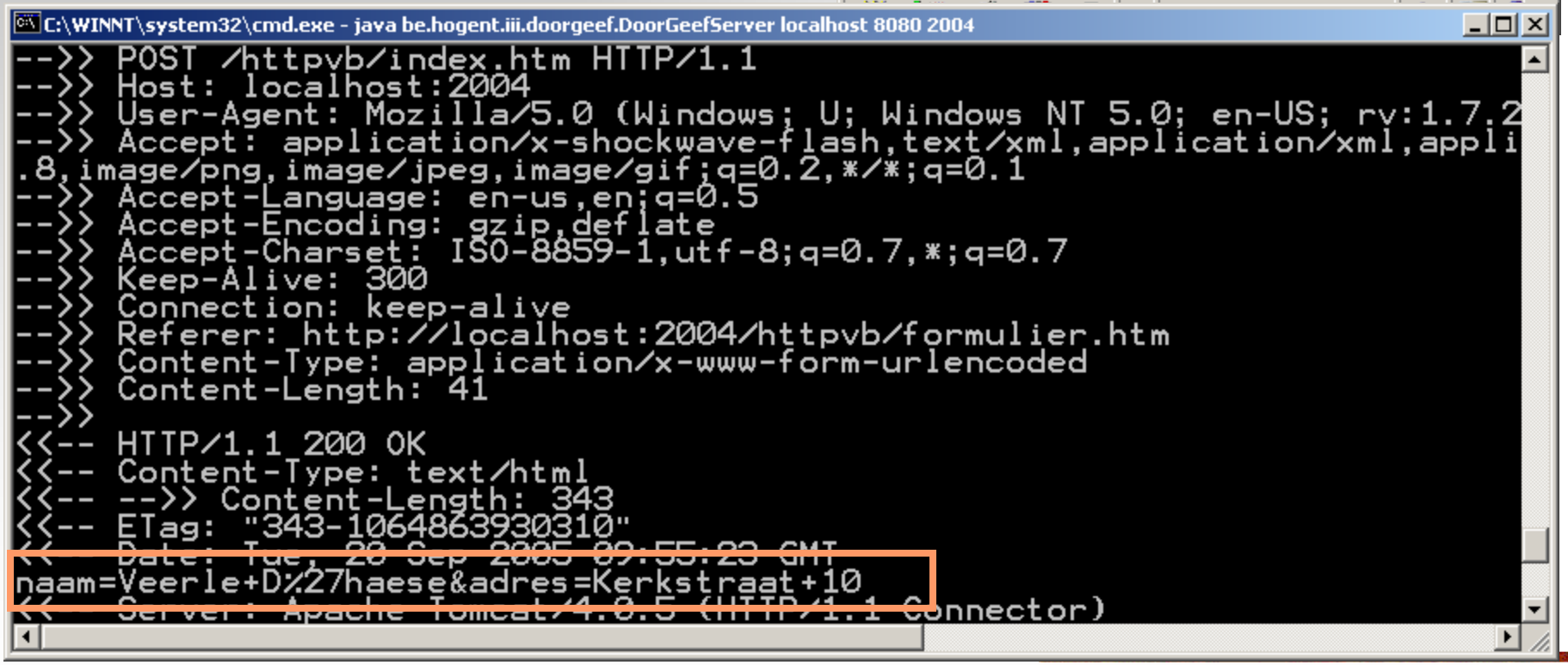
Chunked transfer coding

- Headerlijn

  Transfer-Encoding: chunked
- Corpus verdeeld in verschillende stukken (chunks)
- Chunk
  - Lengte chunk in bytes (hexadecimaal)
  - CRLF
  - Inhoud
- Laatste chunk heeft lengte 0, daarna volgt een lege lijn

Corpus bij formulier





HTML-formulieren

- Tussen <FORM> en </FORM>-tags
  - Attributen van de <FORM>-tag:
    - ACTION: URL “script”
    - METHOD: GET of POST
    - ENCTYPE:
      - Bepaalt structuur van het corpus (HTTP-bericht)
      - application/x-www-form-urlencoded (standaard)
      - multipart/form-data: maakt bestanden opladen mogelijk

Naam/Waarde-paren

- Informatie van een HTML-formulier (form-tag) doorgeven aan een webapplicatie
- Twee types (method-attribuut)
  - GET: querystring in de URL
  - POST: corpus
- Structuur (standaard, NIET bij multipart/form-data):
  - naam=waarde&naam1=waarde1&…

Naam/Waarde-paren: Codering

- codering
  - Speciale tekens: % + 2 hexadecimale cijfers (ASCII-code)
  - Spatie: bovenstaande methode of +

Overzicht

- Structuur Bericht
- Cookies
- Virtuele Servers
- Wachtwoorden
- Persistente verbindingen

Cookies

- HTTP: statusloos
- Een oplossing: cookies
- Klein stukje tekst
- Bewaard door de client
- Teruggestuurd bij elke aanvraag
- In HTTP-headers

Cookies: praktisch

- Bij eerste aanvraag: server stuurt cookie naar de client

SetCookie: NAAM=WAARDE; Domain=DOMEINNAAM; Path=PAD; Max-Age=SECONDEN

SetCookie: uid=vo; Domain=“.rug.ac.be”; Path=“/”; Max-Age=86400

- Bij volgende aanvraag: browser voegt extra header toe

Cookie: uid=vo

Overzicht

- Structuur Bericht
- Cookies
- Virtuele Servers
- Wachtwoorden
- Persistente verbindingen

Virtuele servers

- Eén webserver = HTTP-server voor verschillende instellingen
- Webserver gekend onder verschillende domeinnamen

```
GET / HTTP/1.1
Host: firma1.provider.be
GET / HTTP/1.1
Host: firma2.provider.be
```

Overzicht

- Structuur Bericht
- Cookies
- Virtuele Servers
- Wachtwoorden
- Persistente verbindingen

Wachtwoorden

- Twee wachtwoordschema's
  - Basic: niet veilig, wachtwoorden niet gecodeerd, door bijna alle servers en browsers ondersteund
    - Vb: gonzo
  - Digest: zie cursus beveiliging

Overzicht

- Client: aanvraag URI (niet publiek toegankelijk)
- Server: antwoord 401 Unauthorized

  WWW-Authenticate: Basic
  realm=TripleEyeIntranet

  (wachtwoordschema en rijk)
- Client vraagt gebruiksnaam en paswoord aan gebruiker (indien eerste keer!)

Overzicht

- Client: nieuwe aanvraag zelfde URI met wachtwoordgegevens

  Authorization: Basic
  QWxhZGRpbjpvcGVuIHNlc2FtZQ==

  - Gebruikersnaam en wachtwoord: samengevoegd en omgezet in base64 formaat (ook gebruikt door MIME)
- Client kan bij volgende aanvraag onmiddellijk Authorization-header toevoegen bv. bestanden uit zelfde map of uit subdirectory

Overzicht

- Structuur Bericht
- Cookies
- Virtuele Servers
- Wachtwoorden
- Persistente verbindingen

Persistente verbindingen

- Vroegere versies HTTP: nieuwe connectie voor elke URI
  - Verbinding tussen client en server openen
  - Client stuurt aanvraag met URI erin
  - Server stuurt antwoord
  - Verbinding wordt afgesloten
- Nadeel:
  - achtereenvolgens verschillende verbindingen met zelfde server
  - overlast netwerk

Persistente verbindingen

- HTTP/1.1: meerdere aanvragen over dezelfde connectie
  - Bij elke aanvraag hoort een afzonderlijk antwoord
  - Verbinding wordt niet telkens afgesloten
  - Client kan verschillende aanvragen na elkaar sturen zonder op antwoord te wachten (pipelining)
- Connection-header:

  Connection: close
  - In aanvraag of antwoord
- Time-out