

Project 2

Part 2

cse461

Web Server

- Start with a server that compiles and works correctly on attu.cs
- Then deploy your server to the router by cross-compiling using buildroot

Due: Next Friday (10/31) at 11:59PM

HTTP 1.0

- Client-server request-response protocol
- Application-layer protocol (web-browsers)
- For this assignment, implement a simplified version of HTTP 1.0

HTTP Request/Response

Message	Explanation (required or ignored by the simple web server)
GET / HTTP/1.0	method, request URI, HTTP version (required)
User-Agent: Wget/1.10.2	Request header (ignored)
Accept: */*	Request header (ignored)
Host: www.csc.uvic.ca	Request header (ignored)
Connection: Keep-Alive	Request header (ignored)
	a blank line indicating the end of request headers (required)

Message	Explanation (required or omitted in the simple web server)
HTTP/1.1 200 OK	HTTP version, status code, reason phrase (required)
Date: Fri, 12 Sep ...	response header (omitted)
Server: Apache/2. ...	response header (omitted)
X-Powered-By: PHP/4 ...	response header (omitted)
Connection: c ...	response header (omitted)
Content-Type: text/ ...	response header (omitted)
	a blank line indicating the end of response headers (required)
index.html content	HTTP object(s) returned (required, if any)

HTTP in action

- `wget -d [url]`
 - save file to disk and show HTTP details
- `curl -i [url]`
 - show sent HTTP request and output file to stdout
- `curl -i -v [url]`
 - show HTTP request and response and output file to stdout

Project Requirements

- A web-server that compiles for attu, and compiles for the router with buildroot
- Works with firefox
- Serves files out of the current directory
- Shows a page listing available files for URI '/'
- Handle multiple clients

A Minimal Web Server

- Reads the request line
 - Request Method (case insensitive)
 - Request URI (case sensitive)
 - HTTP version (case insensitive)
- Waits for a new line
- Responds with Bad Request error message if an invalid/incomplete request is received

A Minimal Web Server

- Responds with content of requested file using a valid HTTP 1.0 response message
- Returns a status code, minimally:
 - 200 - OK
 - 400 - Bad Req
 - 404 - Not Found

Full Listing of status codes

```
"200" ; OK
"201" ; Created
"202" ; Accepted
"204" ; No Content
"301" ; Moved Permanently
"302" ; Moved Temporarily
"304" ; Not Modified
"400" ; Bad Request
"401" ; Unauthorized
"403" ; Forbidden
"404" ; Not Found
"500" ; Internal Server Error
"501" ; Not Implemented
"502" ; Bad Gateway
"503" ; Service Unavailable
extension-code
```


TCP Server Outline

- `socket()` - create the socket
- `bind()` - associate to a known port
- `listen()` - wait for incoming connection reqs
- `accept()` - accept a new connection
- `recv()`, `send()` - handle the connection
- `close()` - release allocated resources

TCP Server pseudocode

```
s=socket()  
bind(s)  
listen(s)  
while (1) {  
    s' = accept(s)  
    request = recv(s')  
    response = process(request)  
    send(response)  
    close(s')  
}
```

- **s** is a passive (listening) socket
- **s'** is an active socket

Handling multiple clients

- select

```
FD_ZERO(rfdset)
while (1) {
    rfdset' = rfdset
    ret = select(rfdset', NULL,...)
    if FD_ISSET(s,rfdset') {
        s'=accept(s)
        FD_SET(s',rfdset)
    } else {
        process(rfdset')
    }
}
```

- threads

```
void run(s') {
    request = recv(s')
    response = process(request)
    send(response)
    close(s')
}
while (1) {
    s' = accept(s)
    tid = pthread_create(0,0,run,s')
    pthread_detach(tid) // auto-reap
}
```

Router Note

- The router runs a webserver by default
- To use port 80 you have to disable it:

```
mv /etc/rc.d/S50httpd /etc/rc.d/K50httpd
```

Project Extra Credits

- Implements more than the minimum protocol (e.g. other response codes, POST method..)
- Implements some security
- Serve other dynamic content (besides listing files)
- Other Ideas?

Links

- HTTP 1.0 specs

<http://www.w3.org/Protocols/HTTP/1.0/spec.html>