

ELEC / COMP 177 – Fall 2012

Computer Networking

→ Project #2 – Simple wget

Some slides from Kurose and Ross, *Computer Networking*, 5th Edition

Project #2 – Simple wget

- Implement a simplified version of wget command-line utility
- Demo commands:
 - `wget --server-response http://www.pacific.edu/Documents/registrar/acrobat/2011-2012catalog.pdf`
 - `wget --server-response yahoo.com/privacy`

Demo: wget

```
jshafer@ecs-network:~$ wget --server-response http://www.pacific.edu/Documents/registrars/acrobat/2011-2012catalog.pdf
```

```
--2012-10-18 07:18:00-- http://www.pacific.edu/Documents/registrars/acrobat/2011-2012catalog.pdf
Resolving www.pacific.edu (www.pacific.edu)... 192.168.200.100
Connecting to www.pacific.edu (www.pacific.edu)|192.168.200.100|:80...
connected.
HTTP request sent, awaiting response...
  HTTP/1.1 200 OK
  Content-Type: application/pdf
  Last-Modified: Thu, 07 Jul 2011 23:34:18 GMT
  Accept-Ranges: bytes
  ETag: "845b4e64fe3ccc1:0"
  Server: Microsoft-IIS/7.5
  X-Powered-By: ASP.NET
  Date: Thu, 18 Oct 2012 14:18:00 GMT
  Content-Length: 1819713
Length: 1819713 (1.7M) [application/pdf]
Saving to: `2011-2012catalog.pdf'

100%[=====>] 1,819,713  --.-K/s   in 0.1s

2012-10-18 07:18:00 (13.5 MB/s) - `2011-2012catalog.pdf' saved [1819713/1819713]
```

Demo: HTTP 1.1 at Command Line

```
#> telnet www.google.com 80
```

```
GET / HTTP/1.1
```

```
Host: www.google.com
```

```
Connection: close
```

```
HTTP/1.1 200 OK
```

```
Date: Thu, 18 Oct 2012 00:27:44 GMT
```

```
Expires: -1
```

```
Cache-Control: private, max-age=0
```

```
Content-Type: text/html; charset=ISO-8859-1
```

```
Set-Cookie: <Omitted!>
```

```
Set-Cookie: <Omitted!>
```

```
Server: gws
```

```
X-XSS-Protection: 1; mode=block
```

```
X-Frame-Options: SAMEORIGIN
```

```
Connection: close
```

```
<WEBPAGE FOLLOWS>
```

Project #2 – swget

- No server app this time, just a client!
- Discussion: Argument parsing
 - `swget`
 - `--url=http://www.google.com`
 - `--destdir=/home/shafer/ecpe177`
 - `--verbose`
 - Arguments can occur in any order
 - Tedious to manage yourself
 - Solution: `argp()`!

Project #2 – swget

- Discussion: URL parsing
 - <http://web.pacific.edu/Documents/registrar/acrobat/2011-2012catalog.pdf>
 - Host: web.pacific.edu
 - Path: /Documents/registrar/acrobat/
 - File: 2010-2012catalog.pdf
 - **Where is each used at in this program?**
 - **How do we separate out these pieces?**
 - *Without using a fancy library somebody already built...*

Project #2 – swget

- Discussion: URL parsing
 - <http://www.google.com/>
 - Host: www.google.com
 - Path: ???
 - File: ???
 - **What does the server do here?**
 - **What should swget do when saving to disk?**

Project #2 – swget

- Discussion: HTTP Response Header processing
 - HTTP/1.1 200 OK
Content-Length: 2122207
Content-Type: application/pdf
Server: Microsoft-IIS/6.0
Date: Thu, 28 Oct 2010 16:27:10 GMT
Connection: keep-alive
<Blank line>
<Data starts here>
 - Need to parse the header line by line
 - Each line ends with `\r\n`
 - **Ideas on how to parse?**
 - No need to parse subsequent data – just save it straight to disk

Project #2 – swget

- Implementation strategy
 1. Arguments parsing
 2. URL parsing
 3. Sockets – HTTP request and reply
 - Save full response to disk
 4. HTTP response header parsing
 - Save only *data* response to disk
 5. Testing (compare results with wget)
 6. Extra features (301-redirects), polish, test again

Version Control?

- **Anyone ever used version control?**
- **How are you going to coordinate code updates with your partner?**
- Free subversion or git hosting at <http://offers.assembla.com/free/>
- Free git or mercurial hosting at <https://bitbucket.org/>

Memory Problems?

- Limitless opportunities in C for errors regarding memory
 - Forgetting to free() some dynamic memory
 - Trying to free() dynamic memory more than once
 - Losing a pointer to dynamic memory (memory is “lost”)
 - Accessing array elements past the end of the array
 - Mis-calculating array pointers that miss their desired target

Memory Problems? Use Valgrind

- Valgrind can analyze your code (while running) and detect these memory errors
- <http://cs.ecs.baylor.edu/~donahoo/tools/valgrind/>
 1. Compile your program and produce a binary (ex: myProgram)
 2. Run Valgrind and your program at the same time:

```
unix> valgrind
--tool=memcheck
--leak-check=yes
--show-reachable=yes
--num-callers=20
--track-fds=yes ./myProgram
```