No Web Server Left Behind:

Doing Project 1 in Close to 48 Hours

Athula Balachandran
Wolf Richter
First things First

- Don't sleep, focus on the money

Open Source Web Server Leader NGINX Closes U.S. $3 Million Series A Funding Round.

SAN FRANCISCO and MOSCOW — October 11, 2011

Funding from BV Capital, Runa Capital and MSD Capital Fuels NGINX’s Commercial Plans and New US Presence

Open source web server developer NGINX has received $3 million in a fully subscribed Series A round. Today NGINX powers over 40,000,000 domains on the Internet, and over 20% of the top 1000 busiest websites around the world, including Facebook, Zappos, Groupon, LivingSocial, Hulu, TechCrunch, Dropbox and WordPress.

NGINX’s unique ability to deliver 10 times performance improvements on the existing hardware — combined with lean architecture — scalability and security has propelled the company’s ascent as the fastest and only-growing web server in the world with a market share of 8.5% across all domains. In its most recent survey Netcraft states, “If current trends continue NGINX will soon overtake Microsoft to have the second largest number of active sites.”
CP1: select()able Sockets

- Every fd goes into a select() set
- Every time around the loop
- Single IO per fd returned in sets
  - Only, only, only do things when select() says
- Buffer all IO in and out
- Close out clients on SIGPIPE/write fails
- Close out clients when read fails
andrewid's Connection State

```c
struct connection {
    int socket;
    char buf_in[8192];
    char buf_out[8192];
};
```
CP2: HTTP/1.1

- GET + HEAD + POST
- Read annotated RFC2616
  - A million times, then RFC2616 for anything else
- Pipelining is necessary
- Requests straddling buffers occur
- Connection: Close
- HTTP/1.0 – more for ab nicety
- Errors should become valid error responses
You must handle the case of requests split across buffers.
recv() -> select() -> send() -> accept()
andrewid's Connection State

```c
struct connection {
    int socket;
    char buf_in[8192];
    char buf_out[8192];
    /* HTTP state information */
};
```
CP3: SSL

- Another server socket
- **Check out the SSL example C code**
- Wrap accept()'d fd with `SSL_accept()`
- For IO use `SSL_write()` and `SSL_read()`
- `SSL_write()` and `SSL_read()` may block...
- Use non-blocking sockets underneath
- `fcntl()`, `O_NONBLOCK` etc.
- Read OpenSSL documentation as needed
CP3: SSL Blocking?

These might not come with a single recv() call.
andrewid's Connection State

```c
struct connection {
    int socket;
    char buf_in[8192];
    char buf_out[8192];
    /* HTTP state information */
    bool ssl;
};
```
CP3: CGI

- `<cgi folder> → <cgi web app>`
- All `/cgi/*` URIs handled by single executable
- Parse URI: `<scheme>://<authority><path>?<query>`
- `pipe2()`, `pipe2()`, `fork()`, `dup2()`, `dup2()`...
- Redirect stdin and stdout
- Setup environment variables
  - According to CGI_Requirements doc
- `execve()` executable
CP3: CGI

- More fd's for select()!
- Parent keeps: stdin [r,w], stdout [r,w]
- Child keeps: stdin [r,w], stdout [r,w]
- Child `dup2()`'s these to stdin and stdout
- **Follow the example C code**
- Transparent IO to executed processes
- Response end when stdout read() == 0
CP3: CGI IO

[9,10] [11,12]

Parent

message body

execve()

process

10 → 9

execve()

process

12 → 11

Parent

Response
CP3: CGI Request Anatomy

HTTP Headers

\r\n
Message Body

Environment Variables

stdin
结构体 Connection State

```c
struct connection {
    int socket;
    char buf_in[8192];
    char buf_out[8192];
    /* HTTP state information */
    bool ssl;
    int stdin;
    int stdout;
};
```
Grading?

- CP1: Turn In (5) + 5 from Robust IO
- CP2: Turn In (5) + 5 from HTTP 1.1
- CP3: Final Grade...
- CP1 + CP2 + Read the Rubric...
- CP1 + CP2 + Robust IO (30) +
- HTTP 1.1 (20) + HTTPS via TLS (15) +
- CGI (15) + Robustness and Testing (5) +
- Style (5) == 100
Grading? CP2?

- I personally apologize for the delay...
- So CP2 and CP3 will be coming this weekend together
- Why: HotMobile deadlines Friday
- Wolf – 2
- Athula – 1
- If you need to know about CP2 for CP3, contact me directly
Project 1...

- New for the course
- Parts were ambiguous
  - Sorry!
- But, now you're 15-441 history
- Hopefully you had fun!
So ask away.
GitHub:

Git it, got it, good.

git clone git://github.com/theonewolf/15-441-Recitation-Sessions.git