

CS311 - FA13: Perfect Numbers

Trevor Bramwell

December 8, 2013

1 Design

This project is designed around three interdependent applications: compute, report, and manage. All use internet sockets to communicate, with the end goal of finding perfect numbers. Distributed.

Manage manages compute processes, and hands out ranges of numbers relative to the amount of work each compute process is capable of. Report requests data from manage, and also is used to stop manage, which in turn stops all compute clients.

For the managing of ranges needing to be computed, and distribution, I borrow heavily from The Linux Programming Interface, section 59.11 (Stream Sockets). This section shows how to handling passing ranges from a server to clients, and update a local reference to the next range to compute.

Since manage is also in python, I will be borrowing from the Python Module of the Week website's section on select.

Report will be a simple python script that connects via sockets, and sends XML message for specific pieces of data. The '-k' will send an XML message that tells manage to kill itself.

1.1 XML Protocol

Because this assignment requires that we parse XML by hand, I have created a 'perfect' protocol for communication between the compute processes and manage.

Compute

After each compute process establishes a connection with manage, they send their performance data. In response, manage sends an initial point for compute to start at. This request and reponse is handled by the following XML protocol:

NOTE: Each request and response begins with the string:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Request:

```
<perf>235234</perf>
```

Reponse:

```
<range>1</range>
```

A subsequent request to manage by any client would return 235235, as that will be the next available range.

When a client finds perfect numbers it will send them to the server, and the server will not send a response.

Request:

```
<perfect>
  <number>1</number>
  ...
  <number>8286</number>
</perfect>
```

Report

The other protocol involved is between manage and report. There are only two requests report sends. One for the numbers, and one for shutting down the server.

Request:

```
<report />
```

Response:

```
<perfect>
  <number>1</number>
  <number>6</number>
  ...
  <number>N</number>
</perfect>
<clients>
  <client>
    <ip>352.643.234.8</ip>
    <port>723423</port>
    <perf>235234</perf>
  </client>
  ...
</clients>
```

Request:

```
<quit />
```

2 Deviation

Since this protocol isn't actually XML compliant, I forwent the strict adherence to the xml header, and just wrapped each request and response in:

```
<xml>
...
</xml>
```

3 Questions

What do you think the main point of this assignment is?

The main point of this assignment was to get us to learn how to use `AF_INET` sockets, `select`, and message queues.

How did you ensure your solution was correct? Testing details, for instance. This section should be very thorough.

My testing was done primarily through trial and error. Almost all of it was done on my local machine, using `AF_INET` sockets. This allowed me to iterate quickly on issues and resolve bugs when they cropped up.

What did you learn?

I learned about using sockets, perfect numbers, calculating IOPS, and regular expression groups.

Calls to `send` and `recv`, are not guaranteed to send/recv the full amount requested. These calls can also exit immediately if the socket is set to non-blocking. This caused major issues for me when trying to send 'kill' signals to all the compute processes.

Also, it is very challenging to debug asynchronous network code. This is evident from using `select`. Because a client or server can die at any time, you have to think about the order of actions when reading and writing to sockets.

4 Work Log

commit 1f51d1e95a87c47922c65197b2a458a34df063b0
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sun Dec 8 23:38:41 2013 -0800

CS311: Final draft of writeup.

commit 392bc5768b045814eee14e1191e4f2ff64ca9643
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sun Dec 8 23:36:31 2013 -0800

CS311: Switch cc to icc

commit c3214f098905b6050c4092d52c43cd8accd91de9
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sun Dec 8 23:35:56 2013 -0800

CS311: Add main loop to compute for comm with manage

commit 4ff5ea4b6582c7e64dca723b91336f896fc92a3d
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sun Dec 8 23:14:04 2013 -0800

CS311: Manage server updated to respond with XML.

Currently only handles requests for a range, and reponses of new perfect numbers.

There is a bug in getting the perfect numbers, because regular expression groups are overwritten when matched multiple times.

commit c01efe477bbc5fbbb09058ca7e4189fe116c04b4
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sun Dec 8 17:36:54 2013 -0800

CS311: Killing client crashes server.

conns did not have 'remove' method.

commit b0c52dcf1f5fdad04ec588c6d25825a528b487df
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sun Dec 8 17:35:56 2013 -0800

CS311: Add getopt to compute.

Have seperate main program/code to manage getopt.

commit f4bdd9449f08bee9b8b3b5c3756a7125ce860d18
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sun Dec 8 14:06:21 2013 -0800

CS311: Add initial XML expected response to report

commit 21b2c5ae24c6cfab065853764d241bd29855b856
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sat Dec 7 23:29:20 2013 -0800

CS311: Add design draft to write-up

commit baf204978d541d3a3d50e3e14fba9b9fcdacc42
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sat Dec 7 17:55:25 2013 -0800

CS311: Try to fix bug in signal handling.

commit 6f2db99befcbea01818476c6e585e33617ade047
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sat Dec 7 00:42:34 2013 -0800

CS311: Add basic code for report.

commit 3cfabae199dae8911ed0ce3b8cb37618b974136
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Sat Dec 7 00:22:27 2013 -0800

CS311: Add logging, move parsing to own function.

Bug in Manage.signal. Not all clients recieveing 'die' command.

commit d7aa46c89bb575b49bc35df89859ceed8a9c6773
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Fri Dec 6 19:49:45 2013 -0800

CS311: Add interrupt handler to manage.py.

commit 52df053a9fd07cb39e9bf74b49c74677daf21a04
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Fri Dec 6 00:36:29 2013 -0800

CS311: Manage is now a simple socket select server

commit c921d05a8c8078c48d6269ac2a6c81d5af50eb2c
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Thu Dec 5 21:46:25 2013 -0800

CS311: Add manage.py initial setup.

Currently doles out sequences of numbers.

commit e1805ab1525a7f11c344521e50e28c9071102af1
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Thu Dec 5 00:32:21 2013 -0800

CS311: Remove compute binary

commit ef523db14a04301c040bbb7141f15405293295ba
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Thu Dec 5 00:25:12 2013 -0800

CS311: Convert performance checking to thread.

Add shared header file for errExit.
Add Socket init code for client.

commit d24d91bf955f4e8874990f45befc4f5f86ba301b
Author: Trevor Bramwell <trevor@bramwell.net>
Date: Wed Dec 4 21:53:47 2013 -0800

CS311: Add profile function to compute.

Determines the number of mods a machine can do in 15 seconds.

```
commit aa40b114c5273db8934b129d16404cccbf4723dd
Author: Trevor Bramwell <trevor@bramwell.net>
Date:   Wed Dec 4 11:28:09 2013 -0800
```

CS311: Add client, server, and server_select examples

```
commit 19789e21c190028a0200d63d5b8ff8b3bc04578d
Author: Trevor Bramwell <trevor@bramwell.net>
Date:   Wed Dec 4 11:27:10 2013 -0800
```

CS311: Add instructions

```
commit eb095c2f272c81e3f4cf05f84c210350f2dcbea4
Author: Trevor Bramwell <trevor@bramwell.net>
Date:   Wed Dec 4 08:12:48 2013 -0800
```

CS311: Add manage and report skeletons.

```
commit 74e64e2092853dadf7b03230c296ad379c001fc3
Author: Trevor Bramwell <trevor@bramwell.net>
Date:   Mon Dec 2 12:53:12 2013 -0800
```

CS311: Initial Commit of Assignment 5.