Week 3 Calculation Examples

1. For the following question, proper hexadecimal format is (0xYYYY) where Y will range in (0-9) or (A-F). Only proper formats will be accepted.
   Suppose that we send a DNS request with ID #12162.
   a. What is the little-endian representation (hexadecimal)? □
   b. What is the big-endian representation (hexadecimal)? □
   c. Which representation is required for network communication? (Enter "1" or "2" without quotes) □

Solution:

- Find big-endian hexadecimal representation first.

  ○ Since we are converting this decimal number to a value with base 16, we should first consider the powers of 16.
    ▪ $16^0 = 1$
    ▪ $16^1 = 16$
    ▪ $16^2 = 256$
    ▪ $16^3 = 4{,}096$
    ▪ $16^4 = 65{,}536$

  ○ We can see that our decimal number 12,162 is larger than the power of 3 but smaller than the power of 4.

    So first calculate how many times 12,162 can be divided by 16^3.

    16^3 = 4,096

$$\frac{12{,}162}{4{,}096} = 2.96923828125$$

    Round down to 2. This number can be divided by 16^3 twice, but there is a remainder, so we calculate the value of the remainder.

$$12{,}162 - (4{,}096 * 2) = 12{,}161 - 8{,}192 = 3{,}970$$

    Next, we calculate how many times the remainder 3,970 can be divided by 16^2.

    16^2 = 256

$$\frac{3{,}970}{256} = 15.5078125$$

    Round down to 15. This number can be divided by 16^2 fifteen times, but there is a remainder, so we calculate the value of the remainder.

$$3{,}970 - (256 * 15) = 3{,}970 - 3{,}840 = \ 130$$

Next, we calculate how many times the remainder 130 can be divided by 16^1.

16^1 = 16

$$\frac{130}{16} = 8.125$$

Round down to 8. This number can be divided by 16^1 eight times, but there is a remainder, so we calculate the value of the remainder.

$$130 - (16 * 8) = 130 - 128 = \ 2$$

Next, we calculate how many times the remainder 2 can be divided by 16^0.

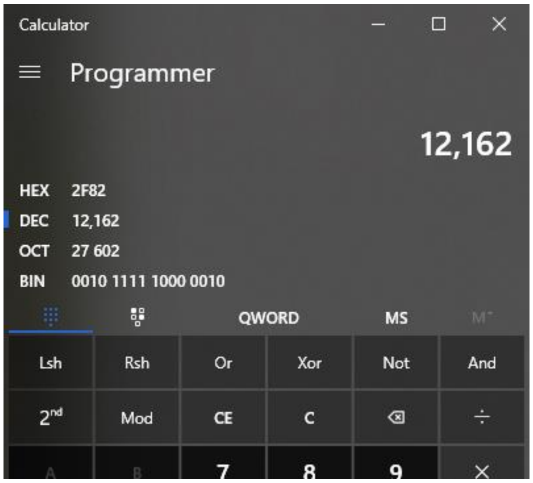
16^0 = 1

$$\frac{2}{1} = 2$$

There is no remainder this time, which is what we would expect since we're dividing by 1.

So, keeping in mind that 15 is equivalent to F in hexadecimal, our big-endian hexadecimal representation is: 2F82

- Note: The calculator on your computer may be capable of computing this. On my Windows PC, I can simply enter the Decimal value into my programming calculator and the big-endian hexadecimal representation is shown.



Calculator

Programmer

12,162

HEX 2F82

DEC 12,162

OCT 27 602

BIN 0010 1111 1000 0010

- Recall from CS271 that the big-endian value can be converted to the little-endian representation by reversing the bytes and so the little-endian value is: 822F

2. A client's browser sends an HTTP request to a website. The website responds with a handshake and sets up a TCP connection. The connection setup takes 2 sec, including the RTT. The browser then sends the request for the website's index file. The index file references 15 additional images, which are to be requested/downloaded by the client's browser. How many requests (including the initial request) must be sent by the browser…

   a) With non-persistent HTTP?
   b) With persistent HTTP?

Solution:

- Persistent means all requests and responses are sent over a single TCP connection.
    - Request 1: Request to set up a TCP connection
    - Request 2: Request to get the website's index file
    - Request 3: Request for first object #1
    - Request 4: Request for object #2

      …

      …
    - Request 16: Request for object #14
    - Request 17: Request for last object #15

$$15 + 2 = 17$$

- Non-persistent means each request/response pair is sent over a separate TCP connection.
    - Request 1: Request to set up a TCP connection
    - Request 2: Request to get the website's index file
    - Request 3: Request to set up a TCP connection for first object #1
    - Request 4: Request for object #1

      …

      …
    - Request 31: Request to set up a TCP connection for last object #15
    - Request 32: Request for object #15

$$(15 * 2) + 2 = 32$$

- How much longer does non-persistent HTTP take than persistent HTTP?
    - Find the difference in the number of requests then multiply this by the amount of time it takes to setup each TCP connection:

      $$(32\ requests - 17\ requests) * 2\ seconds\ per\ request = 30\ seconds$$

3. A client in a network with a proxy server requests a 2 MiB file from an internet server, fakeservername.com. The network's proxy server has a 1.93 Mbps connection to fakeservername.com. The average response time between the network's proxy server and the internet origin server (including RTT) is 1 seconds for a small "header-only" HTTP request/response. The file requested by the client is currently in the proxy server cache, but the proxy server relays the client's request to the internet server with "if-modified since".

   Assume that transmissions between the proxy and the origin servers are stream (not packets) at full bandwidth, with negligible propagation delay.

   How much time is saved if the file has not been modified? (Give answer in seconds, without units, rounded to two decimal places, so for an answer of 1.4233 seconds you would enter "1.42" without the quotes.)

   Solution:

The network proxy server sends an if-modified-since to fakeservername.com. This message returns that the file has not changed. This response is received in a negligible amount of time (from the problem statement). Since the file has not been modified, it can be taken straight from the cache, so the amount of time saved is the amount of time it WOULD have taken to get the file from fakeservername.com to the network proxy… just the data size (L) divided by the connection rate between proxy and fakeservername.com (R)

$$\frac{2\ MiB\ *\left(\frac{1024\ KiB}{1\ MiB}\right)\ *\ \left(\frac{1024\ byte}{1\ KiB}\right)*\left(\frac{8\ bits}{1\ byte}\right)}{1.93\ Mbps*\left(\frac{1{,}000\ Kbps}{1\ Mbps}\right)*\left(\frac{1{,}000\ bps}{1\ Kbps}\right)}\ =\ \frac{16{,}777{,}216\ bits}{1{,}930{,}000\ bps}=8.69\ s$$