Pushkar Joshi
psjoshi@indiana.edu

P538 - Packet Parser

Answers to the questions:

1. Bit width of CPU: The bit width of my CPU is 64 bit. This can be determined by the output of the program data_sizes.c by getting the size of (void*). The size of void pointer is 8(bytes) which means the bit width of CPU is 64 bits.
2. The largest :
   The difference with ranges in signed and unsigned types is because of the leading bit being used for sign.
   a. Short [2 bytes]:
      Signed – [2^15 -1 = 32767]
      Unsigned [2^16 -1 = 65535]
   b. INT [4 bytes]:
      Signed – [2^31 -1 = 2147483647]
      Unsigned [2^32 -1 = 4294967295]
   c. LONG [8 bytes]:
      Signed – [2^64 -1]
      Unsigned [2^64 -1]

3. Casting in C:
   - Type casting is a way to change the representation of data to required type for a single operation.
   - This could be useful in cases where we need to represent data obtained in a form to other as in case of Packet parsing. Or it could be useful in case of arithmetic: (3/5 for instance, would give 0 & not 0.6) Casting result to float would give us the correct answer.
   - When casting data from type to type which have same size or casting data from smaller size type to larger does not cause loss of data. However casting from a larger size type to smaller causes loss of data.
   - One example of this could be converting from float to int. In this case, it would lead to removal of fraction.

4. Hex notations:
   a. Hex notations can be extremely useful when we want to program at a lower level. It would be inconvenient for programmers to write the binary strings when they are working with something that involves bit programming. Hex provides convenience by representing a binary string for byte to 2 digits.

b. Cases: Assembly language, Real time systems. Working with microprocessors and microcontrollers.
c. Two digits can store 1 byte of data. One byte can store 8 bits of data.
d. 1024 = 0x400; 1025 = 0x401

Files:

- data_sizes.c
  - Reviews basics of C:
  - Determine the sizes of data types in C. Get the sizes Built-in data types and user defined type – struct.
  - Reviews type casting in C. Extracting data from a char array in hex format.
  - Casting data from struct to char array in hex format. Exactly opposite of the point above.
- Bit-wise.c
  - Review Bit –wise operations in C.
  - Exercise on bit masking to extract information at a bit level. This is used to check the set flags in TCP header. This is done using '&' operator.
  - Print a bit series: Using bitwise shift operators (>> , <<) find every bit in a character and print out a string 0s & 1s
  - Bit Series for arbitrary data: Same as above, get the bit series for multiple characters instead of just one char.
  - Separate first and second byte of a short. First = byte >> 8; second = byte & 0xFF;
  - Byte swap for a short. Swap first and second byte of a short. (result = arg >> 8 | arg <<8)

- Parse_headers.c
  - Define struct for Ethernet, IP and TCP to get Source and destination addresses at corresponding layers.
    - Src & Dest Mac, Src & Dest IP address, Src & Dest port, and check if TLS is present or not.

Credits:

- http://networksorcery.com/ (For Ethernet & IP headers)
- http://wiki.wireshark.org/Ethernet (& IP as well)
- http://www.cprogramming.com/tutorial/ (Bitwise programming)