ICS 451 Data Networks

HOMEWORK 5 (C or C++ in UNIX) **6 points** **DUE October 12, 2016 noon**

Assume that you have a TCP header in a binary file. Write a program that will do the following:

- Read the binary file into a struct and print the header.
- Create a response header and write it to a new file.
- Read the response binary file and print the header

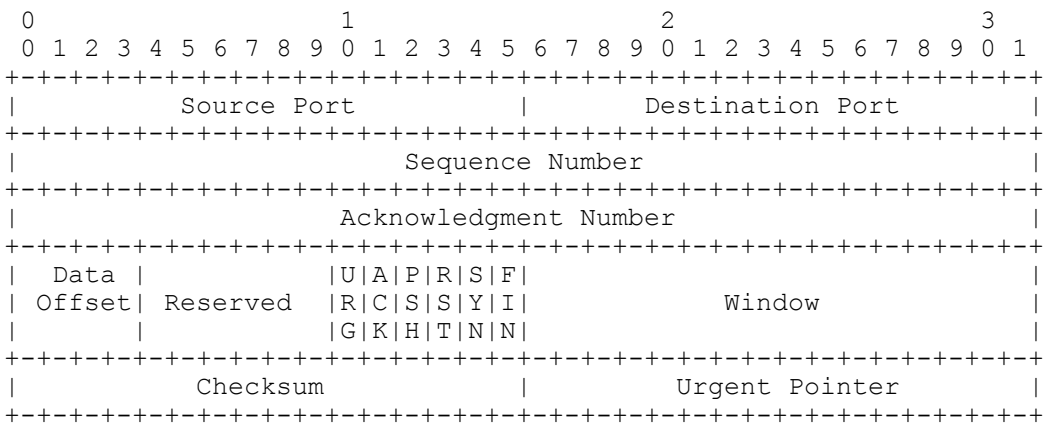
Requirements:

- The response file will switch the source and destination ports
- The acknowledgment number of the request must be sequence number of the response
- If the SYN flag is set in the request, both the SYN and ACK flags must be set in the response
- If the FIN flag is set in the request, only ACK flags must be set in the response
- You must have the following functions:
  - Int printheader();
  - Int readfile(char []);
  - Int writefile(char []);
  - You may use a global structure to share among the various functions.
  - When the header is printed, all values must be printed in decimal except flags and checksum. Checksum must be printed in hex and all of the flags that are set must be printed (e.g. Flags: ACK FIN)
  -

NOTE: you are only writing the TCP header not the entire TCP packet

Submit: Source code, makefile, sample output

TCP Header Format



TCP Header Format

Note that one tick mark represents one bit position.

Source Port: 16 bits

The source port number.

Destination Port: 16 bits

The destination port number.

Sequence Number: 32 bits

Acknowledgment Number: 32 bits

Data Offset: 4 bits

The number of 32 bit words in the TCP Header.

Reserved: 6 bits

Reserved for future use. Must be zero.

Control Bits: 6 bits (from left to right):

```
URG:  Urgent Pointer field significant
ACK:  Acknowledgment field significant
PSH:  Push Function
RST:  Reset the connection
SYN:  Synchronize sequence numbers
FIN:  No more data from sender
```

Window: 16 bits

Must be zero.

Checksum: 16 bits

Must be ffff.

Urgent Pointer: 16 bits

Must be zero.