Homework #11 **M1522.000800 System Programming**

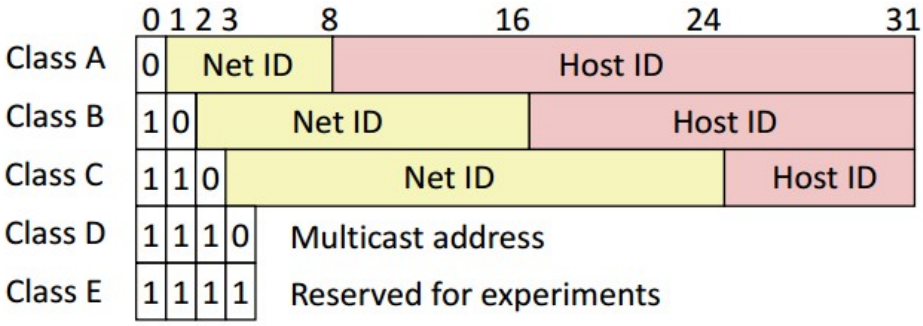
Sample Solution

Question 1

IP v4 Address Structure

The **IPv4** address space is divided into 5 classes.

a) Describe all classes with a bit ordering diagram.



b) Which class do SNU's two networks belong to?

=> Class B

c) How many network nodes can be managed by these classes?

=> 2^16(=65536) – 2 = 65534

d) Each class consists of a `Network ID` and `Host IDs`. Some address are reserved as in each class, e.g., the all-zero-bit address or the all-one-bit address of the `Host ID` cannot be the nodes in a class. The all-zero-bit address and the all-one-bit address of `Network ID` also cannot be used as nodes in a class. Explain why these addresses are reserved and what purpose they serve.

=> the all-zero-bit address of the `Host ID` : Local node
=> the all-ones-bit address of the `Host ID` : Every nodes in the classes (for broadcasting)
=> the all-zero-bit address of the `Network ID` : Local network (current network)
=> the all-ones-bit address of the `Network ID` : Every network

Question 2

Socket Interface

[**Programming Assignment**] Implement the iterative echo client without using the library routines of the csapp's robust I/O library (RIO), i.e., write your code using only Unix I/O functions.

< The Sample Code >

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netdb.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>

#define MAXLINE 100

int main(int argc, char **argv)
{
  int clientfd, port;
  char *host, buf[MAXLINE];
  struct hostent *hp;
  struct sockaddr_in serveraddr;

  if (argc != 3) {
    fprintf(stderr, "usage: %s <host> <port>\n", argv[0]);
    exit(0);
  }
  host = argv[1];
  port = atoi(argv[2]);

  if ((clientfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
    return -1; /* check errno for cause of error */

  /* Fill in the server's IP address and port */
  if ((hp = gethostbyname(host)) == NULL)
    return -2; /* check h_errno for cause of error */
  bzero((char *) &serveraddr, sizeof(serveraddr));
  serveraddr.sin_family = AF_INET;
  bcopy((char *)hp->h_addr_list[0],
      (char *)&serveraddr.sin_addr.s_addr, hp->h_length);
  serveraddr.sin_port = htons(port);

  /* Establish a connection with the server */
  if (connect(clientfd, (const struct sockaddr *) &serveraddr, sizeof(serveraddr)) < 0)
    return -1;

  printf("type:"); fflush(stdout);
  while (fgets(buf, MAXLINE, stdin) != NULL) {
    char *c;
    int ptr = 0;
    write(clientfd, buf, strlen(buf));
    memset(buf, 0, sizeof(char) * MAXLINE);

    do {
      read(clientfd, c, 1);
      buf[ptr++] = *c;
    } while (*c != '\n');

    printf("echo:");
    fputs(buf, stdout);
    printf("type:"); fflush(stdout);
    memset(buf, 0, sizeof(char) * MAXLINE);
  }
  close(clientfd);
  exit(0);
}
```