CSE1004 – Dr. Kanchana Devi V – L52+L53
Lab – 9 – 16/09/2020
Done by ARVIND CB 19BCE1221

Question: Given an IP Address, we have to find the class, its host ID and Network ID and print it. This task has to be done using UDP Protocol.

EXAMPLE:

If I enter 192.168.223.10, the class is C
Host ID is 192.168.223
Network ID is 10

Solution Code:

SERVER

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

#define PORT     8080

char findClass(char str[])
{

    char arr[4];
    int i = 0;
    while (str[i] != '.')
    {
        arr[i] = str[i];
        i++;
    }
    i--;


    int ip = 0, j = 1;
    while (i >= 0)
    {
        ip = ip + (str[i] - '0') * j;
        j = j * 10;
        i--;
    }

    if (ip >=1 && ip <= 126)
        return 'A';

    else if (ip >= 128 && ip <= 191)
        return 'B';

    else if (ip >= 192 && ip <= 223)
        return 'C';

    else if (ip >= 224 && ip <= 239)
        return 'D';

    else
        return 'E';
}

void separate(char str[], char ipClass)
{

    char network[12], host[12];
    for (int k = 0; k < 12; k++)
        network[k] = host[k] = '\0';

    if (ipClass == 'A')
    {
        int i = 0, j = 0;
        while (str[j] != '.')
            network[i++] = str[j++];
        i = 0;
        j++;
        while (str[j] != '\0')
            host[i++] = str[j++];
        printf("Network ID is %s\n", network);
        printf("Host ID is %s\n", host);
    }

    else if (ipClass == 'B')
    {
        int i = 0, j = 0, dotCount = 0;

        while (dotCount < 2)
        {
            network[i++] = str[j++];
            if (str[j] == '.')
                dotCount++;
        }
        i = 0;
        j++;

        while (str[j] != '\0')
            host[i++] = str[j++];

        printf("Network ID is %s\n", network);
        printf("Host ID is %s\n", host);
    }
```

```
    else if (ipClass == 'C')
    {
        int i = 0, j = 0, dotCount = 0;
        while (dotCount < 3)
        {
            network[i++] = str[j++];
            if (str[j] == '.')
                dotCount++;
        }

        i = 0;
        j++;

        while (str[j] != '\0')
            host[i++] = str[j++];

        printf("Network ID is %s\n", network);
        printf("Host ID is %s\n", host);
    }

    else
        printf("In this Class, IP address is not"
           " divided into Network and Host ID\n");
}

int main() {
    int sockfd;
    char buffer[256];
    struct sockaddr_in servaddr, cliaddr;
    if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }
```

```
   memset(&servaddr, 0, sizeof(servaddr));
   memset(&cliaddr, 0, sizeof(cliaddr));

   servaddr.sin_family    = AF_INET;
   servaddr.sin_addr.s_addr = INADDR_ANY;
   servaddr.sin_port = htons(PORT);

   if (bind(sockfd, (const struct sockaddr *)&servaddr,
sizeof(servaddr)) < 0 )
   {
      perror("bind failed");
      exit(EXIT_FAILURE);
   }

   int len, n;
   len = sizeof(cliaddr);

   n = recvfrom(sockfd, (char *)buffer, 256,
            MSG_WAITALL, ( struct sockaddr *) &cliaddr,
            &len);
   buffer[n-1] = '\0';
   printf("Client : %s\n", buffer);
   sendto(sockfd, buffer, sizeof(buffer),
      MSG_CONFIRM, (const struct sockaddr *) &cliaddr,
         len);

   char cls = findClass(buffer);
   printf("Class of IP Address is %c \n", cls);
   separate(buffer, cls);

   return 0;
}
```

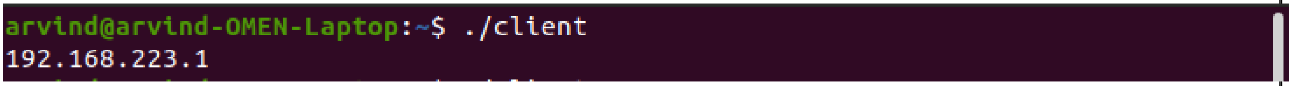
CLIENT:

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

#define PORT     8080
#define MAXLINE 1024
int main() {
   int sockfd;
   char buff[256];
   struct sockaddr_in     servaddr;


   if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
      perror("socket creation failed");
      exit(EXIT_FAILURE);
   }

   memset(&servaddr, 0, sizeof(servaddr));

   servaddr.sin_family = AF_INET;
   servaddr.sin_port = htons(PORT);
   servaddr.sin_addr.s_addr = INADDR_ANY;

   int n, len;
   fgets(buff, 256, stdin);
   sendto(sockfd, buff, sizeof(buff),
      MSG_CONFIRM, (const struct sockaddr *) &servaddr,
         sizeof(servaddr));
```

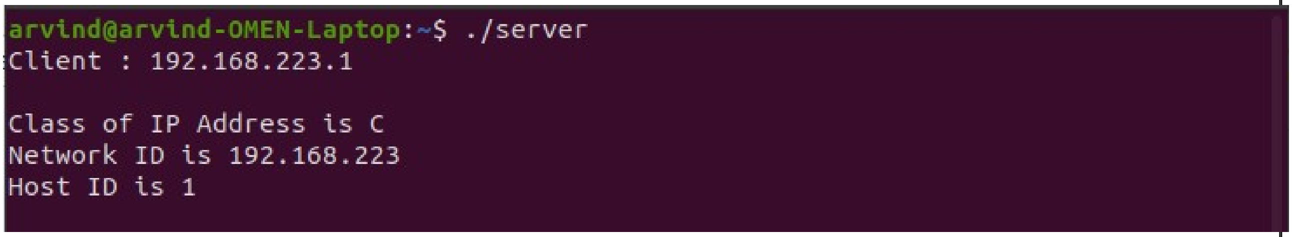
```
    n = recvfrom(sockfd, (char *)buff, MAXLINE,
                MSG_WAITALL, (struct sockaddr *) &servaddr,
                &len);
    buff[n] = '\0';
    close(sockfd);
    return 0;
}
```

OUTPUT SCREENSHOT:

SET 1

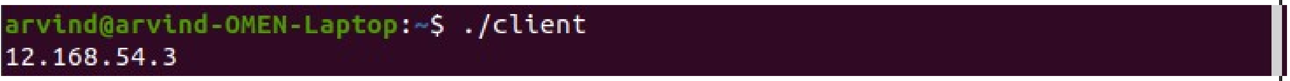


```
arvind@arvind-OMEN-Laptop:~$ ./client
192.168.223.1
```

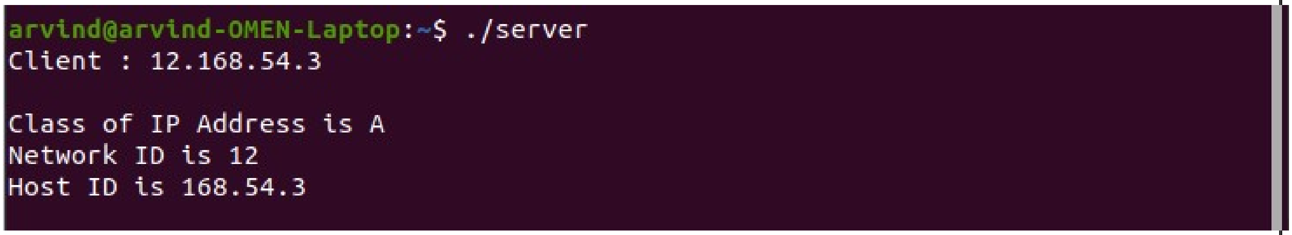


```
arvind@arvind-OMEN-Laptop:~$ ./server
Client : 192.168.223.1

Class of IP Address is C
Network ID is 192.168.223
Host ID is 1
```

SET 2

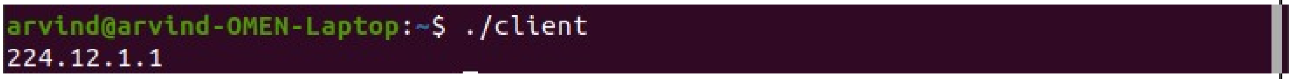


```
arvind@arvind-OMEN-Laptop:~$ ./client
12.168.54.3
```

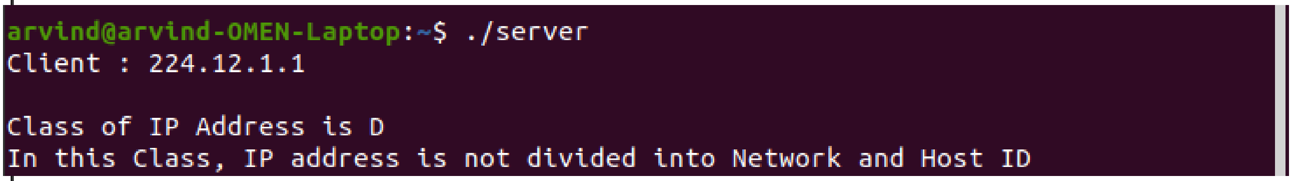


```
arvind@arvind-OMEN-Laptop:~$ ./server
Client : 12.168.54.3

Class of IP Address is A
Network ID is 12
Host ID is 168.54.3
```

SET 3



```
arvind@arvind-OMEN-Laptop:~$ ./client
224.12.1.1
```



```
arvind@arvind-OMEN-Laptop:~$ ./server
Client : 224.12.1.1

Class of IP Address is D
In this Class, IP address is not divided into Network and Host ID
```