arpclient.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#define SA struct sockaddr

int main(int argc, char* argv[])
{
      int sockfd, len;
      char ip[30], mac[30];
      struct sockaddr_in servaddr;
      int choice;

      sockfd = socket(AF_INET, SOCK_DGRAM, 0);

      servaddr.sin_family = AF_INET;
      servaddr.sin_port = htons(atoi(argv[1]));
      servaddr.sin_addr.s_addr = inet_addr("172.17.0.34");

      printf("\nARP AND RARP SIMULATION\n");
      printf("Press 1 for ARP, 2 for RARP, 0 to quit\n");
      do
      {
            printf("\nEnter choice: ");
            scanf("%d",&choice);
            int l = snprintf(NULL, 0, "%d", choice);
            char c[l+1];
```

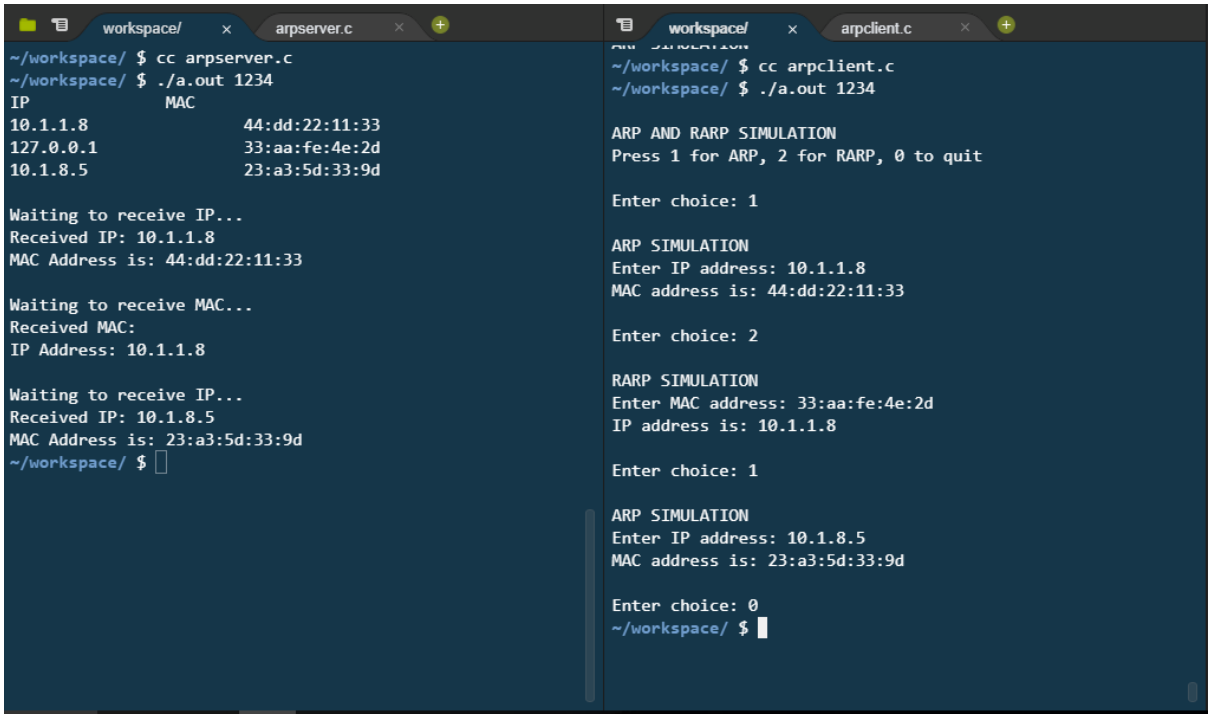
```
            snprintf(c, l+1, "%d", choice);
            sendto(sockfd, c, sizeof(c), 0, (SA*)&servaddr,
sizeof(servaddr));
            if(choice==1)
            {
                  printf("\nARP SIMULATION\n");
                  printf("Enter IP address: ");
                  scanf("%s", ip);
                  sendto(sockfd, ip, sizeof(ip), 0, (SA*)&servaddr,
sizeof(servaddr));
                  len = sizeof(servaddr);
                  recvfrom(sockfd, mac, sizeof(mac), 0,
(SA*)&servaddr, &len);
                  printf("MAC address is: %s\n", mac);
            }

            else if(choice==2)
            {
                  printf("\nRARP SIMULATION\n");
                  printf("Enter MAC address: ");
                  scanf("%s", ip);
                  sendto(sockfd, mac, sizeof(mac), 0, (SA*)&servaddr,
len);
                  recvfrom(sockfd, ip, sizeof(ip), 0, (SA*)&servaddr,
&len);
                  printf("IP address is: %s\n", ip);
            }
            bzero(mac, sizeof(mac));
      }while(choice);

      return 0;
}
```

arpserver.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#define SA struct sockaddr

struct IPmac
{
      char ip[100];
      char mac[100];
};

int main(int argc, char* argv[])
{
      int sockfd, len, i;
      struct sockaddr_in servaddr;
      char buff[30], temp[30], ip[30], mac[30];
      int choice;
      char c[2];

      struct IPmac in[3] =
      {
            {"10.1.1.8","44:dd:22:11:33"},
            {"127.0.0.1","33:aa:fe:4e:2d"},
            {"10.1.8.5","23:a3:5d:33:9d"}
      };
```

```
	printf("IP\t\tMAC\n");
	for(i = 0; i < 3; i++)
		printf("%s\t\t%s\n", in[i].ip, in[i].mac);

	sockfd = socket(AF_INET, SOCK_DGRAM, 0);

	servaddr.sin_family = AF_INET;
	servaddr.sin_port = htons(atoi(argv[1]));
	servaddr.sin_addr.s_addr = INADDR_ANY;

	bind(sockfd, (SA*)&servaddr, sizeof(servaddr));
	do
	{
	len = sizeof(servaddr);
	recvfrom(sockfd, c, sizeof(c), 0, (SA*)&servaddr, &len);
	choice = atoi(c);
		if(choice==1)
		{
			printf("\nWaiting to receive IP...\n");
			recvfrom(sockfd, ip, sizeof(ip), 0, (SA*)&servaddr,
&len);
			printf("Received IP: %s\n", ip);
			for(i = 0; i < strlen(ip); i++)
				temp[i] = ip[i];
			temp[i] = '\0';
			for(i = 0; i < 3; i++)
				if(strcmp(temp, in[i].ip)==0)
				{
					strcpy(mac, in[i].mac);
					break;
				}
			printf("MAC Address is: %s\n", mac);
```

```
                sendto(sockfd, mac, sizeof(mac), 0, (SA*)&servaddr,
len);
                bzero(mac, sizeof(mac));
          }

          else if(choice==2)
          {
                printf("\nWaiting to receive MAC...\n");
                recvfrom(sockfd, mac, sizeof(mac), 0,
(SA*)&servaddr, &len);
                printf("Received MAC: %s\n", mac);
                for(i = 0; i < strlen(mac); i++)
                      temp[i] = mac[i];
                temp[i] = '\0';
                for(i = 0; i < 3; i++)
                      if(strcmp(temp, in[i].mac)==0)
                      {
                            strcpy(ip, in[i].ip);
                            break;
                      }
                printf("IP Address: %s\n", ip);
                sendto(sockfd, ip, sizeof(ip), 0, (SA*)&servaddr,
len);
                bzero(mac, sizeof(mac));
          }
     }while(choice);
     return 0;

}
```


workspace/
arpserver.c
~/workspace/ $ cc arpserver.c
~/workspace/ $ ./a.out 1234
IP MAC
10.1.1.8 44:dd:22:11:33
127.0.0.1 33:aa:fe:4e:2d
10.1.8.5 23:a3:5d:33:9d
Waiting to receive IP...
Received IP: 10.1.1.8
MAC Address is: 44:dd:22:11:33
Waiting to receive MAC...
Received MAC:
IP Address: 10.1.1.8
Waiting to receive IP...
Received IP: 10.1.8.5
MAC Address is: 23:a3:5d:33:9d
~/workspace/ $
workspace/
arpclient.c
~/workspace/ $ cc arpclient.c
~/workspace/ $ ./a.out 1234
ARP AND RARP SIMULATION
Press 1 for ARP, 2 for RARP, 0 to quit
Enter choice: 1
ARP SIMULATION
Enter IP address: 10.1.1.8
MAC address is: 44:dd:22:11:33
Enter choice: 2
RARP SIMULATION
Enter MAC address: 33:aa:fe:4e:2d
IP address is: 10.1.1.8
Enter choice: 1
ARP SIMULATION
Enter IP address: 10.1.8.5
MAC address is: 23:a3:5d:33:9d
Enter choice: 0
~/workspace/ $