Ex No: 9 Pranit Tandon

Date: 21/09/2021 RA1911028010014 (CSE – H2)

**ARP IMPLEMENTATION USING UDP**

**GIVEN REQUIREMENTS:**

There is a single host. The IP address of any Client in the network is given as input and the corresponding hardware address is got as the output.

**TECHNICAL OBJECTIVE**:

Address Resolution Protocol (ARP) is implemented through this program. The IP address of any Client is given as the input. The ARP cache is looked up for the corresponding hardware address. This is returned as the output. Before compiling that Client is pinged.

**METHODOLOGY:**

* Include the necessary header files.
* Create a socket using socket function with family AF\_INET, type as SOCK\_DGRAM.
* Declare structures arpreq ( as NULL structure, if required) and sockaddr\_in.
* Initialize server address to 0 using the bzero function.
* Assign the sin\_family to AF\_INET and sin\_addr using inet\_aton().
* Using the object of arpreq structure assign the name of the Network Device to the data member arp\_dev like, arp\_dev=”eth0”.
* Ping the required Client.
* Using the ioctl() we get the ARP cache entry for the given IP address.
* The output of the ioctl() function is stored in the sa\_data[0] datamember of the arp\_ha structure which is in turn a data member of structure arpreq.
* Print the hardware address of the given IP address on the output console.

**CODING:**

ARP: arp.c

#include<sys/types.h>

#include<sys/socket.h>

#include<net/if\_arp.h>

#include<sys/ioctl.h>

#include<stdio.h>

#include<unistd.h>

#include<netinet/in.h>

#include<arpa/inet.h>

int main(int argc,char \*argv[])

{

struct sockaddr\_in sin={0};

struct arpreq myarp={{0}};

unsigned char \*ptr;

int sd;

sin.sin\_family=AF\_INET;

if(inet\_aton(argv[1],&sin.sin\_addr)==0)

{

printf("Ip address Entered '%s' is not valid \n",argv[1]);

exit(0);

}

memcpy(&myarp.arp\_pa,&sin,sizeof(myarp.arp\_pa));

strcpy(myarp.arp\_dev,"eth0");

sd=socket(AF\_INET,SOCK\_DGRAM,0);

if(ioctl(sd,SIOCGARP,&myarp)==1)

{

printf("No Entry in ARP Cache for '%s'",argv[1]);

exit(0);

}

ptr=&myarp.arp\_ha.sa\_data[0];

printf("MAC Address For '%s' : ",argv[1]);

printf("%X:%X:%X:%X:%X:%X\n",\*ptr,\*(ptr+1),\*(ptr+2),\*(ptr+3),\*(ptr+4),\*(ptr+5),\*(ptr+5));

return 0;

}

**SAMPLE OUTPUT:**

**Host: arp.c**

**(Host Name:Root1)**

[root@localhost 4ita33]# vi arp.c

[root@localhost 4ita33]# ping 172.16.29.51

PING 172.16.29.51 (172.16.29.51) 56(84) bytes of data.  
64 bytes from172.16.29.51: icmp\_seq=1 ttl=64 time=1.19 ms  
64 bytes from 172.16.29.51: icmp\_seq=2 ttl=64 time=0.817 ms  
  
-- 172.16.29.51ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 999ms

rtt min/avg/max/mdev = 0.817/1.005/1.193/0.188 ms

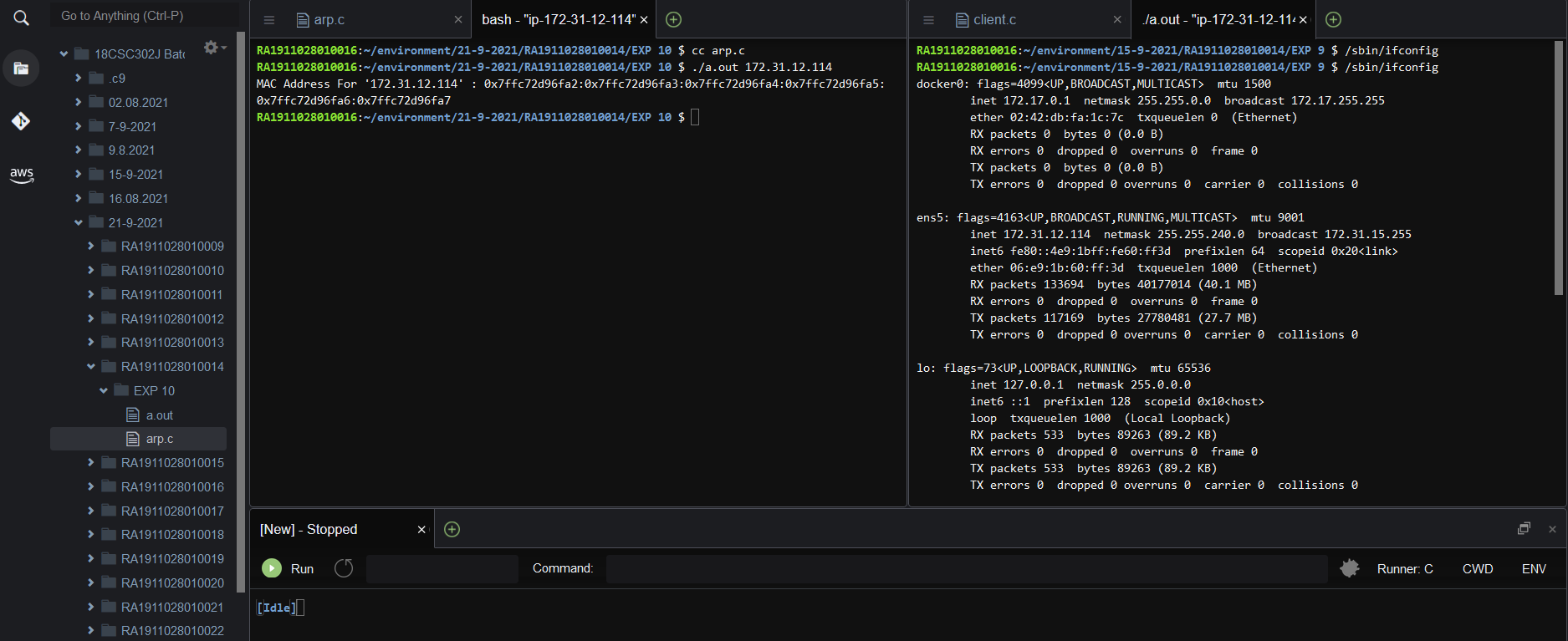
[root@localhost 4ita33]# cc arp.c

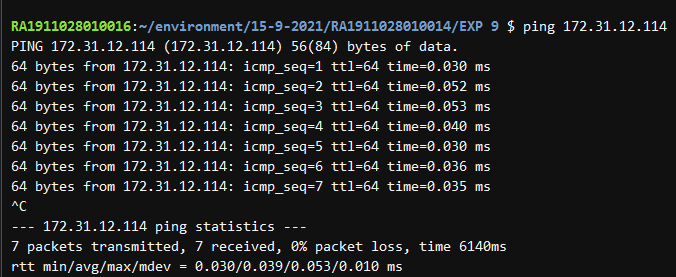
[root@localhost 4ita33]# ./a.out 172.16.29.51

Hardware Address is: 172.16.29.51:

The MAC address is:0:8:5C:5D:47:50:

**OUTPUT**:





**INFERENCE**:

Thus the ARP implementation is developed to gets the MAC address of the remote machine’s IP address from ARP cache and prints it.