Lab Assignment- 1

CSN-361: Computer Networks Laboratory

JULY 25, 2019

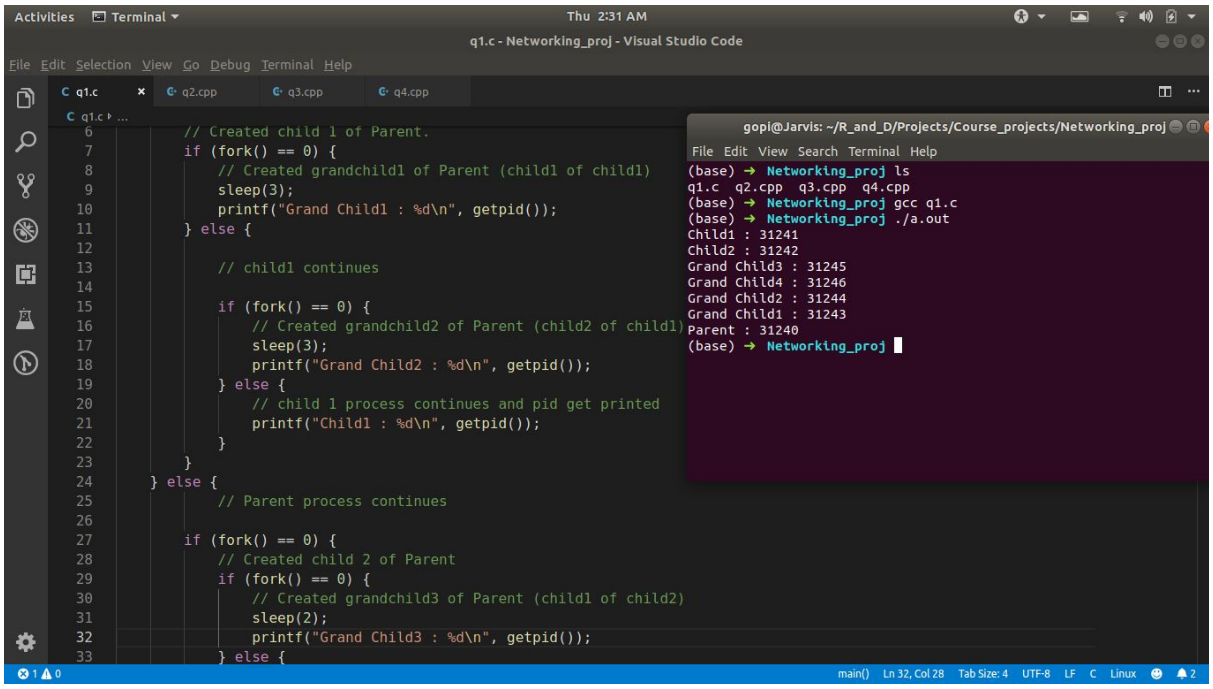
Gopi Kishan
Enrolment No. : 17114035
B.Tech, 3rd Yr
Computer Science and Engineering (CSE)

Problem Statement 1

Write a C program in the UNIX system that creates two children and four grandchildren (two for each child). The program should then print the process-IDs of the two children, four grandchildren and the parent in this order.

- **Algorithms and data structures used in the implementation**
  No specific data structure used

- **Snapshots of running the codes for each of the problems**


Activities
Terminal
Thu 2:31 AM
q1.c - Networking_proj - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
q1.c
q2.cpp
q3.cpp
q4.cpp
gopi@Jarvis: ~/R_and_D/Projects/Course_projects/Networking_proj
File Edit View Search Terminal Help
main() Ln 32, Col 28 Tab Size: 4 UTF-8 LF C Linux

```c
        // Created child 1 of Parent.
        if (fork() == 0) {
            // Created grandchild1 of Parent (child1 of child1)
            sleep(3);
            printf("Grand Child1 : %d\n", getpid());
        } else {

            // child1 continues

            if (fork() == 0) {
                // Created grandchild2 of Parent (child2 of child1)
                sleep(3);
                printf("Grand Child2 : %d\n", getpid());
            } else {
                // child 1 process continues and pid get printed
                printf("Child1 : %d\n", getpid());
            }
        }
    } else {
            // Parent process continues

        if (fork() == 0) {
            // Created child 2 of Parent
            if (fork() == 0) {
                // Created grandchild3 of Parent (child1 of child2)
                sleep(2);
                printf("Grand Child3 : %d\n", getpid());
            } else {
```

```
(base) → Networking_proj ls
q1.c  q2.cpp  q3.cpp  q4.cpp
(base) → Networking_proj gcc q1.c
(base) → Networking_proj ./a.out
Child1 : 31241
Child2 : 31242
Grand Child3 : 31245
Grand Child4 : 31246
Grand Child2 : 31244
Grand Child1 : 31243
Parent : 31240
(base) → Networking_proj
```

Problem Statement 2

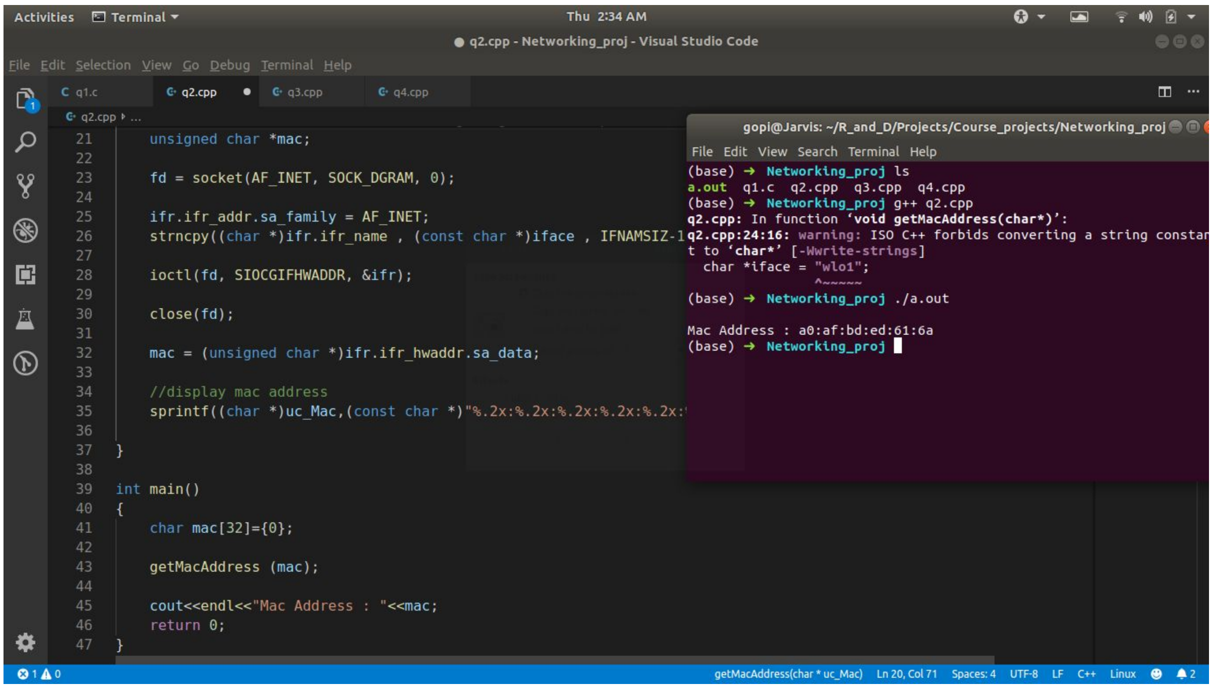
Write a C++ program to print the MAC address of your computer

- **Algorithms and data structures used in the implementation**

  mac : It is a character array used to store Mac Address.

  ifreq : ioctl requests to obtain addresses and requests both to set and retrieve other data and takes the **ifreq** data structure as a parameter this purpose.

- **Snapshots of running the codes for each of the problems**


Activities
Terminal
Thu 2:34 AM
q2.cpp - Networking_proj - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
q1.c
q2.cpp
q3.cpp
q4.cpp
gopi@Jarvis: ~/R_and_D/Projects/Course_projects/Networking_proj
File Edit View Search Terminal Help
getMacAddress(char * uc_Mac) Ln 20, Col 71 Spaces: 4 UTF-8 LF C++ Linux

```cpp
    unsigned char *mac;

    fd = socket(AF_INET, SOCK_DGRAM, 0);

    ifr.ifr_addr.sa_family = AF_INET;
    strncpy((char *)ifr.ifr_name , (const char *)iface , IFNAMSIZ-1

    ioctl(fd, SIOCGIFHWADDR, &ifr);

    close(fd);

    mac = (unsigned char *)ifr.ifr_hwaddr.sa_data;

    //display mac address
    sprintf((char *)uc_Mac,(const char *)"%.2x:%.2x:%.2x:%.2x:%.2x:

}

int main()
{
    char mac[32]={0};

    getMacAddress (mac);

    cout<<endl<<"Mac Address : "<<mac;
    return 0;
}
```

```
(base) ➜ Networking_proj ls
a.out  q1.c  q2.cpp  q3.cpp  q4.cpp
(base) ➜ Networking_proj g++ q2.cpp
q2.cpp: In function ‘void getMacAddress(char*)’:
q2.cpp:24:16: warning: ISO C++ forbids converting a string constan
t to ‘char*’ [-Wwrite-strings]
  char *iface = "wlo1";
                ^~~~~~

(base) ➜ Networking_proj ./a.out

Mac Address : a0:af:bd:ed:61:6a
(base) ➜ Networking_proj 
```

Problem Statement 3

Write your own version of ping program in C language.

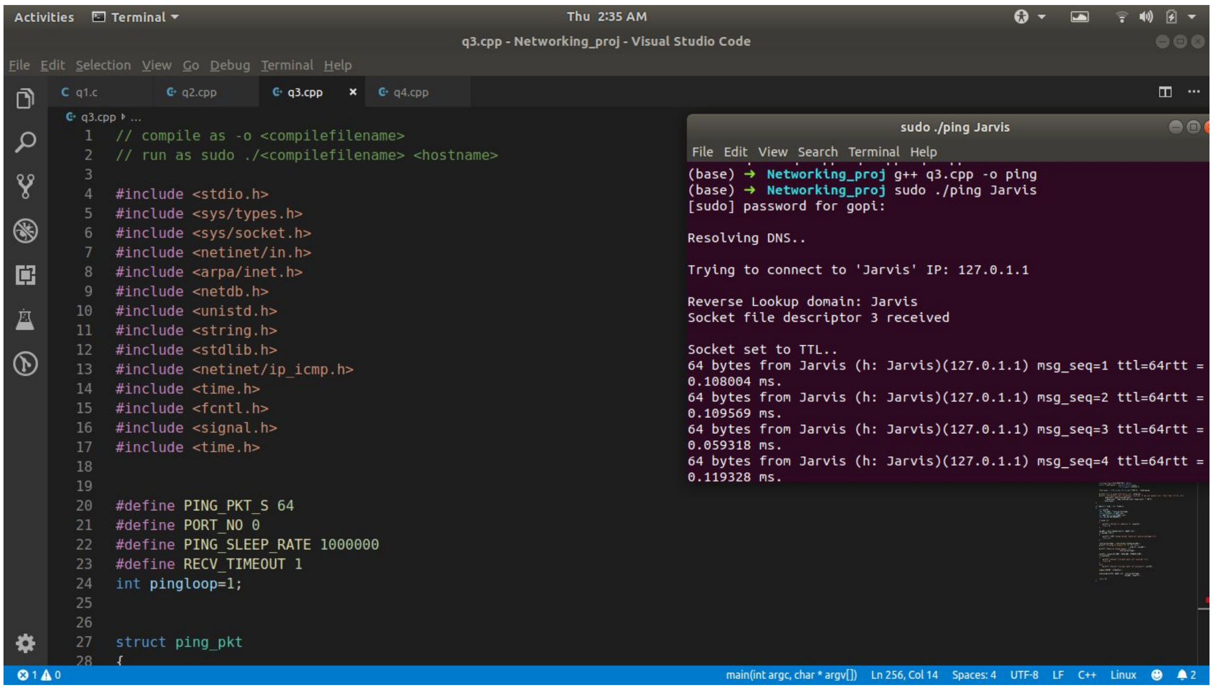
- **Algorithms and data structures used in the implementation**

  The steps followed by a ping program are:

  1. Take a **hostname** as input and do a **DNS lookup** using **gethostbyname()**
  2. Open a **Raw socket** using SOCK_RAW with protocol as IPPROTO_ICMP. Raw socket requires superuser rights so you have to run this code using sudo
  3. Create **icmp packet** and calculate the checksum to be sent.
  4. **Send** the packet.
  5. Wait for it to be **received**

  Data structures used are:
  struct **sockaddr_in** : It is a structure containing an internet address.
  struct **icmphdr** : This is a header (and structure) which is Linux-specific, and will not be present in other operating systems.
  struct **pingpacket** : data packet sent during ping containing request and icmp header
  struct **timeval** : represents time interval passed.
  struct **timespec :** Structure holding an interval broken down into seconds and nanoseconds.

- **Snapshots of running the codes for each of the problems**


Activities
Terminal
Thu 2:35 AM
q3.cpp - Networking_proj - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
q1.c
q2.cpp
q3.cpp
q4.cpp
sudo ./ping Jarvis
File Edit View Search Terminal Help
main(int argc, char * argv[]) Ln 256, Col 14 Spaces: 4 UTF-8 LF C++ Linux

```
// compile as -o <compilefilename>
// run as sudo ./<compilefilename> <hostname>

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>
#include <netinet/ip_icmp.h>
#include <time.h>
#include <fcntl.h>
#include <signal.h>
#include <time.h>


#define PING_PKT_S 64
#define PORT_NO 0
#define PING_SLEEP_RATE 1000000
#define RECV_TIMEOUT 1
int pingloop=1;


struct ping_pkt
{
```

```
(base) ➜ Networking_proj g++ q3.cpp -o ping
(base) ➜ Networking_proj sudo ./ping Jarvis
[sudo] password for gopi:

Resolving DNS..

Trying to connect to 'Jarvis' IP: 127.0.1.1

Reverse Lookup domain: Jarvis
Socket file descriptor 3 received

Socket set to TTL..
64 bytes from Jarvis (h: Jarvis)(127.0.1.1) msg_seq=1 ttl=64rtt =
0.108004 ms.
64 bytes from Jarvis (h: Jarvis)(127.0.1.1) msg_seq=2 ttl=64rtt =
0.109569 ms.
64 bytes from Jarvis (h: Jarvis)(127.0.1.1) msg_seq=3 ttl=64rtt =
0.059318 ms.
64 bytes from Jarvis (h: Jarvis)(127.0.1.1) msg_seq=4 ttl=64rtt =
0.119328 ms.
```

Problem Statement 4

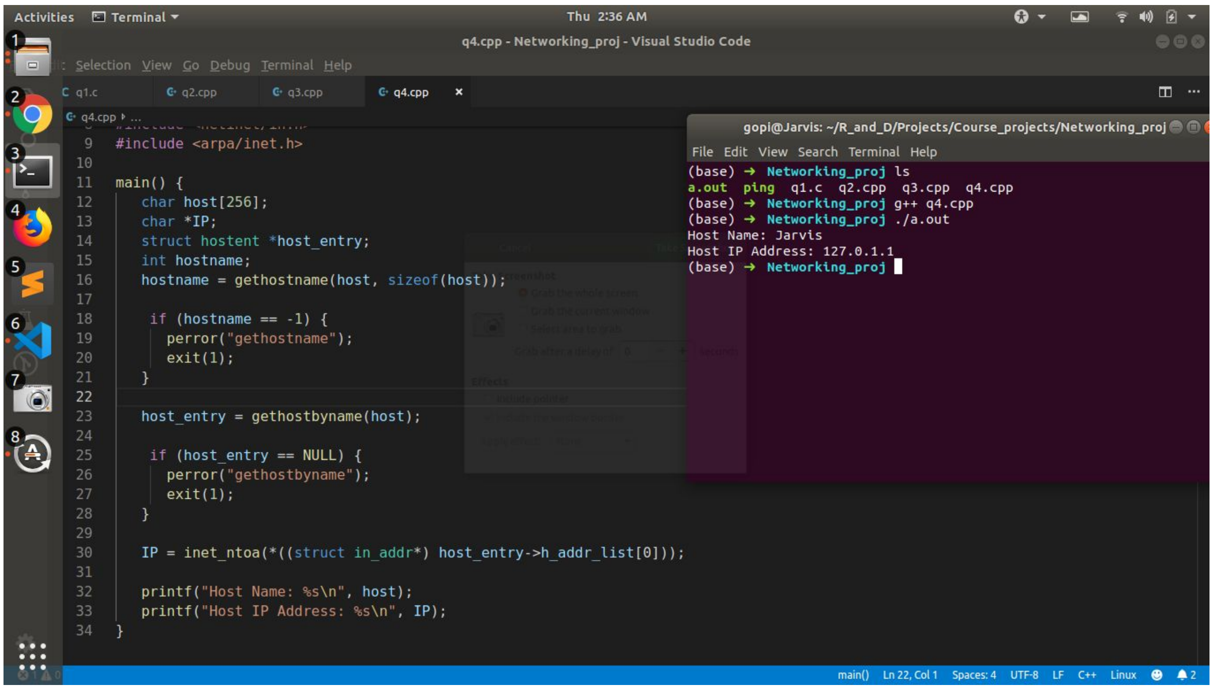
Write a C program to find the host name and the IP address of your computer.

- **Algorithms and data structures used in the implementation**

  hostent: This data structure is used by functions to store information about a given host, such as host name, IPv4 address, and so forth.

  In_addr: This struct data structure stores s_addr field which is internet addresses.

  IP: char array to store IP address.
- **Snapshots of running the codes for each of the problems**



Activities
Terminal
Thu 2:36 AM
q4.cpp - Networking_proj - Visual Studio Code
Selection View Go Debug Terminal Help
q1.c
q2.cpp
q3.cpp
q4.cpp
q4.cpp ▸ ...
gopi@Jarvis: ~/R_and_D/Projects/Course_projects/Networking_proj
File Edit View Search Terminal Help
main() Ln 22, Col 1 Spaces: 4 UTF-8 LF C++ Linux

```
#include <arpa/inet.h>

main() {
    char host[256];
    char *IP;
    struct hostent *host_entry;
    int hostname;
    hostname = gethostname(host, sizeof(host));

     if (hostname == -1) {
       perror("gethostname");
       exit(1);
    }

    host_entry = gethostbyname(host);

     if (host_entry == NULL) {
       perror("gethostbyname");
       exit(1);
    }

    IP = inet_ntoa(*((struct in_addr*) host_entry->h_addr_list[0]));

    printf("Host Name: %s\n", host);
    printf("Host IP Address: %s\n", IP);
}
```

```
(base) → Networking_proj ls
a.out  ping  q1.c  q2.cpp  q3.cpp  q4.cpp
(base) → Networking_proj g++ q4.cpp
(base) → Networking_proj ./a.out
Host Name: Jarvis
Host IP Address: 127.0.1.1
(base) → Networking_proj
```