L-1

CSN-361

August 2019

SUBMITTED TO

Dr. Sudip Roy

CSE, IIT Roorkee

SUBMITTED BY

Natansh Mathur (17114051)

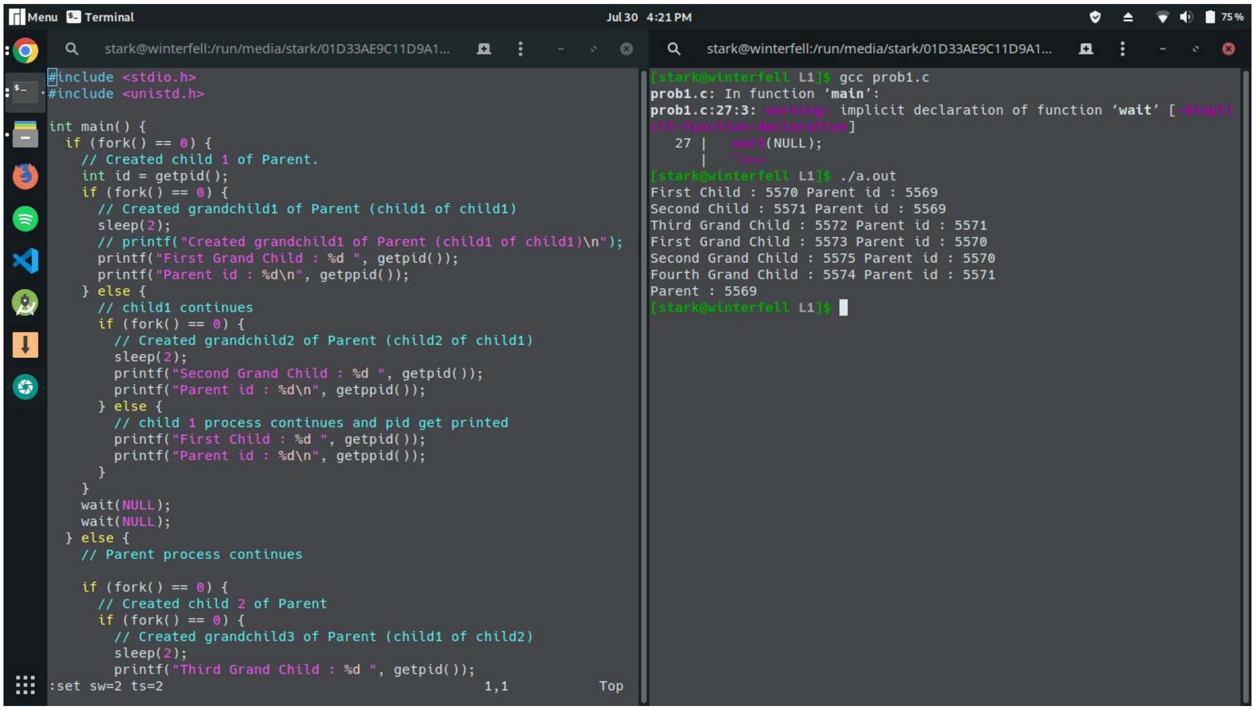
Problem 1

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.

Data Structures and Algorithms used:

None

Screenshots:



Menu
Terminal
Jul 30 4:21 PM
75 %
stark@winterfell:/run/media/stark/01D33AE9C11D9A1...
stark@winterfell:/run/media/stark/01D33AE9C11D9A1...

```c
#include <stdio.h>
#include <unistd.h>

int main() {
  if (fork() == 0) {
    // Created child 1 of Parent.
    int id = getpid();
    if (fork() == 0) {
      // Created grandchild1 of Parent (child1 of child1)
      sleep(2);
      // printf("Created grandchild1 of Parent (child1 of child1)\n");
      printf("First Grand Child : %d ", getpid());
      printf("Parent id : %d\n", getppid());
    } else {
      // child1 continues
      if (fork() == 0) {
        // Created grandchild2 of Parent (child2 of child1)
        sleep(2);
        printf("Second Grand Child : %d ", getpid());
        printf("Parent id : %d\n", getppid());
      } else {
        // child 1 process continues and pid get printed
        printf("First Child : %d ", getpid());
        printf("Parent id : %d\n", getppid());
      }
    }
    wait(NULL);
    wait(NULL);
  } 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("Third Grand Child : %d ", getpid());
:set sw=2 ts=2                                         1,1           Top
```

```
[stark@winterfell L1]$ gcc prob1.c
prob1.c: In function ‘main’:
prob1.c:27:3: warning: implicit declaration of function ‘wait’ [-Wimplicit-function-declaration]
   27 |   wait(NULL);
      |   ^~~~
[stark@winterfell L1]$ ./a.out
First Child : 5570 Parent id : 5569
Second Child : 5571 Parent id : 5569
Third Grand Child : 5572 Parent id : 5571
First Grand Child : 5573 Parent id : 5570
Second Grand Child : 5575 Parent id : 5570
Fourth Grand Child : 5574 Parent id : 5571
Parent : 5569
[stark@winterfell L1]$ 
```

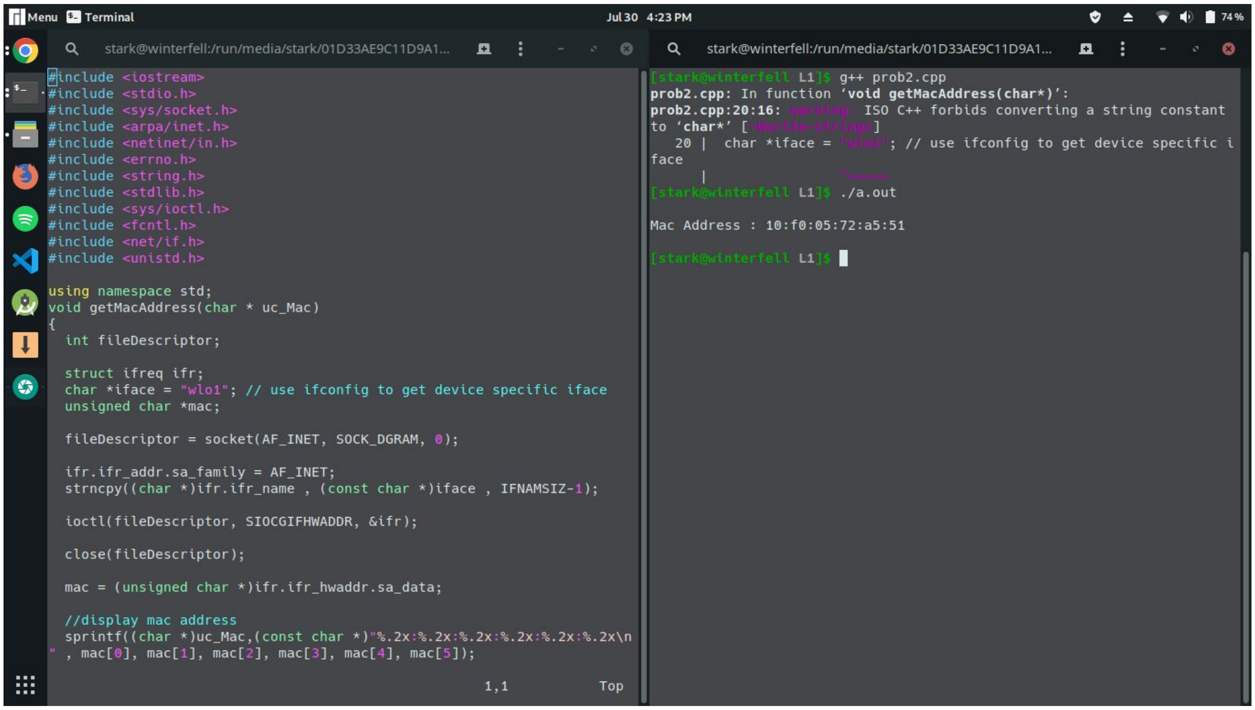
Problem 2

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

Data Structures and Algorithms used:

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.

Screenshots:


Menu
Terminal
Jul 30 4:23 PM
74 %
stark@winterfell:/run/media/stark/01D33AE9C11D9A1...
stark@winterfell:/run/media/stark/01D33AE9C11D9A1...

```
#include <iostream>
#include <stdio.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <net/if.h>
#include <unistd.h>

using namespace std;
void getMacAddress(char * uc_Mac)
{
  int fileDescriptor;

  struct ifreq ifr;
  char *iface = "wlo1"; // use ifconfig to get device specific iface
  unsigned char *mac;

  fileDescriptor = socket(AF_INET, SOCK_DGRAM, 0);

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

  ioctl(fileDescriptor, SIOCGIFHWADDR, &ifr);

  close(fileDescriptor);

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

  //display mac address
  sprintf((char *)uc_Mac,(const char *)"%.2x:%.2x:%.2x:%.2x:%.2x:%.2x\n
" , mac[0], mac[1], mac[2], mac[3], mac[4], mac[5]);

                                                    1,1           Top
```

```
[stark@winterfell L1]$ g++ prob2.cpp
prob2.cpp: In function ‘void getMacAddress(char*)’:
prob2.cpp:20:16: warning: ISO C++ forbids converting a string constant
to ‘char*’ [-Wwrite-strings]
   20 |   char *iface = "wlo1"; // use ifconfig to get device specific i
face
      |                 ^~~~~~
[stark@winterfell L1]$ ./a.out

Mac Address : 10:f0:05:72:a5:51

[stark@winterfell L1]$ 
```

Problem 3

Write your own version of the ping program in C language.

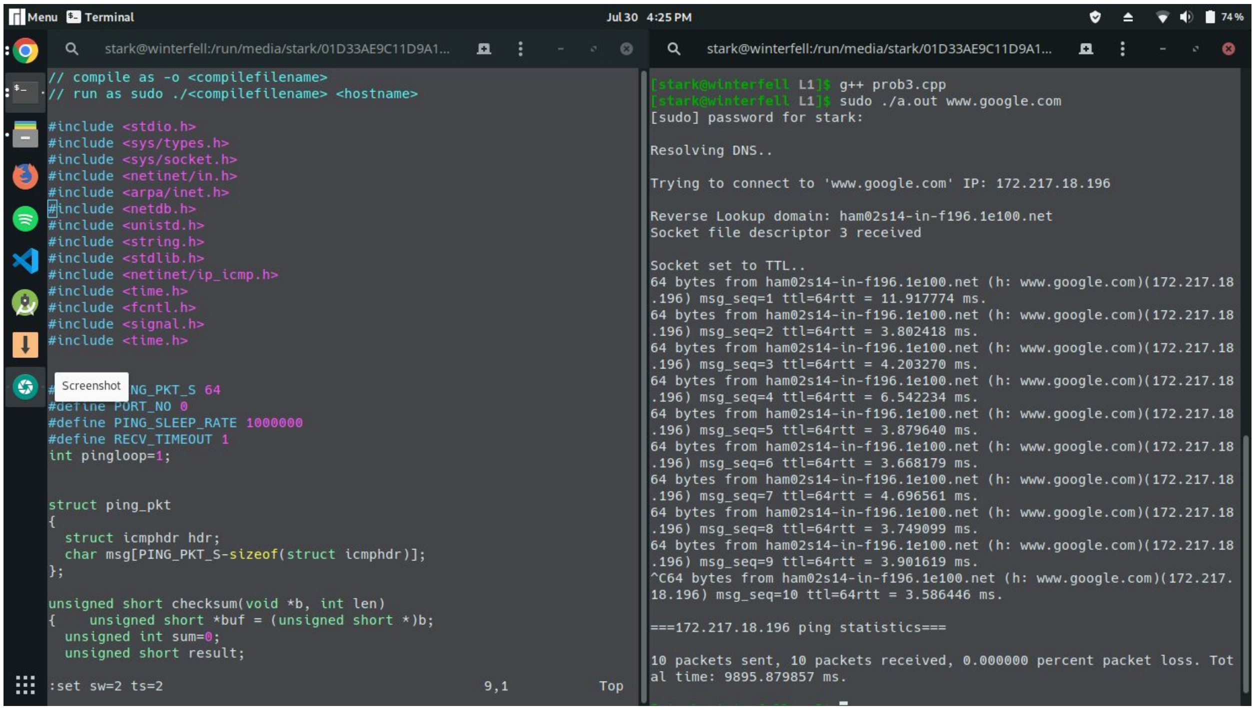
Data Structures and Algorithms used:

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 the 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.

Screenshots:



Menu
Terminal
Jul 30 4:25 PM
74 %
stark@winterfell:/run/media/stark/01D33AE9C11D9A1...
stark@winterfell:/run/media/stark/01D33AE9C11D9A1...
Screenshot

```
// 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>


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


struct ping_pkt
{
  struct icmphdr hdr;
  char msg[PING_PKT_S-sizeof(struct icmphdr)];
};

unsigned short checksum(void *b, int len)
{    unsigned short *buf = (unsigned short *)b;
  unsigned int sum=0;
  unsigned short result;

:set sw=2 ts=2                                    9,1           Top
```

```
[stark@winterfell L1]$ g++ prob3.cpp
[stark@winterfell L1]$ sudo ./a.out www.google.com
[sudo] password for stark:

Resolving DNS..

Trying to connect to 'www.google.com' IP: 172.217.18.196

Reverse Lookup domain: ham02s14-in-f196.1e100.net
Socket file descriptor 3 received

Socket set to TTL..
64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.18
.196) msg_seq=1 ttl=64rtt = 11.917774 ms.
64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.18
.196) msg_seq=2 ttl=64rtt = 3.802418 ms.
64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.18
.196) msg_seq=3 ttl=64rtt = 4.203270 ms.
64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.18
.196) msg_seq=4 ttl=64rtt = 6.542234 ms.
64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.18
.196) msg_seq=5 ttl=64rtt = 3.879640 ms.
64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.18
.196) msg_seq=6 ttl=64rtt = 3.668179 ms.
64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.18
.196) msg_seq=7 ttl=64rtt = 4.696561 ms.
64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.18
.196) msg_seq=8 ttl=64rtt = 3.749099 ms.
64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.18
.196) msg_seq=9 ttl=64rtt = 3.901619 ms.
^C64 bytes from ham02s14-in-f196.1e100.net (h: www.google.com)(172.217.
18.196) msg_seq=10 ttl=64rtt = 3.586446 ms.

===172.217.18.196 ping statistics===

10 packets sent, 10 packets received, 0.000000 percent packet loss. Tot
al time: 9895.879857 ms.
```

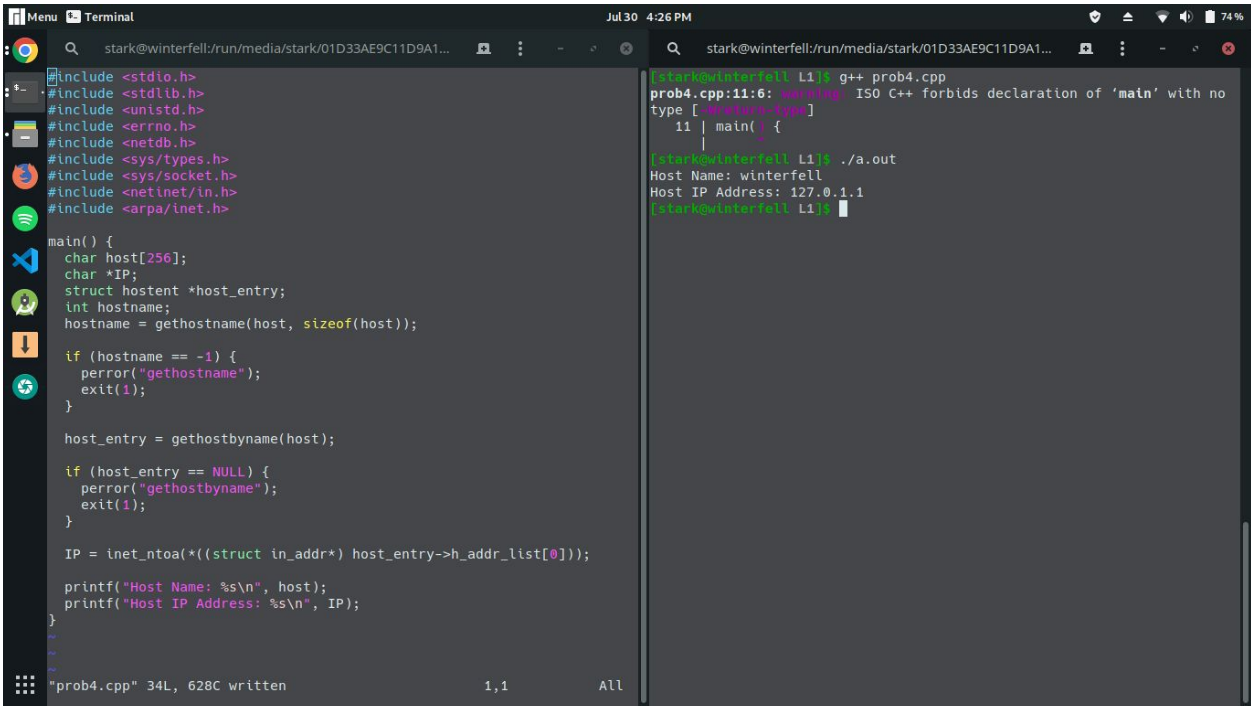
Problem 4

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

Data Structures and Algorithms used:

hostent: This data structure is used by functions to store information about a given host, such as hostname, 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.

Screenshots:


Menu
Terminal
Jul 30 4:26 PM
74 %
stark@winterfell:/run/media/stark/01D33AE9C11D9A1...
stark@winterfell:/run/media/stark/01D33AE9C11D9A1...

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#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);
}
~
~
~
"prob4.cpp" 34L, 628C written                          1,1            All
```

```
[stark@winterfell L1]$ g++ prob4.cpp
prob4.cpp:11:6: warning: ISO C++ forbids declaration of ‘main’ with no
type [-Wreturn-type]
   11 | main() {
      |
[stark@winterfell L1]$ ./a.out
Host Name: winterfell
Host IP Address: 127.0.1.1
[stark@winterfell L1]$ 
```

Thank You!