Assignment L1

(CSN-362)

Submitted by:

Divyanshu Salve

Enr. No: 17114027 (CS-1)

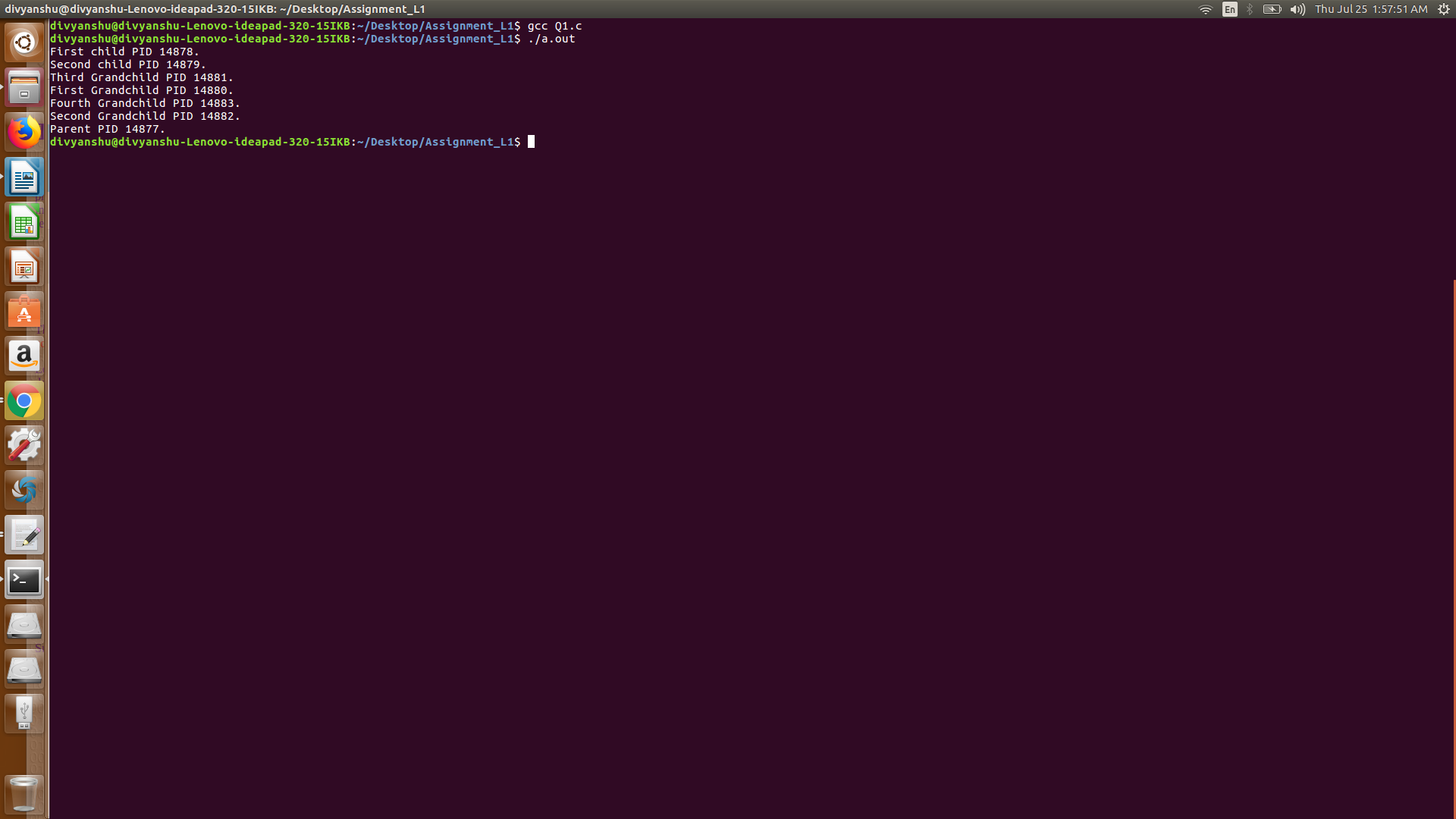
**Write a C program in the UNIX system that creates two children and four grandchildren (two for each child). The program should then print the process-IDs of the two children, four grandchildren and the parent in this order.**

⦁ In our code, we call the fork() system call 2 times.

⦁ When the parent calls the fork() for pid1 and pid2, both pid’s will be assigned positive values which are the two children of the parent C1 and C2(say).

⦁ While the execution of C1, pid1 will be 0 as it is a child process while pid2 will be the process ID of C1’s child, so we will fork the currently executing C1 again and get the second child of the first child C1.

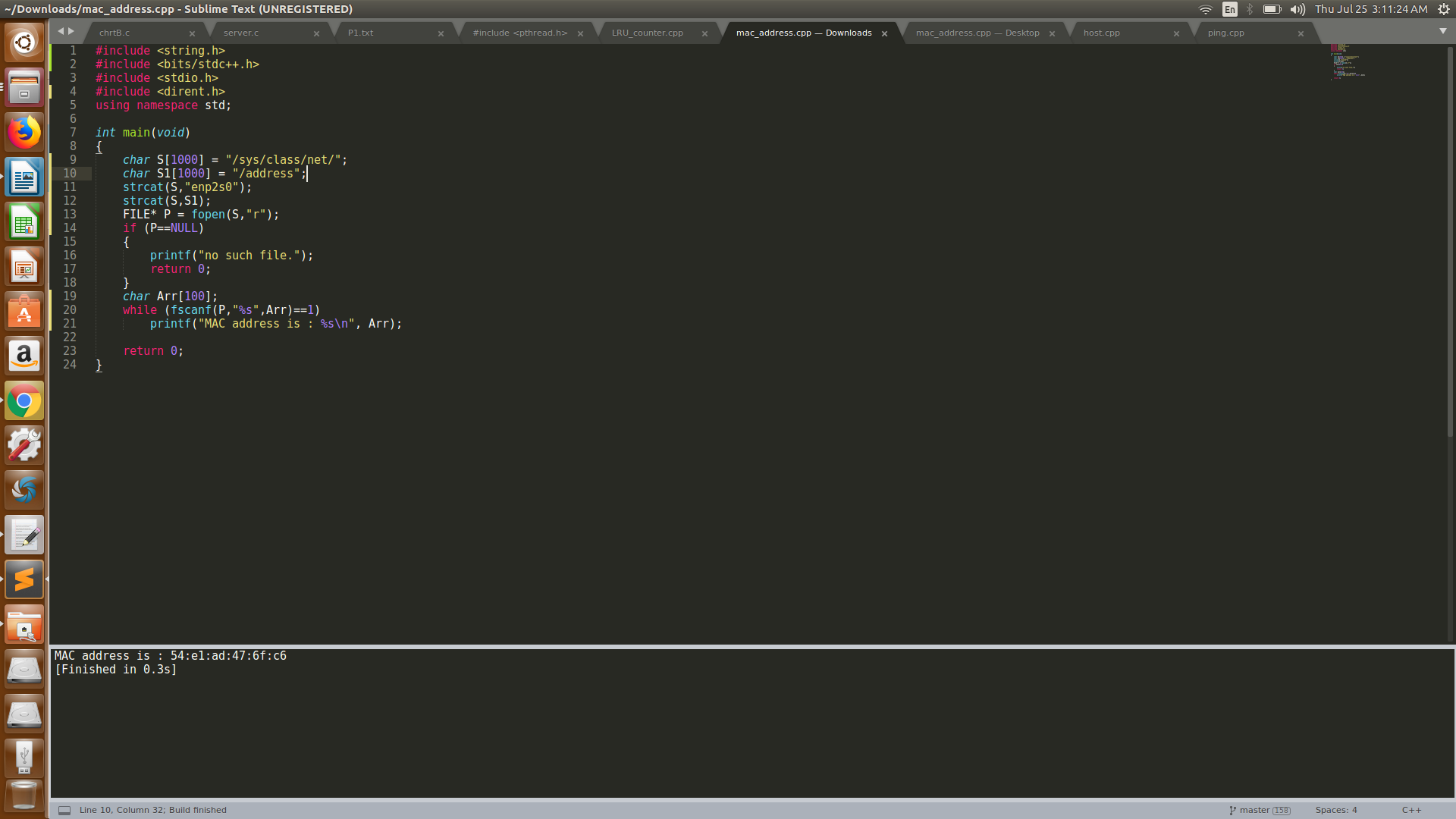
⦁ While the execution of C2, no common fork() system call is left for it, while the process id of p1 == -1 ,the process id of p2 == 0. So here, we fork it two times and print the values as the grandchildren p21 and p22.



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

⦁ The address in my Linux system where the MAC address is stored is "/sys/class/net/enp2s0/address/".

⦁ So we simply use concept of filehandling here and open the location to read the address.



**Write your own version of ping program in C**

#include <sstream>

#include <string>

/\*\*

\* @brief Convert String to Number

\*/

template <typename TP>

TP str2num( std::string const& value ){

std::stringstream sin;

sin << value;

TP output;

sin >> output;

return output;

}

/\*\*

\* @brief Convert number to string

\*/

template <typename TP>

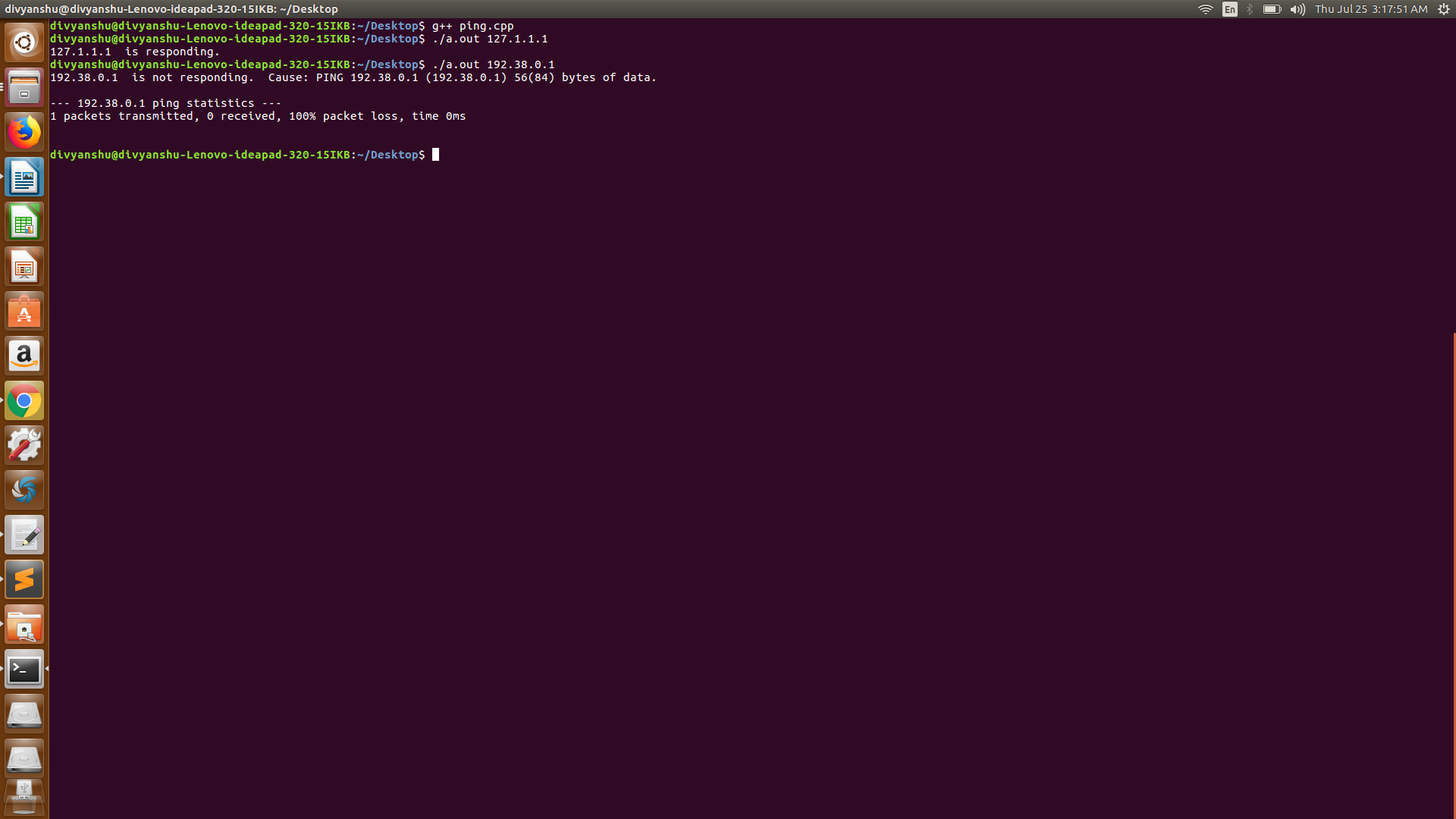
std::string num2str( TP const& value ){

std::stringstream sin;

sin << value;

return sin.str();

}



**Write a C program to find the host name and the IP Address of your computer**

#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>

void checkIPbuffer(char \*IPbuffer)

{

if (NULL == IPbuffer)

{

perror("inet\_ntoa");

exit(1);

}

}

void checkHostEntry(struct hostent \* hostentry)

{

if (hostentry == NULL)

{

perror("gethostbyname");

exit(1);

}

}

void checkHostName(int hostname)

{

if (hostname == -1)

{

perror("gethostname");

exit(1);

}

}

int main()

{

char hostbuffer[256];

char \*IPbuffer;

struct hostent \*host\_entry;

int hostname;

hostname = gethostname(hostbuffer, sizeof(hostbuffer));

checkHostName(hostname);

host\_entry = gethostbyname(hostbuffer);

checkHostEntry(host\_entry);

IPbuffer = inet\_ntoa(\*((struct in\_addr\*)

host\_entry->h\_addr\_list[0]));

printf("Hostname: %s\n", hostbuffer);

printf("Host IP: %s", IPbuffer);

return 0;

}