

Computer Networks Laboratory

CSN-361

Assignment 1 (L1)

Name: Twarit Waikar

Roll.no: 17114074

Class: B.Tech CSE 3rd Year

Problem Statements

1. C program in UNIX that creates two children and 2 grandchildren, and print their process IDs
2. C++ program to print MAC address
3. Ping program in C
4. C program to find hostname and IP of the system

Algorithms and Data Structures Used

Question 1:

The problem is visualised as a tree data structure where the parent node is represented by the parent process the child node is represented by the child processes.

When a fork statement is executed by the parent, a child process is created as you'd expect. You could say that the child process also executes the fork statement but returns a 0, the parent, however, returns the pid. All code after the fork statement is executed by both, the parent and the child.

Question 2:

On Linux-based systems the MAC address of an interface is obtained using the ioctl command SIOCGIFHWADDR. The method described here has five steps:

1. Create an ifreq structure for passing data in and out of ioctl.
2. Provide an open socket descriptor.
3. Invoke ioctl.

4. Check the type of the returned hardware address.
5. Extract the hardware address from the ifreq structure.

Question 3:

The algorithm is simply:

1. The ip packet is set up except checksum
2. IP_HDRINCL must be set on the socket so that the kernel does not attempt to automatically add a default ip header to the packet
3. The icmp packet is created also the ip checksum is generated
4. The packet is sent and then we wait for responses

Question 4:

Algorithm is simply:

1. Get host name by `gethostname()` in netdb.h
2. Get the host information by `gethostbyid()` in netdb.h
3. Process host information to a formatted string and print the result with the host name.

Snapshots

Question1

The screenshot shows a Visual Studio Code editor with a C program named `q1.c` and its execution output in the terminal. The program uses `fork()` to create multiple child processes, each printing its PID.

```

10  if (child1 == 0)
11  {
12      printf("This is a child1 process and pid is %d\n", getpid());
13
14      int child11 = fork();
15      if (child11 == 0)
16      {
17          printf("This is a child11 process and pid is %d\n", getpid());
18      }
19      else
20      {
21          int child12 = fork();
22          if (child12 == 0)
23          {
24              printf("This is a child12 process and pid is %d\n", getpid());
25          }
26      }
27  }
28  else
29  {
30      child2 = fork();
31      if (child2 == 0)
32      {
33          printf("This is a child2 process and pid is %d\n", getpid());
34
35          int child21 = fork();
36          if (child21 == 0)
37          {
38              printf("This is a child21 process and pid is %d\n", getpid());
39          }
40      }
41  }

```

The terminal output shows the execution of the program, displaying the PIDs of the parent and child processes:

```

[running] cd /home/serious/Desktop/netassign/ && gcc q1.c -o q1 && ./q1
This is a child1 process and pid is 16219
This is a child2 process and pid is 16220
This is a child2 process and pid is 16220
This is a child21 process and pid is 16221
This is a child2 process and pid is 16220
This is a child22 process and pid is 16224
This is a child1 process and pid is 16219
This is a child11 process and pid is 16222
This is a child1 process and pid is 16219
This is a child12 process and pid is 16223

```

Question 2

```
1 #include <iostream>
2 #include <stdio.h>
3 #include <sys/socket.h>
4 #include <arpa/inet.h>
5 #include <netinet/in.h>
6 #include <errno.h>
7 #include <string.h>
8 #include <stdlib.h>
9 #include <sys/ioctl.h>
10 #include <fcntl.h>
11 #include <net/if.h>
12 #include <unistd.h>
13
14 using namespace std;
15
16 void getMacAddress(char *uc_Mac);
17
18 int main()
19 {
20     char mac[32] = {0};
21
22     getMacAddress(mac);
23
24     cout << "Mac Address is : " << mac;
25     return 0;
26 }
27
28 void getMacAddress(char *uc_Mac)
29 {
30     // ...
31 }
```

```
[Running] cd "/home/serious/Desktop/netassign/" && g++ q2.cpp -o q2 && "/home/serious/Desktop/netassign/"q2
q2.cpp: In function 'void getMacAddress(char*)':
q2.cpp:33:16: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]
   char *iface = "eth0";
               ^
Mac Address is : 00:00:79:00:00:00
[Done] exited with code=0 in 0.275 seconds
```

Question 3

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <sys/socket.h>
5 #include <netinet/in.h>
6 #include <arpa/inet.h>
7 #include <netdb.h>
8 #include <linux/ip.h>
9 #include <linux/icmp.h>
10 #include <string.h>
11 #include <unistd.h>
12
13 char dst_addr[15];
14 char src_addr[15];
15
16 unsigned short in_cksum(unsigned short *, int);
17 void parse_args(char **, char *, char *);
18 void usage();
19 char *getip();
20
21 int main(int argc, char *argv[])
22 {
23     struct iphdr *ip;
24     struct iphdr *ip_reply;
25     struct icmphdr *icmp;
26     struct sockaddr_in connection;
27     char *packet;
28     char *buffer;
29
30     // ...
31 }
```

```
serious@predator:~/Desktop/netassign$ sudo ./q3 192.168.0.0
Source address: 192.168.227.107
Destination address: 192.168.0.0
Sent 28 byte packet to 192.168.0.0
```

Question 4

```
File Edit Selection View Go Debug Terminal Help
q4.c -- netassign - Visual Studio Code

C q4.c ...
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <errno.h>
5 #include <netdb.h>
6 #include <sys/types.h>
7 #include <sys/socket.h>
8 #include <netinet/in.h>
9 #include <arpa/inet.h>
10
11 void check_host_name(int hostname)
12 {
13     if (hostname == -1)
14     {
15         perror("gethostname");
16         exit(1);
17     }
18 }
19 void check_host_entry(struct hostent *hostentry)
20 {
21     if (hostentry == NULL)
22     {
23         perror("gethostbyname");
24         exit(1);
25     }
26 }
27 void IP_formatter(char *IPbuffer)
28 {
29     if (NULL == IPbuffer)
30     {
31         perror("IP buffer is NULL");
32         exit(1);
33     }
34 }
35
36 int main()
37 {
38     int hostname = -1;
39     struct hostent *hostentry = NULL;
40     char IPbuffer[16] = {0};
41
42     check_host_name(hostname);
43     check_host_entry(hostentry);
44     IP_formatter(IPbuffer);
45
46     printf("Current Host Name: %s\n", gethostname());
47     printf("Host IP: %s\n", IPbuffer);
48
49     return 0;
50 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

[Running] cd "/home/serious/Desktop/netassign/" && gcc q4.c -o q4 && "/home/serious/Desktop/netassign/"q4

q4.c:36:1: warning: return type defaults to 'int' [-Wimplicit-int]

main()

Current Host Name: predator

Host IP: 10.70.84.178

[Done] exited with code=0 in 0.074 seconds

IP_formatter(char *IPbuffer) Ln 33, Col 6 Tab Size: 4 UTF-8 LF C Linux