

Week 6 – Networking

Student number: 569091

Contents

- Assignment 6.1: Working from home..... 2
- Assignment 6.2: IP addresses websites 4
- Assignment 6.3: subnetting..... 6
- Assignment 6.4: HTML..... 7
- Assignment 6.5: Network segment 8

Assignment 6.1: Working from home

Screenshot installation openssh-server:

```
sudo apt install openssh-server
```

```
ray@ray-VMware-Virtual-Platform:~$ sudo systemctl status ssh
○ ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
   Active: inactive (dead)
 TriggeredBy: ● ssh.socket
   Docs: man:sshd(8)
        man:sshd_config(5)
lines 1-6/6 (END)
```

Screenshot successful SSH command execution:

```
C:\Users\Rajesh> ray@192.168.5.128
The authenticity of host '192.168.5.128 (192.168.5.128)' can't be established.
ED25519 key fingerprint is 50A256:V0aeedep3ylyAV2M6QuayZV8u6j4dMz/jIca.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.5.128' (ED25519) to the list of known hosts.
ray@192.168.5.128's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/support

Expanded Security Maintenance for Applications is not enabled.

12N updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ray@ray-VMware-Virtual-Platform:~$
```


Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```
ray@ray-Virtual-Platform:~$ nslookup amazon.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   amazon.com
Address: 52.94.236.248
Name:   amazon.com
Address: 205.251.242.103
Name:   amazon.com
Address: 54.239.28.85

ray@ray-Virtual-Platform:~$ nslookup google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.168.206
Name:   google.com
Address: 208.145.0.80#803:200e

ray@ray-Virtual-Platform:~$ nslookup one.one.one.one
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   one.one.one.one
Address: 1.1.1.1
Name:   one.one.one.one
Address: 1.0.0.1
Name:   one.one.one.one
Address: 206.4700.4700:1001
Name:   one.one.one.one
Address: 2606:4700:4700:1111

ray@ray-Virtual-Platform:~$ nslookup dns.google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   dns.google.com
Address: 8.8.8.8
Name:   dns.google.com
Address: 8.8.4.4
Name:   dns.google.com
Address: 2081:4860:4860:8888
Name:   dns.google.com
Address: 2081:4860:4860:8844

ray@ray-Virtual-Platform:~$ nslookup bol.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
```

```
Non-authoritative answer:
Name:   google.com
Address: 172.217.168.206
Name:   google.com
Address: 208.145.0.80#803:200e

ray@ray-Virtual-Platform:~$ nslookup one.one.one.one
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   one.one.one.one
Address: 1.1.1.1
Name:   one.one.one.one
Address: 1.0.0.1
Name:   one.one.one.one
Address: 2066:4700:4700:1001
Name:   one.one.one.one
Address: 2606:4700:4700:1111

ray@ray-Virtual-Platform:~$ nslookup dns.google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   dns.google.com
Address: 8.8.8.8
Name:   dns.google.com
Address: 8.8.4.4
Name:   dns.google.com
Address: 2081:4860:4860:8888
Name:   dns.google.com
Address: 2081:4860:4860:8844

ray@ray-Virtual-Platform:~$ nslookup bol.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   bol.com
Address: 34.36.121.47

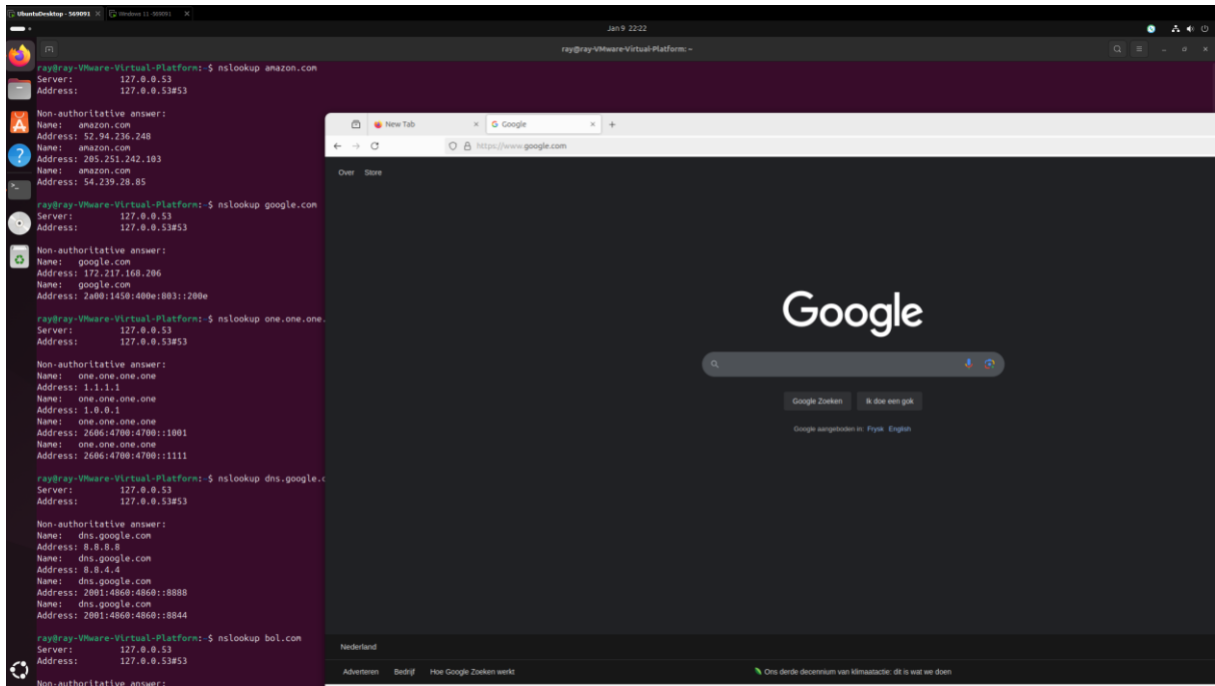
ray@ray-Virtual-Platform:~$ nslookup w3schools.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   w3schools.com
Address: 13.248.248.135
Name:   w3schools.com
Address: 76.223.115.82

ray@ray-Virtual-Platform:~$
```

Screenshot website visit via IP address:

Used the google IP address 172.217.168.206



Assignment 6.3: subnetting

How many IP addresses are in this network configuration 192.168.110.128/25?

A /25 subnet has 128 IP addresses.

What is the usable IP range to hand out to the connected computers?

The usable range for the 192.168.110.128/25 subnet is from 192.168.110.129 to 192.168.110.254.

Check your two previous answers with this Linux command: `ipcalc 192.168.110.128/25`

Assignment from last year instructed to use [this](#) calculator:

The screenshot shows a web-based IP Subnet Calculator. The input IP address is 192.168.110.128 and the subnet mask is 255.255.255.128. The results are as follows:

Field	Value
IP Address	192.168.110.128
Network Address	192.168.110.128
Usable Host IP Range	192.168.110.129 - 192.168.110.254
Broadcast Address	192.168.110.255
Total Number of Hosts	128
Number of Usable Hosts	128
Subnet Mask	255.255.255.128
Wildcard Mask	0.0.0.127
Binary Subnet Mask	11111111 11111111 11111111 10000000
IP Class	C
CIDR Notation	/25
IP Type	Private
Short	192.168.110.128 /25
Binary ID	11000000101010000110110100000000
Integer ID	3232263808
Hex ID	0xc0a86e80
in-addr.arpa	128.110.168.192.in-addr.arpa
IPv4 Mapped Address	:::c0a8:6e80
IPv4 Prefix	2002:c0a8:6e80::/48

Below the results, there is a table titled "All 2 of the Possible /25 Networks for 192.168.110.*":

Network Address	Usable Host Range	Broadcast Address
192.168.110.0	192.168.110.1 - 192.168.110.126	192.168.110.127
192.168.110.128	192.168.110.129 - 192.168.110.254	192.168.110.255

The calculator interface also includes radio buttons for Network Class (Any, A, B, C), a Subnet input field (255.255.255.128 /25), and an IP Address input field (192.168.110.128). There are "Calculate" and "Clear" buttons at the bottom.

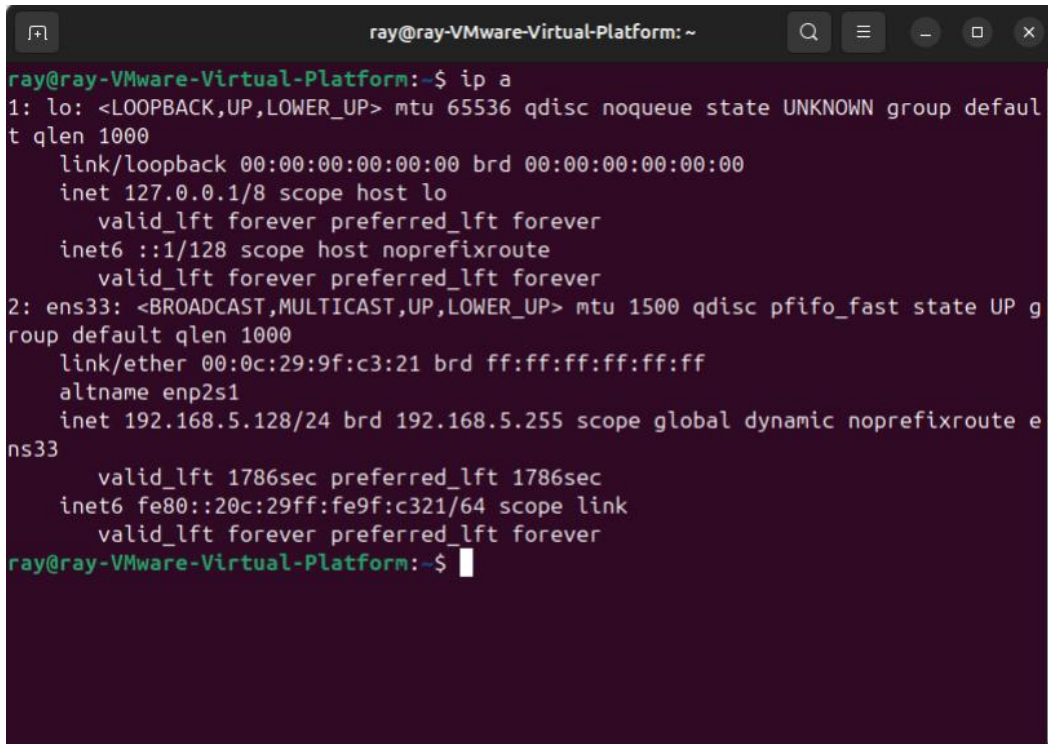
This screenshot will be used as a placeholder until the environment from last year (2024-2025) is rebuilt to gain access to Linux commands.

Explain the above calculation in your own words.

A /25 subnet mask splits the IP range into two parts: the network and the usable IPs. The first 128 IPs belong to the network (192.168.110.0 to 192.168.110.127), and the remaining 128 IPs are usable by devices (192.168.110.128 to 192.168.110.254).

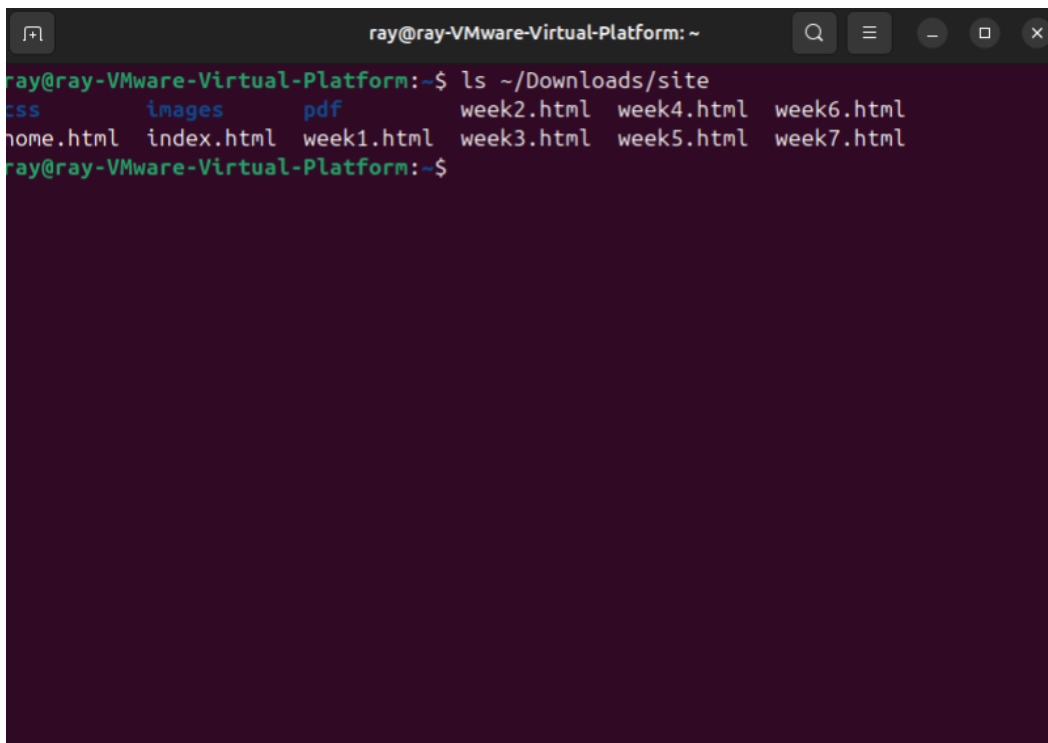
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



```
ray@ray-VMware-Virtual-Platform: ~  
ray@ray-VMware-Virtual-Platform:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 00:0c:29:9f:c3:21 brd ff:ff:ff:ff:ff:ff  
    altname enp2s1  
    inet 192.168.5.128/24 brd 192.168.5.255 scope global dynamic noprefixroute ens33  
        valid_lft 1786sec preferred_lft 1786sec  
    inet6 fe80::20c:29ff:fe9f:c321/64 scope link  
        valid_lft forever preferred_lft forever  
ray@ray-VMware-Virtual-Platform:~$
```

Screenshot of Site directory contents:



```
ray@ray-VMware-Virtual-Platform: ~  
ray@ray-VMware-Virtual-Platform:~$ ls ~/Downloads/site  
css          images      pdf          week2.html  week4.html  week6.html  
home.html   index.html  week1.html  week3.html  week5.html  week7.html  
ray@ray-VMware-Virtual-Platform:~$
```

Screenshot python3 webserver command:

```
ray@ray-VMware-Virtual-Platform:~$ cd ~/Downloads/site
ray@ray-VMware-Virtual-Platform:~/Downloads/site$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Screenshot web browser visits your site



Hello World!

Calisthenics is an alternative training style to lifting weights that uses your own bodyweight.

Assignment 6.5: Network segment

Remember that bitwise java application you've made in week 2? Expand that application so that you can also calculate a network segment as explained in the PowerPoint slides of week 6. Use the bitwise & AND operator. You need to be able to input two Strings. An IP address and a subnet.

IP: 192.168.1.100 and subnet: 255.255.255.224 for /27

Example: 192.168.1.100/27

Calculate the network segment

IP Address: 11000000.10101000.00000001.01100100

Subnet Mask: 11111111.11111111.11111111.11100000

Network Addr: 11000000.10101000.00000001.01100000

This gives 192.168.1.96 in decimal as the network address.

For a /27 subnet, each segment (or subnet) has 32 IP addresses (2^5).

The range of this network segment is from 192.168.1.96 to 192.168.1.127.

Source code

```
public class Main {
    public static void main(String[] args) {
        // Example IP and subnet
        String ip = "192.168.1.100";
        String subnet = "255.255.255.224";

        // Split the IP and subnet into octets
        String[] ipOctets = ip.split("\\.");
        String[] subnetOctets = subnet.split("\\.");

        // Convert the octets into binary
        String ipBinary = "";
        String subnetBinary = "";

        for (int i = 0; i < 4; i++) {
            ipBinary += String.format("%08d",
                Integer.parseInt(Integer.toBinaryString(Integer.parseInt(ipOctets[i]))));
            subnetBinary += String.format("%08d",
                Integer.parseInt(Integer.toBinaryString(Integer.parseInt(subnetOctets[i]))));
        }

        // Print the binary representation of IP and subnet mask
        System.out.println("IP Address (binary): " + ipBinary);
        System.out.println("Subnet Mask (binary): " + subnetBinary);

        // Perform the bitwise AND operation
        StringBuilder networkBinary = new StringBuilder();
        for (int i = 0; i < ipBinary.length(); i++) {
            char ipBit = ipBinary.charAt(i);
            char subnetBit = subnetBinary.charAt(i);

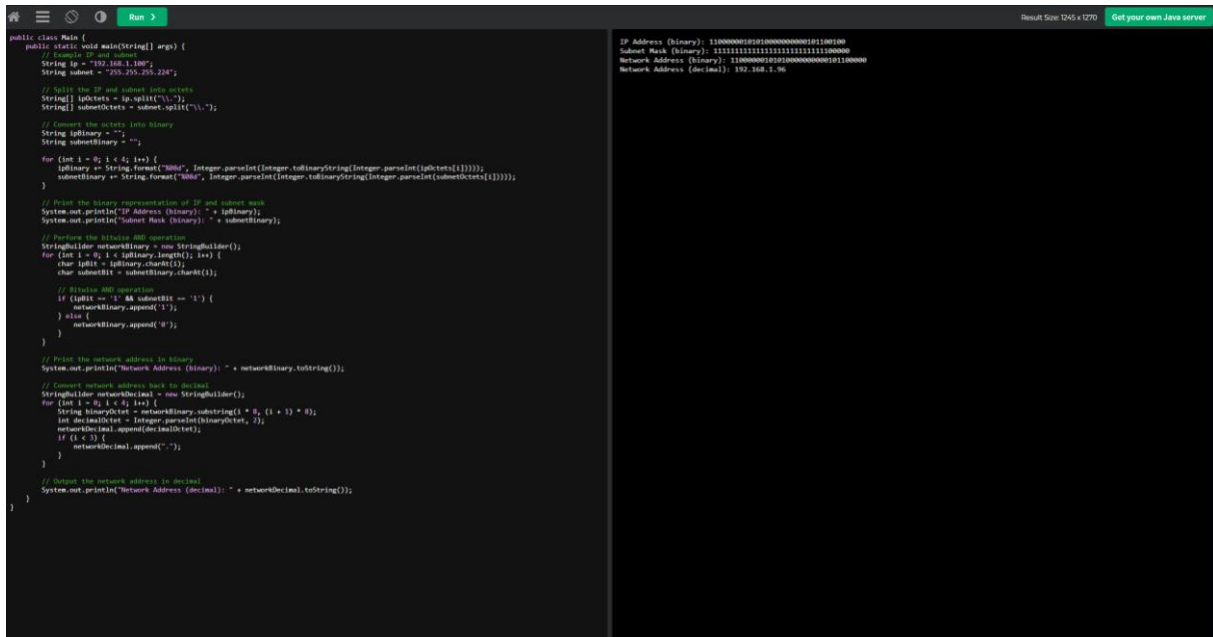
            // Bitwise AND operation
            if (ipBit == '1' && subnetBit == '1') {
                networkBinary.append('1');
            } else {
                networkBinary.append('0');
            }
        }

        // Print the network address in binary
        System.out.println("Network Address (binary): " + networkBinary.toString());

        // Convert network address back to decimal
        StringBuilder networkDecimal = new StringBuilder();
        for (int i = 0; i < 4; i++) {
            String binaryOctet = networkBinary.substring(i * 8, (i + 1) * 8);
            int decimalOctet = Integer.parseInt(binaryOctet, 2);
            networkDecimal.append(decimalOctet);
            if (i < 3) {
                networkDecimal.append(".");
            }
        }

        // Output the network address in decimal
        System.out.println("Network Address (decimal): " + networkDecimal.toString());
    }
}
```

Screenshot



```
public class Main {
    public static void main(String[] args) {
        // Example IP and subnet
        String ip = "192.168.1.100";
        String subnet = "255.255.255.224";

        // Split the IP and subnet into octets
        String[] ipOctets = ip.split("\\.");
        String[] subnetOctets = subnet.split("\\.");

        // Convert the octets into binary
        String ipBinary = "";
        String subnetBinary = "";

        for (int i = 0; i < 4; i++) {
            ipBinary += String.format("%08s", Integer.parseInt(Integer.toString(Integer.parseInt(ipOctets[i]))));
            subnetBinary += String.format("%08s", Integer.parseInt(Integer.toString(Integer.parseInt(subnetOctets[i]))));
        }

        // Print the binary representation of IP and subnet mask
        System.out.println("IP Address (binary): " + ipBinary);
        System.out.println("Subnet Mask (binary): " + subnetBinary);

        // Perform the bitwise AND operation
        StringBuilder networkBinary = new StringBuilder();
        for (int i = 0; i < ipBinary.length(); i++) {
            char ipBit = ipBinary.charAt(i);
            char subnetBit = subnetBinary.charAt(i);

            // Bitwise AND operation
            if (ipBit == '1' && subnetBit == '1') {
                networkBinary.append('1');
            } else {
                networkBinary.append('0');
            }
        }

        // Print the network address in binary
        System.out.println("Network Address (binary): " + networkBinary.toString());

        // Convert network address back to decimal
        StringBuilder networkDecimal = new StringBuilder();
        for (int i = 0; i < 4; i++) {
            String binaryOctet = networkBinary.substring(i * 8, (i + 1) * 8);
            int decimalOctet = Integer.parseInt(Integer.toString(binaryOctet), 2);
            networkDecimal.append(decimalOctet);

            if (i < 3) {
                networkDecimal.append(".");
            }
        }

        // Output the network address in decimal
        System.out.println("Network Address (decimal): " + networkDecimal.toString());
    }
}
```

IP Address (binary): 11000000110100000000000101100100
Subnet Mask (binary): 11111111111111111111111111000000
Network Address (binary): 11000000110100000000000101100100
Network Address (decimal): 192.168.1.96