

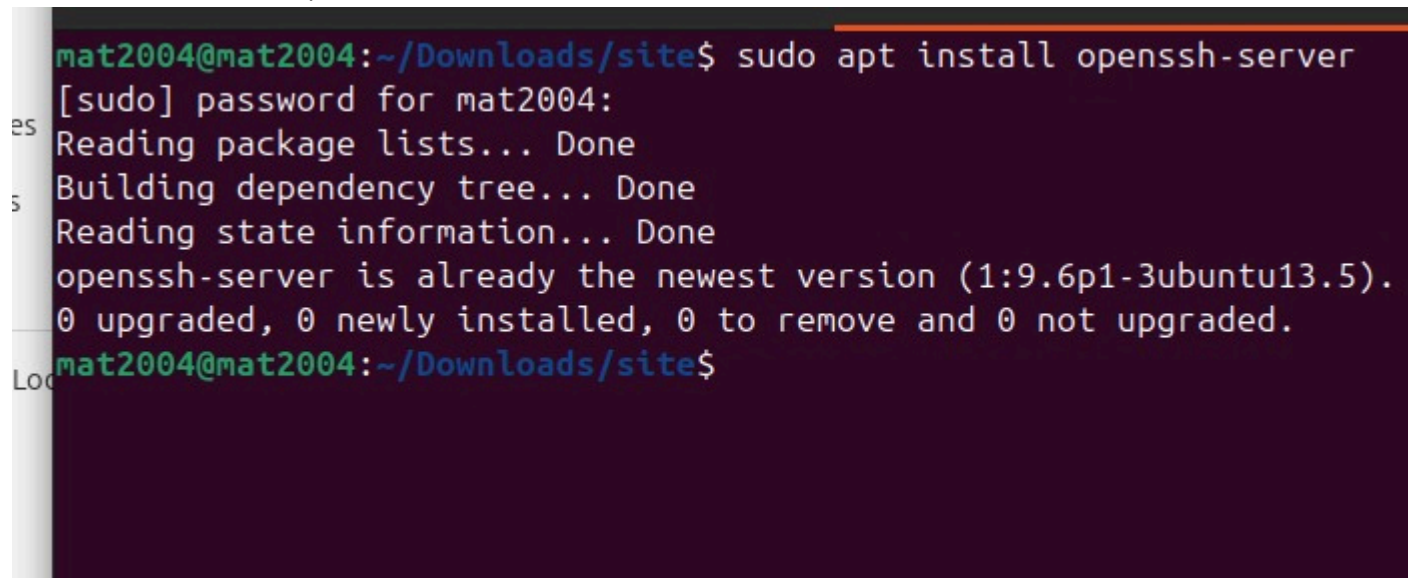
Template Week 6 – Networking

Student number:

545676

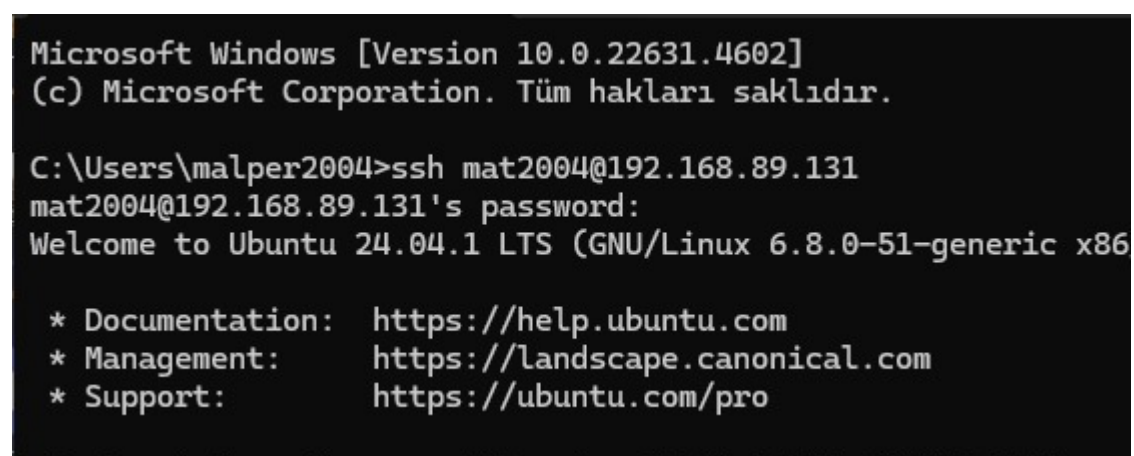
Assignment 6.1: Working from home

Screenshot installation openssh-server:

A terminal window with a dark purple background. The prompt is 'mat2004@mat2004:~/Downloads/site\$'. The user enters 'sudo apt install openssh-server'. The terminal shows the password prompt, then 'Reading package lists... Done', 'Building dependency tree... Done', and 'Reading state information... Done'. It then states 'openssh-server is already the newest version (1:9.6p1-3ubuntu13.5). 0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.' The prompt returns to 'mat2004@mat2004:~/Downloads/site\$'.

```
mat2004@mat2004:~/Downloads/site$ sudo apt install openssh-server
[sudo] password for mat2004:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
mat2004@mat2004:~/Downloads/site$
```

Screenshot successful SSH command execution:

A Windows terminal window with a black background. It shows the Windows version '10.0.22631.4602' and copyright information. The user runs 'ssh mat2004@192.168.89.131'. The terminal shows the password prompt, then 'Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic x86_64)'. It lists documentation, management, and support links.

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\malper2004>ssh mat2004@192.168.89.131
mat2004@192.168.89.131's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic x86_64)

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

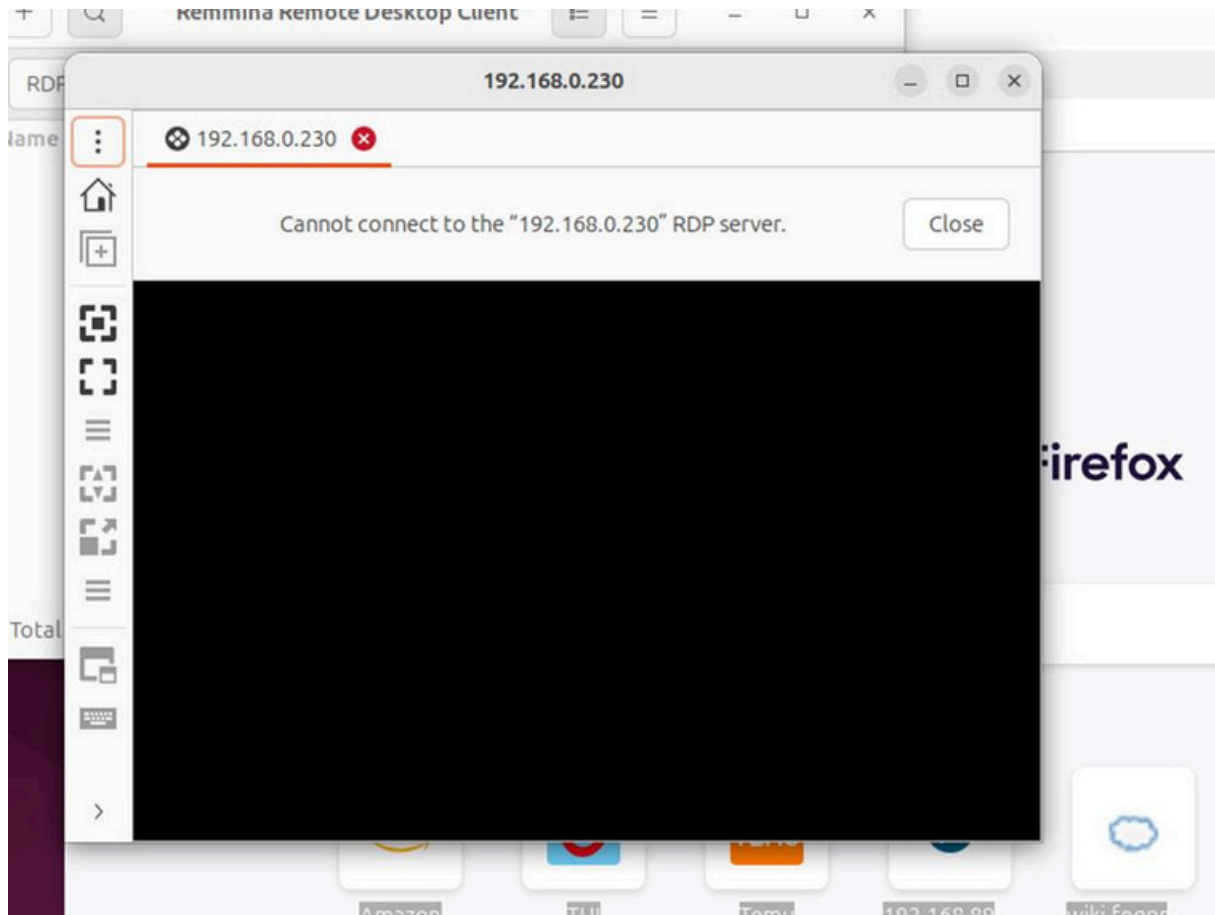
Screenshot successful execution SCP command:

```
Administrator: Komut İstemi
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Windows\System32>scp "C:\Users\malper2004\Desktop\example.txt" mat2004@192.168.89.131:/home/mat2004/
mat2004@192.168.89.131's password:
example.txt                                     100%   0   0.0KB/s   00:00

C:\Windows\System32>
```

Screenshot remmina:



Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

```
mat2004@mat2004:~$ nslookup amazon.com
Server:      127.0.0.53
Address:     127.0.0.53#53

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

Screenshot website visit via IP address:



Assignment 6.3: subnetting

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

128

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

Between 192.168.110.128 - 192.168.110.255

Check your two previous answers with this calculator:

<https://www.calculator.net/ip-subnet-calculator.html>

Explain the above calculation in your own words.

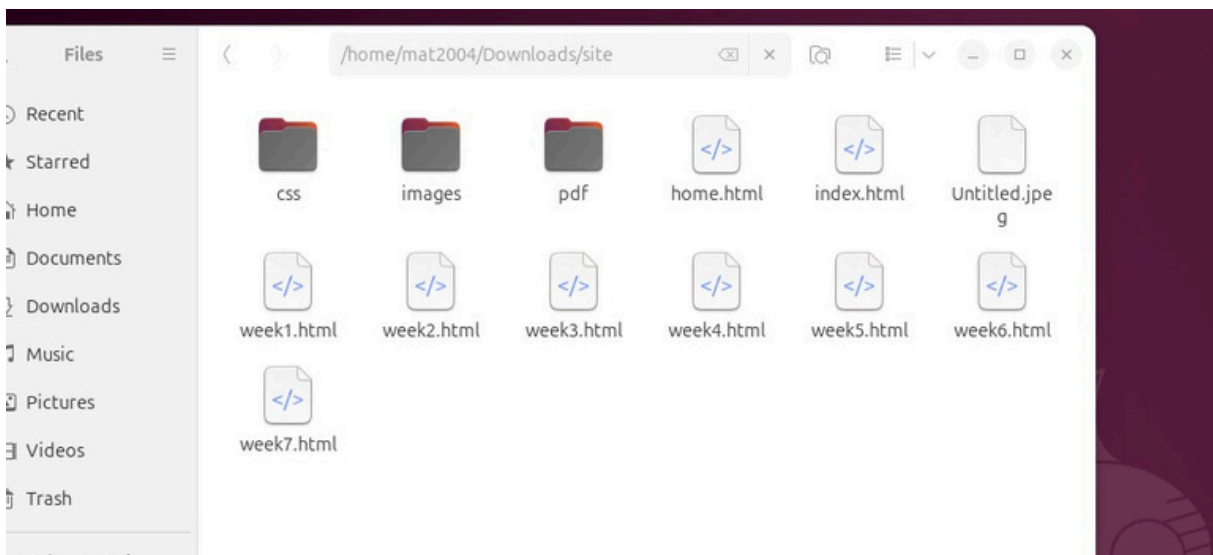
First answer uses 25 bits. Second answer 2 over 7.

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

```
mat2004@mat2004:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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 fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:7e:0d:ff brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.89.131/24 brd 192.168.89.255 scope global dynamic noprefixroute ens33
        valid_lft 1058sec preferred_lft 1058sec
    inet6 fe80::20c:29ff:fe7e:dff/64 scope link
        valid_lft forever preferred_lft forever
mat2004@mat2004:~$
```

Screenshot of Site directory contents:



Screenshot python3 webserver command:

```
mat2004@mat2004:~/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



Bonus point assignment – week 6

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 (25).

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

Paste source code here, with a screenshot of a import java.util.Scanner;

```

public class NetworkCalculator {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter IP address and subnet (e.g., 192.168.1.100/27): ");
        String input = scanner.nextLine();

        String[] parts = input.split("/");
        String ip = parts[0];
        int subnetPrefix = Integer.parseInt(parts[1]);

        int[] ipAddress = ipToBinary(ip);
        int[] subnetMask = createSubnetMask(subnetPrefix);

        int[] networkAddress = new int[4];
        for (int i = 0; i < 4; i++) {
            networkAddress[i] = ipAddress[i] & subnetMask[i];
        }

        System.out.println("Network Address: " + binaryToDecimal(networkAddress));

        int networkStart = ipToInt(networkAddress);
        int networkEnd = networkStart + (int) Math.pow(2, 32 - subnetPrefix) - 1;
    }
}

```

```

        System.out.println("IP Range: " + intToIp(networkStart) + " - " + intToIp(networkEnd));
    }

```

```

public static int[] ipToBinary(String ip) {
    String[] octets = ip.split("\\.");
    int[] binaryIp = new int[4];
    for (int i = 0; i < 4; i++) {
        binaryIp[i] = Integer.parseInt(octets[i]);
    }
    return binaryIp;
}

```

```

public static int[] createSubnetMask(int prefix) {
    int[] subnetMask = new int[4];
    for (int i = 0; i < 4; i++) {
        if (prefix >= 8) {
            subnetMask[i] = 255;
            prefix -= 8;
        } else if (prefix > 0) {
            subnetMask[i] = (int) (256 - Math.pow(2, 8 - prefix));
            prefix = 0;
        } else {
            subnetMask[i] = 0;
        }
    }
    return subnetMask;
}

```

```

// Method to convert binary IP address to decimal string format

```

```

public static String binaryToDecimal(int[] binaryIp) {
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < 4; i++) {
        sb.append(binaryIp[i]);
        if (i < 3) sb.append(".");
    }
    return sb.toString();
}

```

```

public static int ipToInt(int[] ip) {
    return (ip[0] << 24) | (ip[1] << 16) | (ip[2] << 8) | ip[3];
}

```

```

public static String intToIp(int ip) {
    return ((ip >> 24) & 0xFF) + "." + ((ip >> 16) & 0xFF) + "." + ((ip >> 8) & 0xFF) + "." + (ip & 0xFF);
}

```

}working application.

Output

```
Enter IP address and subnet (e.g., 192.168.1.100/27): |
```

Ready? Save this file and export it as a pdf file with the name: [week6.pdf](#)