

Template Week 6 – Networking

Student number: 580521

Assignment 6.1: Working from home

Screenshot installation openssh-server:

```
constantin@580521:~$ sudo apt install openssh-server
[sudo] password for constantin:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-client openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 4 newly installed, 0 to remove and 179 not upgraded.
Need to get 1,738 kB of archives.
After this operation, 6,743 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://nl.archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-client amd64 1:9.6p1-3ubuntu13.14 [906 kB]
Get:2 http://nl.archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-sftp-server amd64 1:9.6p1-3ubuntu13.14 [37.3 kB]
Get:3 http://nl.archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-server amd64 1:9.6p1-3ubuntu13.14 [510 kB]
Get:4 http://nl.archive.ubuntu.com/ubuntu noble/main amd64 ncurses-term all 6.4+
```

Screenshot successful SSH command execution:

```
C:\Users\580521>ssh constantin@192.168.139.128
The authenticity of host '192.168.139.128 (192.168.139.128)' can't be established.
ED25519 key fingerprint is SHA256:1zOqRerZWwgGDFjPbo+77sLOfdUHHJNBvCow7zSZYio4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.139.128' (ED25519) to the list of known hosts.
constantin@192.168.139.128's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-36-generic x86_64)

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

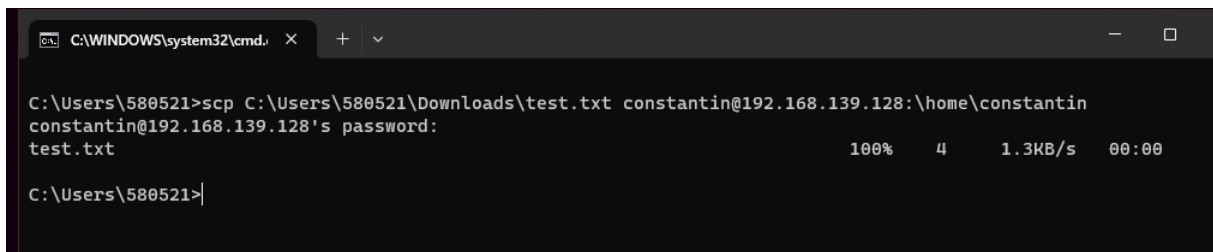
Expanded Security Maintenance for Applications is not enabled.

187 updates can be applied immediately.
36 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

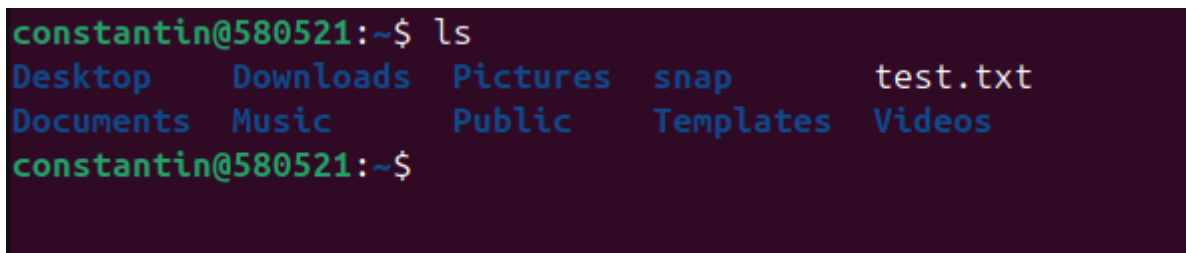
Last login: Mon Dec 29 12:23:11 2025 from 192.168.139.1
constantin@580521:~$ |
```

Screenshot successful execution SCP command:



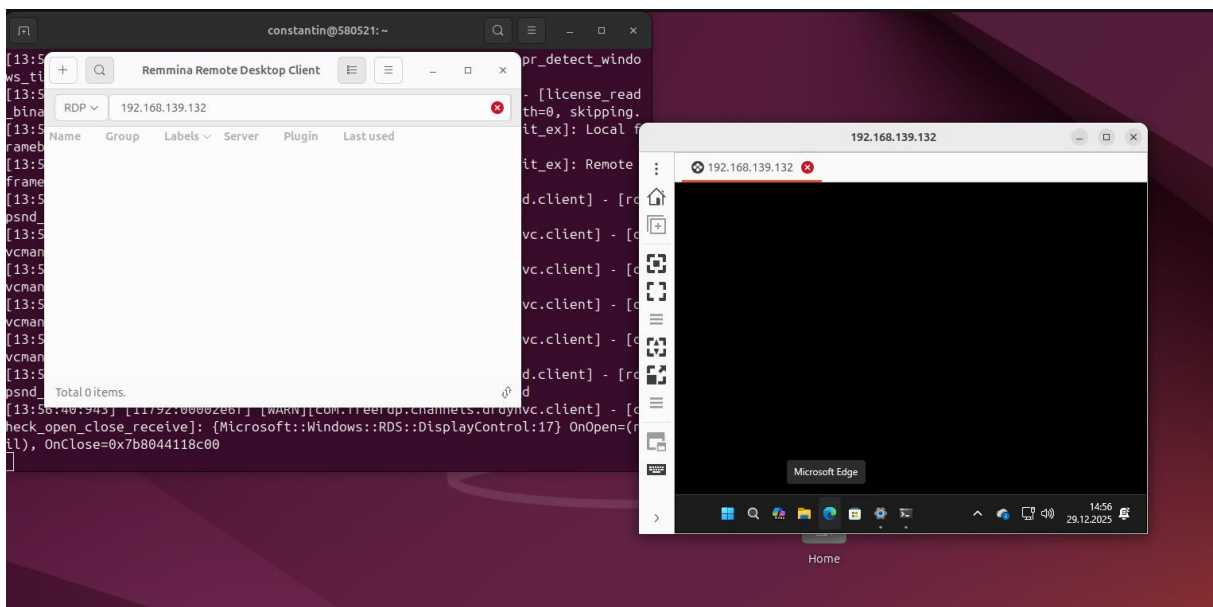
```
C:\WINDOWS\system32\cmd.exe
C:\Users\580521>scp C:\Users\580521\Downloads\test.txt constantin@192.168.139.128:/home/constantin
constantin@192.168.139.128's password:
test.txt                                100%   4    1.3KB/s   00:00

C:\Users\580521>
```



```
constantin@580521:~$ ls
Desktop    Downloads  Pictures   snap       test.txt
Documents  Music      Public     Templates  Videos
constantin@580521:~$
```

Screenshot remmina:



Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

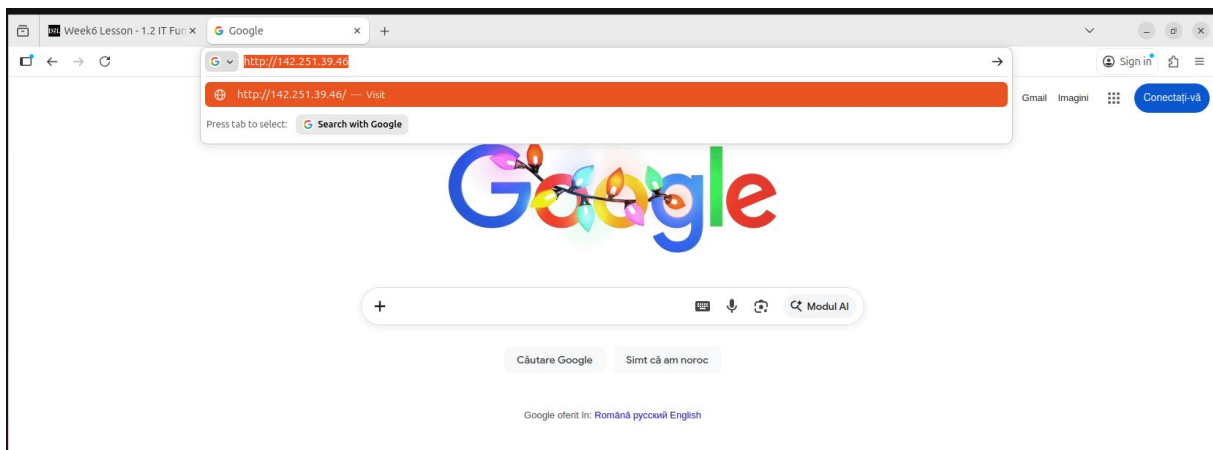
```
constantin@580521:~$ nslookup
>
> amazon.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   amazon.com
Address: 98.87.170.71
Name:   amazon.com
Address: 98.87.170.74
Name:   amazon.com
Address: 98.82.161.185
> google.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.251.39.46
Name:   google.com
Address: 2a00:1450:400d:80d::200e
> 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: 2606:4700:4700::1111
Name:   one.one.one.one
Address: 2606:4700:4700::1001
> dns.google.com
```

Screenshot website visit via IP address:



Assignment 6.3: subnetting

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

2 to the power of 7 = 128

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

192.168.110.129 → 192.168.110.254

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

```
constantin@580521:~$ ipcalc 192.168.110.128/25
Address: 192.168.110.128      11000000.10101000.01101110.1 00000000
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 00000000
Wildcard: 0.0.0.127          00000000.00000000.00000000.0 11111111
=>
Network: 192.168.110.128/25  11000000.10101000.01101110.1 00000000
HostMin: 192.168.110.129    11000000.10101000.01101110.1 00000001
HostMax: 192.168.110.254    11000000.10101000.01101110.1 11111110
Broadcast: 192.168.110.255  11000000.10101000.01101110.1 11111111
Hosts/Net: 126               Class C, Private Internet

constantin@580521:~$
```

Explain the above calculation in your own words.

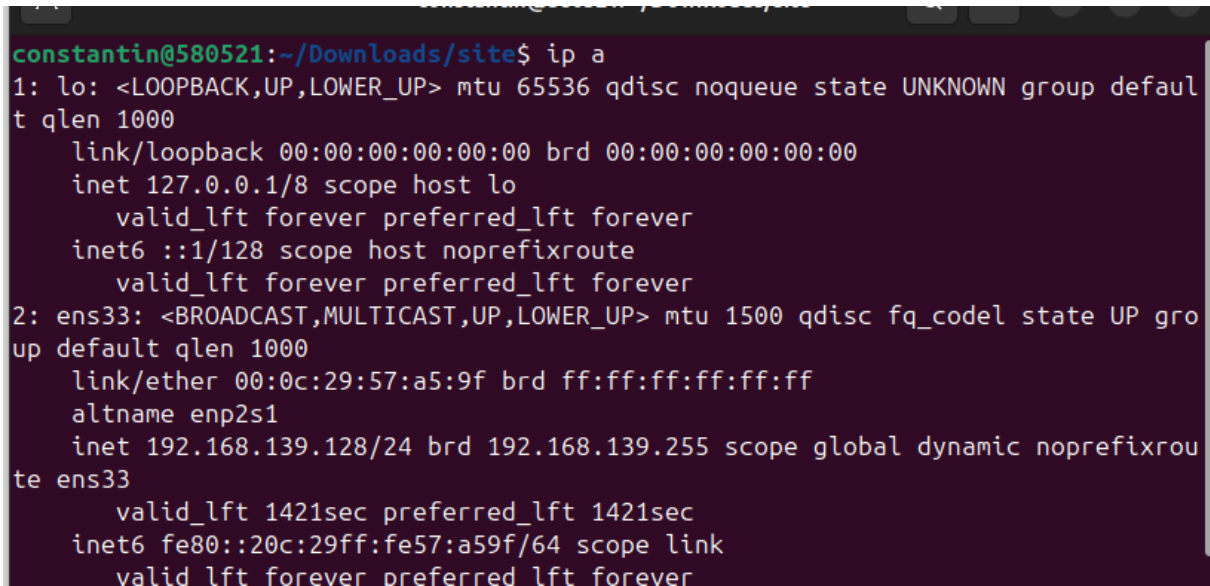
The /25 is the amount of used bits and the rest are free for the hosts, so number of available addresses is 2 to the power of 7 which is 32-25

The result is 128 but 2 are reserved being the first one x.x.x.128 and x.x.x.255

So the remaining ip addresses can be used for hosts

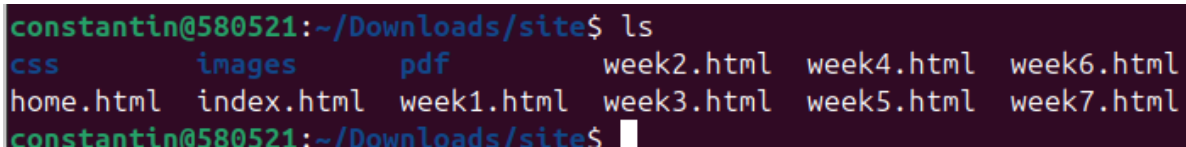
Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:



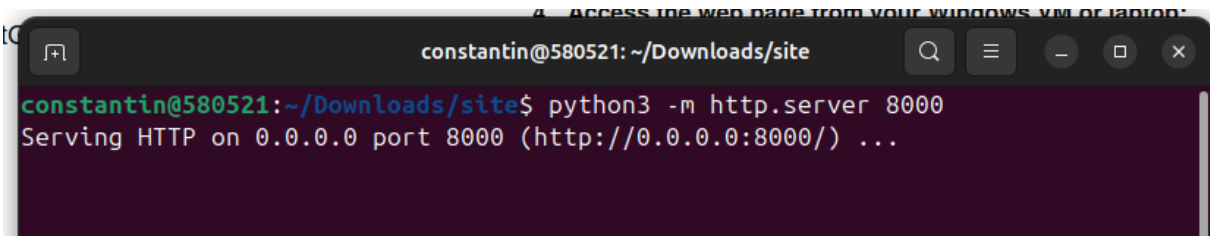
```
constantin@580521:~/Downloads/site$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope 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:29:57:a5:9f brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.139.128/24 brd 192.168.139.255 scope global dynamic noprefixroute ens33
        valid_lft 1421sec preferred_lft 1421sec
    inet6 fe80::20c:29ff:fe57:a59f/64 scope link
        valid_lft forever preferred_lft forever
```

Screenshot of Site directory contents:



```
constantin@580521:~/Downloads/site$ ls
css      images      pdf          week2.html  week4.html  week6.html
home.html index.html  week1.html  week3.html  week5.html  week7.html
constantin@580521:~/Downloads/site$
```

Screenshot python3 webserver command:



```
constantin@580521: ~/Downloads/site
constantin@580521:~/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



Welcome to My Basketball Hobby Page!

Basketball is one of my favorite hobbies! I love playing it with my friends after school. It's a great way to stay active and have fun. Whether it's shooting hoops at the park or playing a full game, basketball always helps me relax and enjoy the moment.

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.

Paste source code here, with a screenshot of a working application.

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter IP address: ");
        String ip = scanner.nextLine();

        System.out.print("Enter subnet mask: ");
        String subnet = scanner.nextLine();
    }
}
```

```

String[] ipParts = ip.split("\\.");
String[] subnetParts = subnet.split("\\.");

int[] ipNums = new int[4];
int[] subnetNums = new int[4];
int[] networkNums = new int[4];

for (int i = 0; i < 4; i++) {
    ipNums[i] = Integer.parseInt(ipParts[i]);
    subnetNums[i] = Integer.parseInt(subnetParts[i]);
    networkNums[i] = ipNums[i] & subnetNums[i];
}

System.out.print("IP Address: ");
for (int i = 0; i < 4; i++) {
    System.out.print(String.format("%8s",
Integer.toBinaryString(ipNums[i])).replace(' ', '0'));
    if (i < 3) System.out.print(".");
}
System.out.println();

System.out.print("Subnet Mask: ");
for (int i = 0; i < 4; i++) {
    System.out.print(String.format("%8s",
Integer.toBinaryString(subnetNums[i])).replace(' ', '0'));
    if (i < 3) System.out.print(".");
}
System.out.println();

System.out.print("Network Addr: ");
for (int i = 0; i < 4; i++) {
    System.out.print(String.format("%8s",
Integer.toBinaryString(networkNums[i])).replace(' ', '0'));
    if (i < 3) System.out.print(".");
}
System.out.println();

System.out.print("This gives ");
for (int i = 0; i < 4; i++) {
    System.out.print(networkNums[i]);
    if (i < 3) System.out.print(".");
}
System.out.println(" in decimal as the network address.");

int totalIPs = 32;

int lastOctetStart = networkNums[3];
int lastOctetEnd = lastOctetStart + totalIPs - 1;

System.out.println("For a /27 subnet, each segment (or
subnet) has 32 IP addresses (2^5).");

```

```

        System.out.println("The range of this network segment is
from " + networkNums[0] + "." + networkNums[1] + "." +
networkNums[2] + "." + lastOctetStart +
        " to " + networkNums[0] + "." +
networkNums[1] + "." + networkNums[2] + "." + lastOctetEnd + ".");
    }
}

```

```

Enter IP address: 192.168.1.100
Enter subnet mask: 255.255.255.224
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.

=== Code Execution Successful ===

```

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