

Template Week 6 – Networking

Student number: 571994

Assignment 6.1: Working from home

Screenshot installation openssh-server:

Screenshot successful SSH command execution:

Screenshot successful execution SCP command:

Screenshot remmina:

Assignment 6.2: IP addresses websites

Relevant screenshots nslookup command:

Screenshot website visit via IP address:

Assignment 6.3: subnetting

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

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

Check your two previous answers with this calculator:

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

Explain the above calculation in your own words.

Assignment 6.4: HTML

Screenshot IP address Ubuntu VM:

Screenshot of Site directory contents:

Screenshot python3 webserver command:

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 (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.

Maak een keuze uit het menu:

1. Controleer of het getal oneven is
2. Controleer of het getal een macht van 2 is
3. Bereken het twee-complement van het getal
4. Bereken het netwerksegment
5. Sluit de applicatie af

Uw keuze: 4

Voer een geldig IP-adres in (bijvoorbeeld 192.168.1.10): 192.168.1.10

Geef een subnetmasker op (bijvoorbeeld 255.255.255.0): 255.255.255.0

Netwerkadres (decimaal): 192.168.1.0

Netwerkadres (binair): 11000000101010000000000100000000

```
System.out.println("4. Bereken het netwerksegment");
System.out.println("5. Sluit de applicatie af");
System.out.print("Uw keuze: ");
int choice = scanner.nextInt();

if (choice == 1) {
    System.out.println("Het getal " + number + (checkIfOdd(number) ? " is oneven." : " is even."));
} else if (choice == 2) {
    System.out.println("Het getal " + number + (checkIfPowerOfTwo(number) ? " is een macht van 2." : " is geen macht van 2."));
} else if (choice == 3) {
    System.out.println("Het twee-complement van " + number + " is: " + calculateTwosComplement(number));
} else if (choice == 4) {
    scanner.nextLine();
    System.out.print("Voer een geldig IP-adres in (bijvoorbeeld 192.168.1.10): ");
    String ip = scanner.nextLine();
    System.out.print("Geef een subnetmasker op (bijvoorbeeld 255.255.255.0): ");
    String subnetMask = scanner.nextLine();
    calculateNetworkSegment(ip, subnetMask);
} else if (choice == 5) {
    System.out.println("De applicatie wordt afgesloten. Bedankt voor het gebruik!");
    running = false;
} else {
```

```
private static void calculateNetworkSegment(String ipAddress, String subnetMask) { 1 usage
    int ipDecimal = convertToDecimal(ipAddress);
    int maskDecimal = convertToDecimal(subnetMask);
    int networkAddress = ipDecimal & maskDecimal;

    System.out.println("Netwerkadres (decimaal): " + convertToDottedDecimal(networkAddress));
    System.out.println("Netwerkadres (binair): " + String.format("%32s", Integer.toBinaryString(networkAddress)).replace(' ', '0'));
}

private static int convertToDecimal(String address) { 2 usages
    String[] octets = address.split(regex: "\\.");
    int result = 0;
    for (int i = 0; i < octets.length; i++) {
        result |= (Integer.parseInt(octets[i]) << (24 - (i * 8)));
    }
    return result;
}

private static String convertToDottedDecimal(int decimalAddress) { 1 usage
    return ((decimalAddress >> 24) & 0xFF) + "." +
        ((decimalAddress >> 16) & 0xFF) + "." +
        ((decimalAddress >> 8) & 0xFF) + "." +
        (decimalAddress & 0xFF);
}
```

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