

Template Week 6 – Networking

Student number:

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

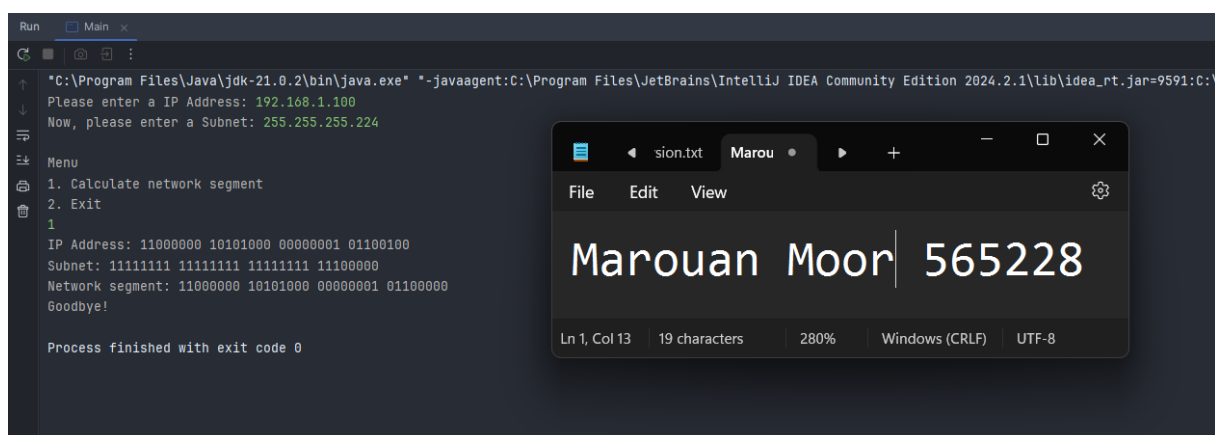
Network Addr: 11000000.10101000.00000001.01100000

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.



```

import java.util.Scanner;
import java.util.concurrent.ConcurrentNavigableMap;

class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Please enter a IP Address: ");
        String ipAddress = scanner.nextLine();
        System.out.print("Now, please enter a Subnet: ");
        String subnet = scanner.nextLine();
        String networkSegment = calculateNetworkSegment(ipAddress, subnet);

        while (true) {

            System.out.println("\nMenu");
            System.out.println("1. Calculate network segment");
            System.out.println("2. Exit");

            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.println("IP Address: " +
convertToBinary(String.valueOf(ipAddress)));
                    System.out.println("Subnet: " + convertToBinary(String.valueOf(subnet)));
                    System.out.println("Network segment: " + networkSegment);

                case 2:
                    System.out.println("Goodbye!");
                    scanner.close();
                    return;

            }

        }

        public static String convertToBinary(String address) {
            String[] octets = address.split("\\.");
            StringBuilder binary = new StringBuilder();

            for (String octet : octets){
                int value = Integer.parseInt(octet);
                String binaryOctet = String.format("%08d",
Integer.parseInt(Integer.toBinaryString(value)));
                binary.append(binaryOctet).append(" ");
            }
            return binary.toString();
        }

        public static String calculateNetworkSegment(String ipAddress, String subnet){

            String[] iPOctets = ipAddress.split("\\.");
            String[] subnetOctets = subnet.split("\\.");
            StringBuilder NetworkSegment = new StringBuilder();

            for (int i = 0; i < 4; i++) {
                int ip = Integer.parseInt(iPOctets[i]);
                int subnetMask = Integer.parseInt(subnetOctets[i]);
                int network = ip & subnetMask;
            }
        }
    }
}

```

```
        String binaryOctet = String.format("%08d",
Integer.parseInt(Integer.toBinaryString(network)));
        NetworkSegment.append(binaryOctet).append(" ");

    }
    return NetworkSegment.toString();
}
}
```

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