

PRACTICAL NO. 1

Aim: Write a program for implementing a Client Server communication model using TCP.

Practical 1A: A client server based program using TCP to find if the number entered is prime.

Server.java:

```
import java.net.*;
import java.io.*;

public class Server {
    Run | Debug
    public static void main(String[] args) throws Exception {
        ServerSocket ss = new ServerSocket(port: 5000);
        Socket s = ss.accept();
        BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));
        PrintWriter pw = new PrintWriter(s.getOutputStream(), autoFlush: true);

        int n = Integer.parseInt(br.readLine());
        boolean prime = true;

        if (n <= 1) prime = false;
        for (int i = 2; i <= n/2; i++)
            if (n % i == 0) prime = false;

        pw.println(prime ? "Prime Number" : "Not Prime");

        s.close();
        ss.close();
    }
}
```

Client.java:

```
import java.net.*;
import java.io.*;

public class Client {
    Run | Debug
    public static void main(String[] args) throws Exception {
        Socket s = new Socket(host: "localhost", port: 5000);
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        PrintWriter pw = new PrintWriter(s.getOutputStream(), autoFlush: true);
        BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));

        System.out.print(s: "Enter a number: ");
        pw.println(br.readLine());

        System.out.println("Result: " + in.readLine());

        s.close();
    }
}
```

Output:

Compile first:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
● PS C:\Users\HP\Documents\ACC_Prac\Prac1> javac Server.java  
● PS C:\Users\HP\Documents\ACC_Prac\Prac1> javac Client.java  
◆ PS C:\Users\HP\Documents\ACC_Prac\Prac1>
```

Run both in different terminal:

```
◆ PS C:\Users\HP\Documents\ACC_Prac\Prac1> java Server  
Server Started.....  
● PS C:\Users\HP\Documents\ACC_Prac\Prac1> java Client  
Enter a number: 20  
Result: Not Prime  
◆ PS C:\Users\HP\Documents\ACC_Prac\Prac1>
```

Practical 1B: A client server TCP based chatting application.

ChatServer.java:

```
import java.net.*;  
import java.io.*;  
  
public class ChatServer {  
    Run | Debug  
    public static void main(String[] args) throws Exception {  
        ServerSocket ss = new ServerSocket(port: 5000);  
        Socket s = ss.accept();  
  
        BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));  
        PrintWriter out = new PrintWriter(s.getOutputStream(), autoFlush: true);  
        BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));  
  
        while (true) {  
            System.out.print(s: "Server: ");  
            String send = kb.readLine();  
            out.println(send); // send to client  
  
            String recv = in.readLine(); // receive from client  
            System.out.println("Client: " + recv);  
  
            if (send.equals(anObject: "exit") || recv.equals(anObject: "exit"))  
                break;  
        }  
  
        s.close();  
        ss.close();  
    }
```

ChatClient.java:

```
import java.net.*;
import java.io.*;

public class ChatClient {
    Run | Debug
    public static void main(String[] args) throws Exception {
        Socket s = new Socket(host: "localhost", port: 5000);

        BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));
        PrintWriter out = new PrintWriter(s.getOutputStream(), autoFlush: true);
        BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));

        while (true) {
            String recv = in.readLine(); // receive from server
            System.out.println("Server: " + recv);

            System.out.print(s: "Client: ");
            String send = kb.readLine();
            out.println(send); // send to server

            if (send.equals(anObject: "exit") || recv.equals(anObject: "exit"))
                break;
        }

        s.close();
    }
}
```

Output:

Compile it:

```
PS C:\Users\HP\Documents\ACC_Prac\Prac1> javac ChatServer.java
PS C:\Users\HP\Documents\ACC_Prac\Prac1> javac ChatClient.java
```

Run:

```
● PS C:\Users\HP\Documents\ACC_Prac\Prac1> java ChatServer
    Server: Hello
    Client: Hello Server
    Server: How Are you?
    Client: I am fine! How r u?
    Server: exit
    ↴ Client: exit
    PS C:\Users\HP\Documents\ACC_Prac\Prac1> □
```

```
PS C:\Users\HP\Documents\ACC_Prac\Prac1> java ChatClient
● Server: Hello
    Client: Hello Server
    Server: How Are you?
    Client: I am fine! How r u?
    Server: exit
    Client: exit
    ↴ PS C:\Users\HP\Documents\ACC_Prac\Prac1> □
```

PRACTICAL NO. 2

Aim: Write a program for implementing a Client Server communication model using UDP.

Practical 2A: A client server based program using UDP to find if the number entered is even or odd.

UDPServer.java:

```
import java.net.*;

public class UDPServer {
    Run | Debug
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket(port: 5000);
        byte[] buf = new byte[100];

        DatagramPacket dp = new DatagramPacket(buf, buf.length);
        ds.receive(dp); // receive number

        String numStr = new String(dp.getData()).trim();
        int n = Integer.parseInt(numStr);

        String result = (n % 2 == 0) ? "Even Number" : "Odd Number";
        byte[] sendData = result.getBytes();

        DatagramPacket send = new DatagramPacket(
            sendData, sendData.length,
            dp.getAddress(), dp.getPort());
    }

    ds.send(send); // send result
    ds.close();
}
```

UDPClient.java:

```
import java.net.*;
import java.io.*;

public class UDPClient {
    Run | Debug
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket();

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print(s: "Enter a number: ");
        String num = br.readLine();

        byte[] data = num.getBytes();
        InetAddress ip = InetAddress.getByName(host: "localhost");

        DatagramPacket dp = new DatagramPacket(data, data.length, ip, port: 5000);
        ds.send(dp); // send number

        byte[] buf = new byte[100];
        DatagramPacket reply = new DatagramPacket(buf, buf.length);
        ds.receive(reply); // receive result

        System.out.println("Result: " + new String(reply.getData()).trim());
        ds.close();
    }
}
```

Output:

```
● PS C:\Users\HP\Documents\ACC_Prac> javac UDPServer.java
● PS C:\Users\HP\Documents\ACC_Prac> javac UDPClient.java
● PS C:\Users\HP\Documents\ACC_Prac> java UDPServer
● PS C:\Users\HP\Documents\ACC_Prac> java UDPServer
❖ PS C:\Users\HP\Documents\ACC_Prac> 
```

```
PS C:\Users\HP\Documents\ACC_Prac> java UDPClient
Enter a number: 5
Result: Odd Number
PS C:\Users\HP\Documents\ACC_Prac> java UDPClient
Enter a number: 2
Result: Even Number
PS C:\Users\HP\Documents\ACC_Prac> 
```

Practical 2B: A client server based program using UDP to find the factorial of the entered number.

UDPServer1.java:

```
import java.net.*;

public class UDPServer1 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket(port: 5000);
        byte[] buf = new byte[100];

        DatagramPacket dp = new DatagramPacket(buf, buf.length);
        ds.receive(dp); // receive number

        String numStr = new String(dp.getData()).trim();
        int n = Integer.parseInt(numStr);

        // factorial logic (simple)
        int fact = 1;
        for (int i = 1; i <= n; i++)
            fact *= i;

        byte[] result = String.valueOf(fact).getBytes();

        DatagramPacket reply = new DatagramPacket(
            result, result.length,
            dp.getAddress(), dp.getPort());

        ds.send(reply); // send factorial
        ds.close();
    }
}
```

UDPClient1.java:

```
import java.net.*;
import java.io.*;

public class UDPClient1 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket();

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter a number: ");
        String num = br.readLine();

        byte[] data = num.getBytes();
        InetAddress ip = InetAddress.getByName(host: "localhost");

        DatagramPacket dp = new DatagramPacket(data, data.length, ip, port: 5000);
        ds.send(dp); // send number

        byte[] buf = new byte[100];
        DatagramPacket reply = new DatagramPacket(buf, buf.length);
        ds.receive(reply); // receive factorial

        System.out.println("Factorial = " + new String(reply.getData()).trim());
        ds.close();
    }
}
```

Run:

```
PS C:\Users\HP\Documents\ACC_Prac> javac UDPServer1.java
PS C:\Users\HP\Documents\ACC_Prac> javac UDPClient1.java
PS C:\Users\HP\Documents\ACC_Prac> java UDPServer1
PS C:\Users\HP\Documents\ACC_Prac>
```

```
PS C:\Users\HP\Documents\ACC_Prac> java UDPClient1
Enter a number: 5
Factorial = 120
PS C:\Users\HP\Documents\ACC_Prac>
```

Practical 2C: A program to implement simple calculator operations like addition, subtraction, multiplication and division.

CalcServer.java:

```
import java.net.*;  
  
public class CalcServer {  
    Run | Debug  
    public static void main(String[] args) throws Exception {  
  
        DatagramSocket ds = new DatagramSocket(port: 5000);  
  
        byte[] buf = new byte[200];  
        DatagramPacket dp = new DatagramPacket(buf, buf.length);  
  
        ds.receive(dp); // receive 2 numbers  
  
        String received = new String(dp.getData()).trim();  
        String[] parts = received.split(regex: " ");  
  
        int n1 = Integer.parseInt(parts[0]);  
        int n2 = Integer.parseInt(parts[1]);  
  
        int add = n1 + n2;  
        int sub = n1 - n2;  
        int mul = n1 * n2;  
        String div = (n2 != 0) ? String.valueOf((double) n1 / n2) : "Undefined";  
  
        String result = "Add = " + add +  
                      " | Sub = " + sub +  
                      " | Mul = " + mul +  
                      " | Div = " + div;  
  
        byte[] sendData = result.getBytes();  
  
        DatagramPacket reply = new DatagramPacket(  
            sendData, sendData.length,  
            dp.getAddress(), dp.getPort());  
  
        ds.send(reply);  
        ds.close();  
    }  
}
```

CalcClient.java

```
import java.net.*;
import java.io.*;

public class CalcClient {
    Run | Debug
    public static void main(String[] args) throws Exception {

        DatagramSocket ds = new DatagramSocket();
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.print(s: "Enter first number: ");
        String n1 = br.readLine();

        System.out.print(s: "Enter second number: ");
        String n2 = br.readLine();

        String message = n1 + " " + n2;
        byte[] data = message.getBytes();

        InetAddress ip = InetAddress.getByName(host: "localhost");

        DatagramPacket dp = new DatagramPacket(data, data.length, ip, port: 5000);
        ds.send(dp); // send two numbers

        byte[] buf = new byte[200];
        DatagramPacket reply = new DatagramPacket(buf, buf.length);
        ds.receive(reply);

        System.out.println("\nResults:\n" + new String(reply.getData()).trim());

        ds.close();
    }
}
```

Run:

```
PS C:\Users\HP\Documents\ACC_Prac> javac CalcServer.java
PS C:\Users\HP\Documents\ACC_Prac> javac CalcClient.java
PS C:\Users\HP\Documents\ACC_Prac> java CalcServer
PS C:\Users\HP\Documents\ACC_Prac> java CalcServer
PS C:\Users\HP\Documents\ACC_Prac>
```

- PS C:\Users\HP\Documents\ACC_Prac> java CalcClient

```
Enter first number: 2
Enter second number: 4
```

Results:

Add = 6 | Sub = -2 | Mul = 8 | Div = 0.5

- ❖ PS C:\Users\HP\Documents\ACC_Prac>

Practical 2D: A program that finds the square, square root, cube and cube root of the entered number.

MathServer:

```
import java.net.*;  
  
public class MathServer {  
    Run | Debug  
    public static void main(String[] args) throws Exception {  
        DatagramSocket ds = new DatagramSocket(port: 6000);  
        byte[] buf = new byte[100];  
  
        DatagramPacket dp = new DatagramPacket(buf, buf.length);  
        ds.receive(dp); // receive number  
  
        String numStr = new String(dp.getData()).trim();  
        double n = Double.parseDouble(numStr);  
  
        double sq = n * n;  
        double sqrt = Math.sqrt(n);  
        double cube = n * n * n;  
        double cbr = Math.cbrt(n);  
  
        String result = "Square=" + sq +  
                      " Sqrt=" + sqrt +  
                      " Cube=" + cube +  
                      " CubeRoot=" + cbr;  
  
        byte[] sendData = result.getBytes();  
  
        DatagramPacket reply = new DatagramPacket(  
            sendData, sendData.length,  
            dp.getAddress(), dp.getPort());  
  
        ds.send(reply); // send all results  
        ds.close();  
    }  
}
```

MathClient:

```
import java.net.*;
import java.io.*;

public class MathClient {
    Run | Debug
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket();
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.print(s: "Enter a number: ");
        String num = br.readLine();

        byte[] data = num.getBytes();
        InetAddress ip = InetAddress.getByName(host: "localhost");

        DatagramPacket dp = new DatagramPacket(data, data.length, ip, port: 6000);
        ds.send(dp); // send number

        byte[] buf = new byte[200];
        DatagramPacket reply = new DatagramPacket(buf, buf.length);
        ds.receive(reply); // receive results

        System.out.println("Results: " + new String(reply.getData()).trim());
        ds.close();
    }
}
```

Run:

```
PS C:\Users\HP\Documents\ACC_Prac> javac MathServer.java
● PS C:\Users\HP\Documents\ACC_Prac> javac MathClient.java
● PS C:\Users\HP\Documents\ACC_Prac> java MathServer
❖ PS C:\Users\HP\Documents\ACC_Prac> 
```

```
● PS C:\Users\HP\Documents\ACC_Prac> java MathClient
Enter a number: 4
Results: Square=16.0 Sqrt=2.0 Cube=64.0 CubeRoot=1.5874010519681996
❖ PS C:\Users\HP\Documents\ACC_Prac> 
```

PRACTICAL NO. 3

Aim: A multicast Socket example.

MulticastServer:

```
import java.net.*;  
  
public class MulticastServer {  
    Run | Debug  
    public static void main(String[] args) throws Exception {  
  
        DatagramSocket ds = new DatagramSocket();  
        String msg = "Hello from Multicast Server";  
  
        InetAddress group = InetAddress.getByName(host: "224.0.0.5"); // multicast group  
        byte[] data = msg.getBytes();  
  
        DatagramPacket dp = new DatagramPacket(data, data.length, group, port: 5000);  
        ds.send(dp);  
  
        System.out.println(x: "Message sent to multicast group.");  
        ds.close();  
    }  
}
```

MulticastClient:

```
import java.net.*;  
  
public class MulticastClient {  
    Run | Debug  
    public static void main(String[] args) throws Exception {  
  
        InetAddress group = InetAddress.getByName(host: "224.0.0.5");  
        NetworkInterface ni = NetworkInterface.getByInetAddress(InetAddress.getLocalHost());  
  
        MulticastSocket ms = new MulticastSocket(port: 5000);  
        ms.joinGroup(new InetSocketAddress(group, port: 5000), ni);  
  
        byte[] buf = new byte[200];  
        DatagramPacket dp = new DatagramPacket(buf, buf.length);  
  
        System.out.println(x: "Waiting for multicast message...");  
        ms.receive(dp);  
  
        System.out.println("Received: " + new String(dp.getData()).trim());  
  
        ms.leaveGroup(new InetSocketAddress(group, port: 5000), ni);  
        ms.close();  
    }  
}
```

Run Client first:

```
● PS C:\Users\HP\Documents\ACC_Prac> javac MulticastServer.java
● PS C:\Users\HP\Documents\ACC_Prac> javac MulticastClient.java
● PS C:\Users\HP\Documents\ACC_Prac> java MulticastClient
    Waiting for multicast message...
    Received: Hello from Multicast Server
◆◆ PS C:\Users\HP\Documents\ACC_Prac> █
```



```
● PS C:\Users\HP\Documents\ACC_Prac> java MulticastServer
    Message sent to multicast group.
◆◆ PS C:\Users\HP\Documents\ACC_Prac> █
```

PRACTICAL NO. 4

Aim: Write a program to show the object communication using RMI

Practical 4A: A RMI based application program to display current date and time.

DateTimeInterface.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface DateTimeInterface extends Remote {
    String getDateTime() throws RemoteException;
}
```

DateTimeServer.java

```
import java.rmi.server.UnicastRemoteObject;
import java.rmi.RemoteException;
import java.rmi.Naming;
import java.util.Date;

public class DateTimeServer extends UnicastRemoteObject implements DateTimeInterface {

    protected DateTimeServer() throws RemoteException {
        super();
    }

    @Override
    public String getDateTime() throws RemoteException {
        Date d = new Date();
        return d.toString();
    }

    Run | Debug
    public static void main(String[] args) {
        try {
            DateTimeServer obj = new DateTimeServer();
            Naming.rebind(name: "rmi://localhost:1099/DateTimeService", obj);
            System.out.println("RMI Server is running...");
        } catch (Exception e) {
            System.out.println("Server Error: " + e.getMessage());
        }
    }
}
```

DateTimeClient.java

```
import java.rmi.Naming;

public class DateTimeClient {
    Run | Debug
    public static void main(String[] args) {
        try {
            DateTimeInterface obj = (DateTimeInterface) Naming.lookup(name: "rmi://localhost:1099/DateTimeService");
            System.out.println("Current Date and Time: " + obj.getDateTime());
        } catch (Exception e) {
            System.out.println("Client Error: " + e.getMessage());
        }
    }
}
```

Run:

- PS C:\Users\HP\Documents\ACC_Prac> **javac** DateTimeInterface.java
- PS C:\Users\HP\Documents\ACC_Prac> **javac** DateTimeServer.java
- PS C:\Users\HP\Documents\ACC_Prac> **javac** DateTimeClient.java
- ❖ PS C:\Users\HP\Documents\ACC_Prac> **rmiregistry**

```
PS C:\Users\HP\Documents\ACC_Prac> java DateTimeServer
RMI Server is running...
```

```
PS C:\Users\HP\Documents\ACC_Prac> java DateTimeClient
● Current Date and Time: Thu Dec 04 21:42:25 IST 2025
● PS C:\Users\HP\Documents\ACC_Prac> java DateTimeClient
    Current Date and Time: Thu Dec 04 21:42:59 IST 2025
❖ PS C:\Users\HP\Documents\ACC_Prac>
```

Practical 4B: A RMI based application program that converts digits to words, e.g. 123 will be converted to one two three.

DigitToWordInterface.java:

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface DigitToWordInterface extends Remote {
    String convertToWords(String number) throws RemoteException;
}
```

DigitToWordServer.java:

```
import java.rmi.server.UnicastRemoteObject;
import java.rmi.Naming;
import java.rmi.RemoteException;

public class DigitToWordServer extends UnicastRemoteObject implements DigitToWordInterface {

    protected DigitToWordServer() throws RemoteException {
        super();
    }

    @Override
    public String convertToWords(String number) throws RemoteException {
        String[] words = { "zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine" };
        StringBuilder sb = new StringBuilder();

        for (char c : number.toCharArray()) {
            if (Character.isDigit(c)) {
                sb.append(words[c - '0']).append(" ");
            } else {
                sb.append("? ");
            }
        }
        return sb.toString().trim();
    }

    Run | Debug
    public static void main(String[] args) {
        try {
            DigitToWordServer obj = new DigitToWordServer();
            Naming.rebind(name: "rmi://localhost:1099/DigitToWordService", obj);
            System.out.println(x: "DigitToWord RMI Server is running...");
        } catch (Exception e) {
            System.out.println("Server Error: " + e.getMessage());
        }
    }
}
```

DigitToWordClient.java:

```
import java.rmi.Naming;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class DigitToWordClient {
    Run | Debug
    public static void main(String[] args) {
        try {
            DigitToWordInterface obj = (DigitToWordInterface) Naming.lookup(name: "rmi://localhost:1099/DigitToWordService");
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

            System.out.print(s: "Enter a number: ");
            String number = br.readLine();

            String result = obj.convertToWords(number);
            System.out.println("In Words: " + result);

        } catch (Exception e) {
            System.out.println("Client Error: " + e.getMessage());
        }
    }
}
```

Run:

```
PS C:\Users\HP\Documents\ACC_Prac> javac DigitToWordInterface.java
PS C:\Users\HP\Documents\ACC_Prac> javac DigitToWordServer.java
PS C:\Users\HP\Documents\ACC_Prac> javac DigitToWordClient.java
PS C:\Users\HP\Documents\ACC_Prac> rmiregistry
PS C:\Users\HP\Documents\ACC_Prac> java DigitToWordServer
DigitToWord RMI Server is running...
[ ]
```



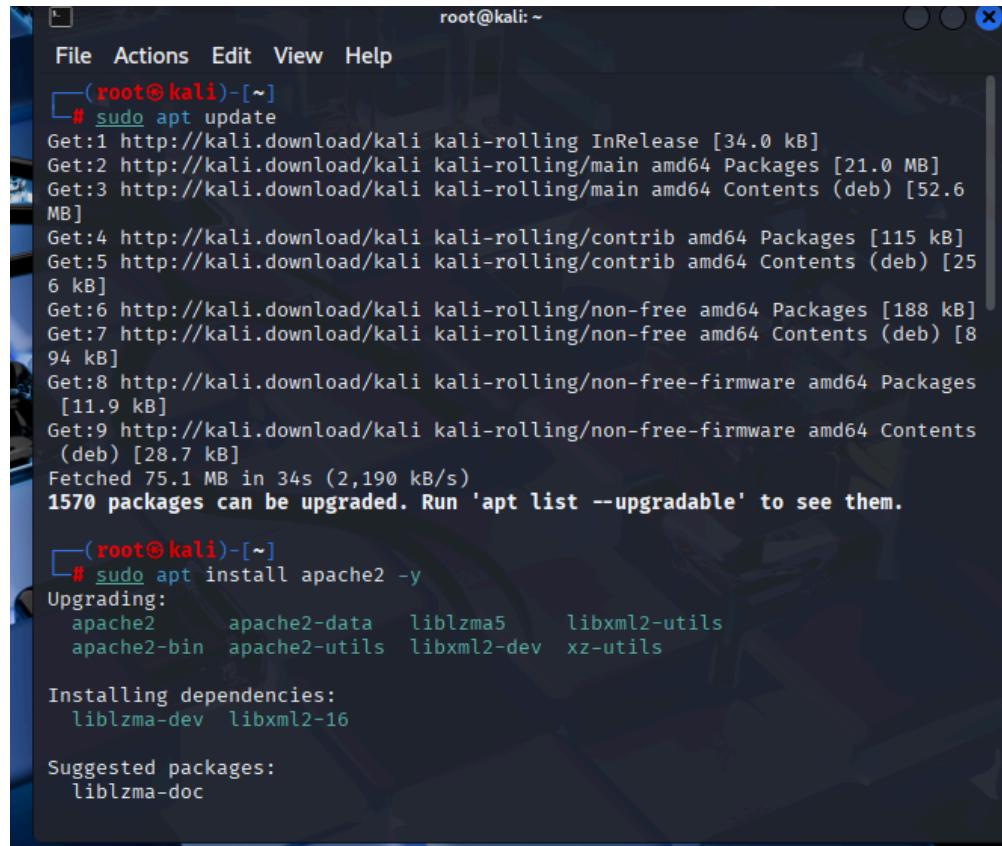
```
PS C:\Users\HP\Documents\ACC_Prac> java DigitToWordClient
Enter a number: 7654
In Words: seven six five four
PS C:\Users\HP\Documents\ACC_Prac> java DigitToWordClient
Enter a number: 4321
In Words: four three two one
PS C:\Users\HP\Documents\ACC_Prac> [ ]
```

PRACTICAL NO. 5

Hosting Multiple Websites on a Single Apache Server using Virtual Hosts

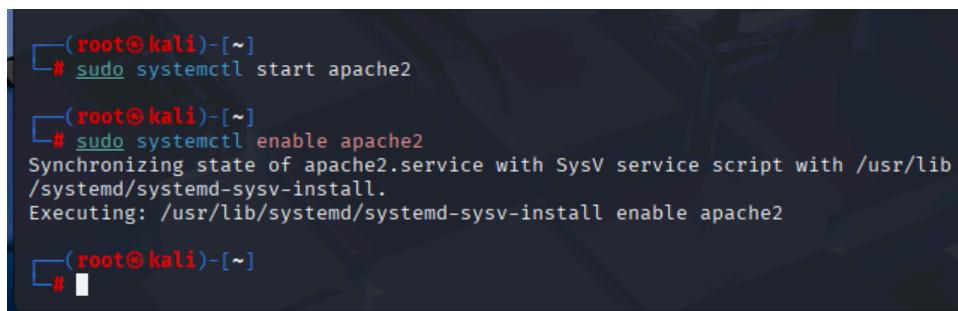
Aim: To configure and host multiple websites on a single Apache server inside a Virtual Machine using Virtual Hosts, and access them from a host machine by mapping domain names to the VM's IP address.

```
sudo apt update  
sudo apt install apache2 -y
```



```
root@kali: ~  
File Actions Edit View Help  
[(root@kali)-[~]  
# sudo apt update  
Get:1 http://kali.download/kali kali-rolling InRelease [34.0 kB]  
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]  
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [52.6  
MB]  
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]  
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [25  
6 kB]  
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [188 kB]  
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [8  
94 kB]  
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages  
[11.9 kB]  
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents  
(deb) [28.7 kB]  
Fetched 75.1 MB in 34s (2,190 kB/s)  
1570 packages can be upgraded. Run 'apt list --upgradable' to see them.  
[(root@kali)-[~]  
# sudo apt install apache2 -y  
Upgrading:  
 apache2      apache2-data    liblzma5      libxml2-utils  
 apache2-bin   apache2-utils  libxml2-dev   xz-utils  
  
Installing dependencies:  
 liblzma-dev  libxml2-16  
  
Suggested packages:  
 liblzma-doc
```

```
sudo systemctl start apache2  
sudo systemctl enable apache2
```



```
[(root@kali)-[~]  
# sudo systemctl start apache2  
[(root@kali)-[~]  
# sudo systemctl enable apache2  
Synchronizing state of apache2.service with SysV service script with /usr/lib  
/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2  
[(root@kali)-[~]  
# ]
```

```
sudo mkdir -p /var/www/example.com
```

```
sudo mkdir -p /var/www/test.com
```

Add simple HTML files:

```
echo "<h1>Hello from example.com</h1>" | sudo tee /var/www/example.com/index.html
```

```
echo "<h1>Hello from test.com</h1>" | sudo tee /var/www/test.com/index.html
```



```
(root㉿kali)-[~]
# sudo mkdir -p /var/www/example.com
(root㉿kali)-[~]
# sudo mkdir -p /var/www/test.com
(root㉿kali)-[~]
# echo "<h1>Hello from example.com</h1>" | sudo tee /var/www/example.com/index.html
<h1>Hello from example.com</h1>
(root㉿kali)-[~]
# echo "<h1>Hello from test.com</h1>" | sudo tee /var/www/test.com/index.html
<h1>Hello from test.com</h1>
(root㉿kali)-[~]
```

```
sudo chown -R www-data:www-data /var/www/example.com
```

```
sudo chown -R www-data:www-data /var/www/test.com
```

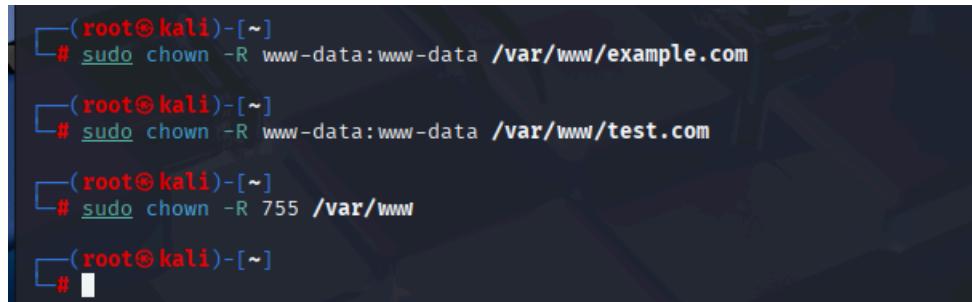
```
sudo chmod -R 755 /var/www
```

chown = **change ownership**

-R = apply changes **recursively** (to all files and folders inside)

www-data:www-data = set the **owner and group** to Apache's default user (www-data)

/var/www/example.com = the folder you are changing



```
(root㉿kali)-[~]
# sudo chown -R www-data:www-data /var/www/example.com
(root㉿kali)-[~]
# sudo chown -R www-data:www-data /var/www/test.com
(root㉿kali)-[~]
# sudo chown -R 755 /var/www
(root㉿kali)-[~]
```

```
sudo nano /etc/apache2/sites-available/example.com.conf

<VirtualHost *:80>
    ServerName example.com
    DocumentRoot /var/www/example.com

    <Directory /var/www/example.com>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/example.com-error.log
    CustomLog ${APACHE_LOG_DIR}/example.com-access.log combined
</VirtualHost>
```

```
sudo nano /etc/apache2/sites-available/test.com.conf
```

```
<VirtualHost *:80>
    ServerName test.com
    DocumentRoot /var/www/test.com

    <Directory /var/www/test.com>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/test.com-error.log
    CustomLog ${APACHE_LOG_DIR}/test.com-access.log combined
</VirtualHost>
```

```
sudo a2ensite example.com.conf
sudo a2ensite test.com.conf
sudo systemctl reload apache2
```

Check config:

```
sudo apache2ctl configtest
```

```
(root㉿kali)-[~]
└─# sudo nano /etc/apache2/sites-available/000-default.conf

(root㉿kali)-[~]
└─# sudo nano /etc/apache2/sites-available/example.com.conf

(root㉿kali)-[~]
└─# sudo nano /etc/apache2/sites-available/test.com.conf

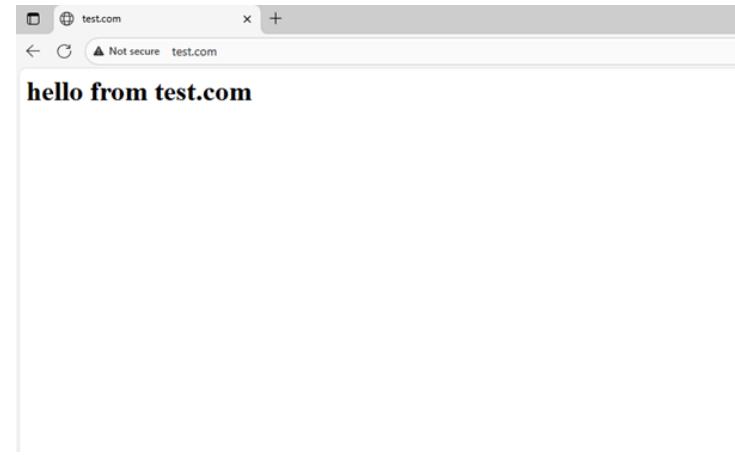
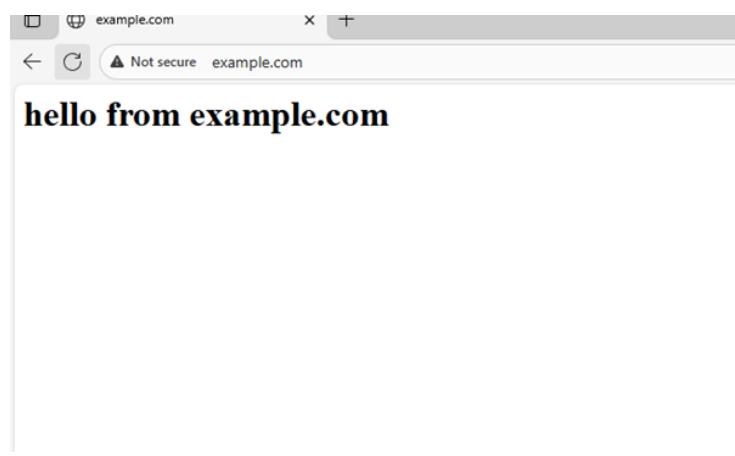
(root㉿kali)-[~]
└─# a2ensite example.com.conf
Site example.com already enabled

(root㉿kali)-[~]
└─# a2ensite test.com.conf
Site test.com already enabled

(root㉿kali)-[~]
└─# sudo systemctl reload apache2

(root㉿kali)-[~]
└─# sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK

(root㉿kali)-[~]
└─#
```



PRACTICAL NO. 6

SSH Key Login

On a cloud server (like AWS EC2), if anyone knows the password, they can login and hack the server.

So we do 3 things:

1. Create a new admin user (instead of using root)
2. Use SSH Key Login (so no one can hack using passwords)
3. Disable Root Login (so no one can target root)

```
sudo adduser adminuser6
```

```
sudo usermod -aG sudo adminuser6
```

```
(root@kali)-[~]
# sudo adduser adminuser6
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for adminuser6
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y

(root@kali)-[~]
# sudo usermod -aG sudo adminuser6

(root@kali)-[~]
#
```

Now in MobaXterm,

```
ssh adminuser6@192.168.0.106
```

```
ssh-keygen
```

```
2025-12-04 21:02:45 /home/mobaxterm ssh adminuser6@192.168.0.106
adminuser6@192.168.0.106's password:
Linux kali 6.12.25-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.25-1kali1 (2025-04-30)
x86_64

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

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  4 10:30:28 2025 from 192.168.0.103
```

```
[~] (adminuser6㉿ kali) ~
$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/adminuser6/.ssh/id_ed25519):
/home/adminuser6/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase for "/home/adminuser6/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/adminuser6/.ssh/id_ed25519
Your public key has been saved in /home/adminuser6/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:IQJPWbioLseyswDWerMucQNeg/9/vgLgp4MnDOPqYYY adminuser6@kali
The key's randomart image is:
+--[ED25519 256]--+
|   . +.
|   +o
|   ..o...
| oo+... .
| oo=.o S
| Bo.= o
| EB+=o .
| BB+=o. ..
| BB=o. ..+o.
+---[SHA256]-----+
```

Ssh-copy-id adminuser6@192.168.0.106

```
[~] (adminuser6㉿ kali) ~
$ ssh-copy-id adminuser6@192.168.0.106
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/adminuser6/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
adminuser6@192.168.0.106's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'adminuser6@192.168.0.106'"
and check to make sure that only the key(s) you wanted were added.
```

ssh adminuser6@192.168.0.106

```
[~] (adminuser6㉿ kali) ~
$ ssh adminuser6@192.168.0.106
Enter passphrase for key '/home/adminuser6/.ssh/id_ed25519':
Linux kali 6.12.25-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.25-1kali1 (2025-04-30)
x86_64

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

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  4 10:34:06 2025 from 192.168.0.103
[~] (adminuser6㉿ kali) ~
$
```

On your server,

open the SSH config:

```
sudo nano /etc/ssh/sshd_config
```

Find or add these lines (add if missing):

```
PermitRootLogin no
```

```
PasswordAuthentication no
```

Save and exit:

- Press CTRL + O then Enter to save

- Press CTRL + X to exit nano

Restart SSH service to apply changes:

```
sudo systemctl restart ssh
```

The image shows a terminal window with a dark background and light-colored text. It displays three lines of command-line input from a root shell on a Kali Linux system. The first line is the command to edit the SSH configuration file. The second line is the command to restart the SSH service. The third line is a blank line, indicating the end of the session.

```
[root@kali)-[~]
# sudo nano /etc/ssh/sshd_config

[root@kali)-[~]
# sudo systemctl restart ssh

[root@kali)-[~]
#
```

PRACTICAL NO. 7

Network & Compliance Hardening for a Linux Server

PART 1 Firewall Configuration Using UFW

```
sudo apt install ufw -y
```

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

```
sudo ufw allow 22/tcp
```

```
sudo ufw allow 443/tcp
```

```
sudo ufw deny 80/tcp
```

```
ubuntu@ubuntu:~$ sudo apt install ufw -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ubuntu:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
ubuntu@ubuntu:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
ubuntu@ubuntu:~$ sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
ubuntu@ubuntu:~$ sudo ufw allow 443/tcp
Rules updated
Rules updated (v6)
ubuntu@ubuntu:~$ sudo ufw deny 80/tcp
Rules updated
Rules updated (v6)
```

```
sudo ufw enable
```

```
sudo ufw status
```

```
ubuntu@ubuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
ubuntu@ubuntu:~$ sudo ufw status
Status: active

To                         Action      From
--                         --         --
22/tcp                      ALLOW       Anywhere
443/tcp                     ALLOW       Anywhere
80/tcp                      DENY        Anywhere
22/tcp (v6)                 ALLOW       Anywhere (v6)
443/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                 DENY        Anywhere (v6)
```

PART 2 Compliance Logging Using auditd

sudo apt install auditd -y

```
vboxuser@Ubuntu:~$ sudo apt install auditd -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libauparse0t64
Suggested packages:
  audispd-plugins
The following NEW packages will be installed:
  auditd libauparse0t64
0 upgraded, 2 newly installed, 0 to remove and 113 not upgraded.
Need to get 274 kB of archives.
After this operation, 893 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 libauparse0t64 amd64 1:3.1.2-2.1build1.1 [58.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 auditd amd64 1:3.1.2-2.1build1.1 [215 kB]
Fetched 274 kB in 1s (188 kB/s)
Selecting previously unselected package libauparse0t64:amd64.
(Reading database ... 150416 files and directories currently installed.)
Preconfiguring packages ...
Setting up libauparse0t64:amd64 (1:3.1.2-2.1build1.1) ...
Setting up auditd (1:3.1.2-2.1build1.1) ...
Processing triggers for man-db (2.8.3-2.1ubuntu0.1) ...
```

sudo systemctl start auditd

```
vboxuser@Ubuntu:~$ sudo auditctl -w /etc/passwd -p wa -k passwd_changes
vboxuser@Ubuntu:~$ sudo ausearch -k passwd_changes
-----
time->Thu Dec  4 17:05:31 2025
type=PROCTITLE msg=audit(1764867931.921:200): proctitle=617564697463746C002D7700
2F6574632F706173737764002D70007761002D6B007061737377645F6368616E676573
type=SYSCALL msg=audit(1764867931.921:200): arch=c000003e syscall=44 success=yes
  exit=1084 a0=4 a1=7ffe8e684930 a2=43c a3=0 items=0 ppid=3524 pid=3525 auid=1000
  uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=3 comm="au
ditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1764867931.921:200): auid=1000 ses=3 subj=unconfine
d op=add rule key="passwd changes" list=4 res=1
```

sudo auditctl -w /etc/passwd -p wa -k passwd_changes
sudo ausearch -k passwd_changes

sudo adduser testuser3

sudo ausearch -k passwd_changes

```
vboxuser@Ubuntu:~$ sudo adduser testuser3
info: Adding user `testuser3' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `testuser3' (1003) ...
info: Adding new user `testuser3' (1003) with group `testuser3 (1003)' ...
info: Creating home directory `/home/testuser3' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for testuser3
Enter the new value, or press ENTER for the default
      Full Name []:
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []
Is the information correct? [Y/n] y
info: Adding new user `testuser3' to supplemental / extra groups `users' ...
info: Adding user `testuser3' to group `users' ...
```

```
vboxuser@Ubuntu:~$ sudo ausearch -k passwd_changes
-----
time->Thu Dec  4 17:05:31 2025
type=PROCTITLE msg=audit(1764867931.921:200): proctitle=617564697463746C002D770E
2F6574632F706173737764002D70007761002D6B007061737377645F6368616E676573
type=SYSCALL msg=audit(1764867931.921:200): arch=c000003e syscall=44 success=yes
exit=1084 a0=4 a1=7ffe8e684930 a2=43c a3=0 items=0 ppid=3524 pid=3525 auid=1000
uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=3 comm="au
ditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1764867931.921:200): auid=1000 ses=3 subj=unconfine
d op=add_rule key="passwd_changes" list=4 res=1
-----
time->Thu Dec  4 17:07:52 2025
type=PROCTITLE msg=audit(1764868072.292:220): proctitle=2F7362696E2F757365726164
64002D64002F686F6D652F746573747573657233002D670031303033002D73002F62696E2F626173
68002D75003130303300746573747573657233
type=PATH msg=audit(1764868072.292:220): item=0 name="/etc/passwd" inode=1313512
dev=08:02 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi
=0 cap_fn=0 cap_fver=0 cap_frootid=0
```

Sudo aureport -f -i

```
vboxuser@Ubuntu:~$ sudo aureport -f -i

File Report
=====
# date time file syscall success exe auid event
=====
1. 12/04/2025 17:07:52 /etc/passwd openat yes /usr/sbin/useradd vboxuser 220
2. 12/04/2025 17:07:52 /etc/ rename yes /usr/sbin/useradd vboxuser 222
3. 12/04/2025 17:08:03 /etc/passwd openat yes /usr/bin/chfn vboxuser 226
4. 12/04/2025 17:08:03 /etc/ rename yes /usr/bin/chfn vboxuser 227
```

```
sudo nano /etc/security/limits.conf
```

Step 2: Add These Lines at Bottom

```
* hard nproc 200  
* soft nofile 100
```

```
ulimit -a
```

```
vboxuser@Ubuntu:~$ sudo nano /etc/security/limits.conf  
vboxuser@Ubuntu:~$ ulimit -a  
real-time non-blocking time (microseconds, -R) unlimited  
core file size (blocks, -c) 0  
data seg size (kbytes, -d) unlimited  
scheduling priority (-e) 0  
file size (blocks, -f) unlimited  
pending signals (-i) 7556  
max locked memory (kbytes, -l) 251988  
max memory size (kbytes, -m) unlimited  
open files (-n) 1024  
pipe size (512 bytes, -p) 8  
POSIX message queues (bytes, -q) 819200  
real-time priority (-r) 0  
stack size (kbytes, -s) 8192  
cpu time (seconds, -t) unlimited  
max user processes (-u) 7556  
virtual memory (kbytes, -v) unlimited  
file locks (-x) unlimited  
vboxuser@Ubuntu:~$
```

PART 3 Checking Open Network Ports (Monitoring Active Services)

```
sudo ss -tuln
```

```
vboxuser@Ubuntu:~$ sudo ss -tuln  
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process  
udp UNCONN 0 0 127.0.0.54:53 0.0.0.0:*  
udp UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:*  
udp UNCONN 0 0 0.0.0.0:41169 0.0.0.0:*  
udp UNCONN 0 0 0.0.0.0:5353 0.0.0.0:*  
udp UNCONN 0 0 [::]:54166 [::]:*  
udp UNCONN 0 0 [::]:5353 [::]:*  
tcp LISTEN 0 4096 127.0.0.1:631 0.0.0.0:*  
tcp LISTEN 0 4096 127.0.0.54:53 0.0.0.0:*  
tcp LISTEN 0 4096 127.0.0.53%lo:53 0.0.0.0:*  
tcp LISTEN 0 4096 [::1]:631 [::]:*
```

PART 4 Enable Automatic Security Updates

```
sudo apt install unattended-upgrades -y
```

```
sudo dpkg-reconfigure unattended-upgrades
```

Select Yes when prompted.

```
vboxuser@Ubuntu:~$ sudo apt install unattended-upgrades -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unattended-upgrades is already the newest version (2.9.1+nmu4ubuntu1).
unattended-upgrades set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 113 not upgraded.
vboxuser@Ubuntu:~$ sudo dpkg-reconfigure unattended-upgrades
vboxuser@Ubuntu:~$
```

PART 5 Disable Unnecessary Services

```
systemctl list-unit-files --type=service
```

```
vboxuser@Ubuntu:~$ systemctl list-unit-files --type=service
UNIT FILE                                     STATE   PRESET
accounts-daemon.service                      enabled enabled
alsa-restore.service                         static  -
alsa-state.service                           static  -
alsa-utils.service                          masked  enabled
anacron.service                            enabled enabled
apparmor.service                           enabled enabled
apport-autoreport.service                   static  -
apport-coredump-hook@.service                static  -
apport-forward@.service                     static  -
apport.service                             enabled enabled
apt-daily-upgrade.service                  static  -
apt-daily.service                          static  -
apt-news.service                          static  -
auditd.service                            enabled enabled
autovt@.service                           alias   -
```

Step 2: Stop an Unused Service (Example: Bluetooth)

```
sudo systemctl stop Bluetooth
```

Step 3: Disable the Service Permanently

```
sudo systemctl disable bluetooth
(Service will not start after reboot)
```