# Konzept für den Programmcode

Ziel: Unser Ziel ist es, ein Python-Programm zu entwickeln, mit dem sich ein virtueller Linux Ubuntu-Server kommunizieren lässt.

Voraussetzungen: Ein Linux Ubuntu-Server soll vorhanden sein. Der Server muss über eine Netzwerkverbindung mit dem Client (dem Computer, auf dem das Python-Programm läuft) verbunden sein. Eine SSH-Verbindung zwischen dem Client und dem Server ist erforderlich. Python-Programmiersprache muss installiert sein.

Funktionalität: Der Benutzer muss in der Lage sein, über das Python-Programm Befehle an den Server zu senden und mit diesem zu kommunizieren. Das Programm sollte in der Lage sein, Dateien auf dem Server zu lesen, zu schreiben und zu bearbeiten. Die Kommunikation zwischen dem Client und dem Server muss über SSH erfolgen. Das Programm sollte auch in der Lage sein, verschiedene Arten von Daten zu empfangen und zu senden.

Testen: Die Funktionalität des Programms muss auf dem Diplomarbeits-Server getestet werden. Der Server muss die oben genannten Voraussetzungen erfüllen. Eine Reihe von Testszenarien muss erstellt werden, um das Programm zu testen. Die Ergebnisse der Tests müssen überprüft und analysiert werden, um sicherzustellen, dass das Programm wie erwartet funktioniert.

# Programmcode Funktion

Dieser Programmcode erstellt eine Verbindung zu einem Server mit der IP-Adresse "127.0.0.1" und dem Port 23456. Es wird eine Nachricht über "Hallo hier ist Patrick Edelmann von der HTL-Anichstraße 5AHEL" versendet und die Antwort des Servers über die angegebenen 1024 Bytes empfangen. Die Nachricht und die Antwort werden in cp850 codiert und dekodiert. Schlussendlich wird die Verbindung geschlossen.

# Programmcode als Screenshot

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung Screenshot des Client Programmes

# Programmcode als Text

#Autor: Edelmann Patrick

#Datum der Finalisierung: 29.12.2022

#Klasse:5AHEL

# Dieses Programm erstellt einen Socket und versucht, eine Verbindung zu einem Server mit der IP-Adresse 127.0.0.1 und dem Port 23456 herzustellen.

#Wenn die Verbindung erfolgreich ist, sendet es eine Nachricht mit dem Namen des Absenders an den Server, der dann eine Antwort zurückgibt.

#Anschließend wird die Verbindung geschlossen.

import socket # importiere das socket modul

host = "127.0.0.1"

port = 23456

s=socket.create\_connection(("localhost",port))

s.send("Hallo hier ist Patrick Edelmann von der HTL-Anichstraße 5AHEL".encode("cp850")) # schreiben der Daten die an Server geschickt werden sollen

bytes=s.recv(1024) # Daten die empangen werden

print("Response from server:",bytes.decode("cp850")) #empfangene Daten dekodieren und ausgeben

s.close() #Verbindung wird geschlossen wenn die Übertragung dieser Nachricht beendet ist

# Programm Ausgabe

Es wurde eine ncat.exe Datei verwendet. Die Datei per Drag and Drop auf den Virtuellen Server hineinscheiben. Danach hinter dem Link -v-l Port Nummer hinschreiben und auf Enter drücken. Jetzt hört der Server auf diesen Port, bis das Programm gestartet wird. Mehr dazu bei der Funktion des Codes.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung Server Ausgabe von dem Programmcode