

Projekt:Netzwerkporgrammierung 2017

Sören Giller

July 24, 2017

Contents

| | | |
|----------|-----------------------------------|----------|
| 1 | Version | 2 |
| 2 | Imports | 2 |
| 2.1 | Server | 2 |
| 3 | Vorbereitung für den Start | 2 |
| 4 | Beispielablauf | 2 |
| 5 | Flask Interface | 3 |
| 5.1 | Verbundene CLienten | 3 |
| 5.2 | Updatehistorie | 3 |
| 5.3 | verfügbare Updates | 4 |
| 5.4 | Downloads | 4 |
| 6 | Update | 4 |

1 Version

Für die Programmierung wurde Python3 verwendet, es muss also auch mit python3 compiliert werden.

2 Imports

Client

Folgende Pakete werden benötigt um den Clienten auszuführen. Diese werden größtenteils verwendet um an die Hardware-Infos heranzukommen. psutil und cpuinfo müssen über das Terminal mit pip-install installiert werden.

```
import socket, os, psutil, subprocess, platform, time, sys, math, urllib.request, zipfile, cpuinfo
from threading import Thread
from pathlib import Path
```

2.1 Server

Der Server benötigt socket, time, flask und threading als Importe um eine Verbindung zum Sockel aufzubauen und den Server zu starten und zu threaden.

```
from flask import *
import socket, time, os
from threading import Thread
```

3 Vorbereitung für den Start

Bevor man den Server startet, muss man seine eigene Ip-Adresse im Netzwerk herausfinden(entweder über das Terminal mit: ip addr show oder in den Verbindungseinstellungen). Diese muss man nun an in der ersten Zeile code(nach den Importen) der Datei thinClient.py ersetzen.

```
4
5 host = '192.168.0.31' # ip des hostst
```

4 Beispielablauf

Schritt1: Server starten

```
Sorens-MacBook-Air:thinServer sorengiller$ python3 thinServer.py
server gestartet
Warte auf Verbindung eines Clienten
* Running on http://0.0.0.0:12345/ (Press CTRL+C to quit)
```

Nun ist es möglich, über die Hostip und dem Port:12345 auf den Flask server zuzugreifen.

| Ip | Hostname | System | Verbindung | Datum | Prozessor | Ram |
|----|----------|--------|------------|-------|-----------|-----|
|----|----------|--------|------------|-------|-----------|-----|

Schritt2: mit Client verbinden

```
[Sorens-MacBook-Air:thinServer sorengiller$ python3 thinclient.py
```

Verbindet sich ein Client mit dem Server, schickt er automatisch seine Hardware-Informationen und sein installiertes Update an den Server. Dieser überprüft, ob die auf dem Clienten installierte Version aktuell ist. Ist dies nicht der Fall, schickt er einen Link an den Clienten, um das entsprechende Paket upzugraden. Es ist nun möglich sich mehreren Clients gleichzeitig auf den Server zu verbinden. Diese werden dann auch im Flask-Interface angezeigt. Diese Server-Client - Kommunikation lässt sich über das Terminal oder über den Flask-Server beobachten.

5 Flask Interface

5.1 Verbundene Clienten

Hier werden alle verbundenen Clienten mit ihren Hardware-Informationen angezeigt. Alive/notAlive gibt an, ob der Client aktuell mit dem Server verbunden ist.

| Verbundene Clienten! | | | | | | |
|----------------------|---------------------------------|--------|------------|---------------------|--|--------|
| Ip | Hostname | System | Verbindung | Datum | Prozessor | Ram |
| 192.168.0.42 | david-Lenovo-IdeaPad-U530-Touch | linux | alive | 23.07.2017 15:44:09 | Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz | 7.72GB |
| 192.168.0.31 | Sorens-MacBook-Air.local | darwin | notAlive | 23.07.2017 15:33:59 | Intel(R) Core(TM) i5-5250U CPU @ 1.60GHz | 8.0GB |
| 192.168.0.14 | Soeren-pc | linux | alive | 23.07.2017 15:42:09 | Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz | 5.6GB |

5.2 Updatehistorie

Verbindet sich ein Client mit dem Server, wird sein Update, mit welchem er sich verbunden hat, angezeigt. Wird das Paket geupdatet, wird dies ebenfalls in der Updatehistorie vermerkt.

| Updatehistorie | | | | | | |
|----------------|---------------------|-------------|---------------|------------|--------------------|--------|
| Ip | Zeit | Updatenname | Updateversion | Pruefsumme | Link | Befehl |
| 192.168.0.14 | 23.07.2017 17:54:23 | Update 3 | 3.0 | ghi | /downloads/update3 | tanxyz |
| 192.168.0.31 | 23.07.2017 17:55:07 | Update 3 | 3.0 | ghi | /downloads/update3 | tanxyz |
| 192.168.0.14 | 23.07.2017 17:56:00 | Update 1 | 1.0 | abc | /downloads/update1 | tanxyz |
| 192.168.0.14 | 23.07.2017 17:56:00 | Update 3 | 3.0 | ghi | /downloads/update3 | tanxyz |




5.3 verfügbare Updates

Hier werden alle möglichen Informationen aller Updates angezeigt.

| Verbundene Clienten! Updatehistorie Verfügbare Updates Downloads | | | | | |
|--|------------|---------------|------------|--------------------------------------|--------|
| Ip | Updatename | Updateversion | Pruefsumme | Link | Befehl |
| 192.168.0.42 | Update 3 | 3.0 | ghi | 192.168.0.31:12345/downloads/update3 | tarxyz |
| 192.168.0.31 | Update 3 | 3.0 | ghi | 192.168.0.31:12345/downloads/update3 | tarxyz |
| 192.168.0.14 | Update 3 | 3.0 | ghi | 192.168.0.31:12345/downloads/update3 | tarxyz |
| 192.168.0.14 | Update 3 | 3.0 | ghi | 192.168.0.31:12345/downloads/update3 | tarxyz |
| 192.168.0.42 | Update 3 | 3.0 | ghi | 192.168.0.31:12345/downloads/update3 | tarxyz |

5.4 Downloads

Unter Downloads befinden sich ein paar Informationen zum Download sowie ein Downloadlink um sich das Update manuell herunterzuladen.

| Verbundene Clienten! Updatehistorie Verfügbare Updates Downloads | | |
|--|---------------|--|
| Updatename | Updateversion | Download |
| Update 1 | 1.00 |  Download |
| Update 2 | 2.00 |  Download |
| Update 3 | 3.00 |  Download |

6 Update

Die Updates befinden sich im Downloadordner des Projektes(zipfile) und sind nach dem entpacken immer gleich aufgebaut.

'Update 1', '1.0', 'abc', '/downloads/update1', 'tarxyz'

ist ein Beispielupdate: Update 1=name, 1.0=Version, abc=Prüfsumme, /downloads/update1=Downloadlink, tarxyz=Befehl.

Diese Datei kann ersetzt werden(muss gleiches Muster haben werte in ' ' können variieren.) oder gelöscht werden(dann wird das neuste Update direkt auf den Clienten geladen).