

Netzwerkprogrammierung 2017/ ThinClient

David Weinholz

July 23, 2017

1 Aufgabe

Aufgabe war es eine Client/Server Lösung zu erstellen welche einen Updatemechanismus eines Clienten implementiert.

2 Benötigte Importe

Es wurde Python3 zum Programmieren genutzt.

2.1 Server

Import	Wofür
socket	Verbindung von Clienten annehmen
flask	Nebenbei einen Flask Server laufen lassen, welcher Clienten/Updates anzeigt
threading	Verschiedene Worker Threads pro Client
time	Verschiedene Abfragen wie z.B. Alive nach bestimmter Zeit
datetime	Aktualisieren wann letzter Zugriff eines Clienten war
zipfile	Erstellen von Zips als Updates
json	Parsen von JSON-String
hashlib	Checksum der Updates erstellen
flask_sqlalchemy	Als Datenbank für die verschiedenen Clienten/Updates

2.2 Client

Import	Wofür
socket	Verbindung zum Server aufbauen
json	Parsen von JSON-String
time	Reconnect Versuch nach bestimmter Zeit
math	Aufrunden RAM Angabe
psutil	Angabe RAM(benötigt python3 dev)
subprocess	Ausführen von Shell Befehlen wie Unzip etc
urllib.request	Downloaden Paket von einer URL
cpuinfo	Für Angabe der CPU des Clienten

3 Usage

3.1 Server

Den Server kann man ganz normal mit aufrufen des Scriptes aus der Konsole öffnen. Es wird aber Python3 benötigt. Wichtig ist vorher im Script das URL und LocalUrL Feld anzupassen und auf die IP des Hostes dort zu ändern

Der Serversocket läuft über Port 5001 und die Flask App über Port 5000.

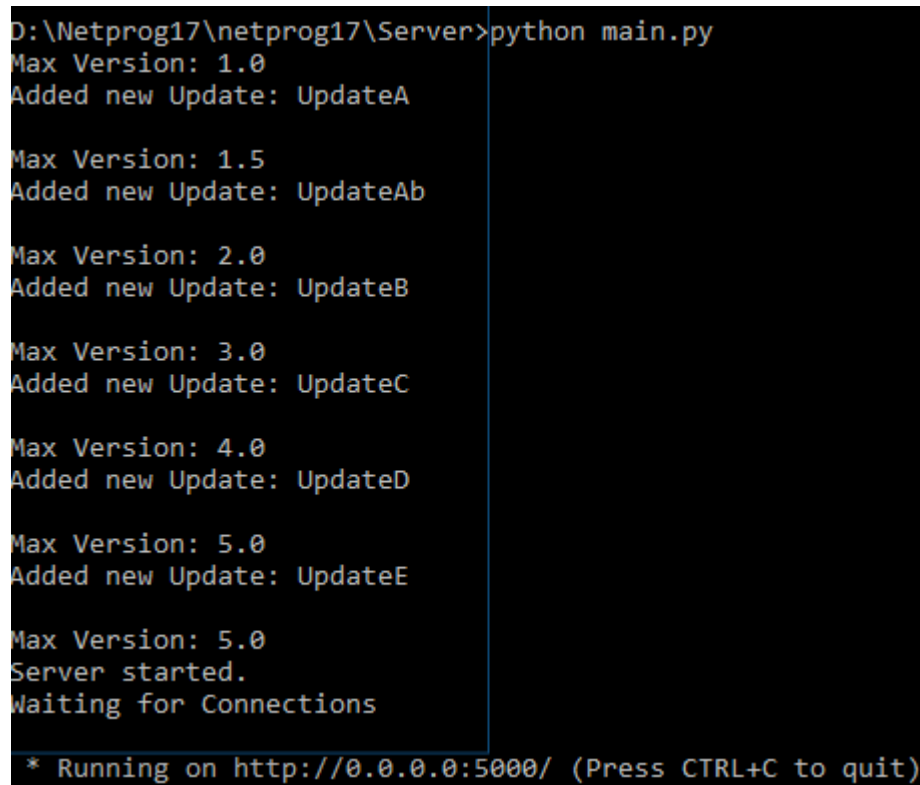
```
LocalUrl='http://192.168.0.59:5000/updates/downloads/'
```

```
URL="http://192.168.0.59:5000"
```

Jeder der sich im selben Netzwerk befindet kann sich mit dem Server verbinden.

Die schonmal verbundenen Clienten bzw. vorhandene Updates werden aus einer Datenbank gelesen.

Existiert diese noch nicht bzw. es existieren keine Updates wird sie mit fünf Updates zum Testen gestartet.



```
D:\Netprog17\netprog17\Server>python main.py
Max Version: 1.0
Added new Update: UpdateA

Max Version: 1.5
Added new Update: UpdateAb

Max Version: 2.0
Added new Update: UpdateB

Max Version: 3.0
Added new Update: UpdateC

Max Version: 4.0
Added new Update: UpdateD

Max Version: 5.0
Added new Update: UpdateE

Max Version: 5.0
Server started.
Waiting for Connections

* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Max Version zeigt dabei an welche Version der Updates gerade die höchste ist.

Ab diesem Punkt kann der Server Verbindungen annehmen.

Zudem ist es möglich zur Laufzeit neue Updates hinzuzufügen.

```

For Entering new Update write New:New
Name of Update: DynamicUpdate
Version of Update: 7
Max Version: 7.0
Added new Update: DynamicUpdate

```

Falls man statt New Quit eingibt beendet sich der Server. Verbindet sich ein neuer Client wird dessen IP angezeigt und es kommt eine Meldung das sich ein neuer Client mit [Hostname] angemeldet hat.

Dieser sendet UpdateRequests, welche der Server beantwortet.

```

('192.168.0.38', 42530)
New Client david-Lenovo-IdeaPad-U530-Touch added!
UpdateMessageSend
192.168.0.38 - - [23/Jul/2017 21:54:21] "GET /updates/downloads/DynamicUpdate HTTP/1

```

In diesem Fall wurde sofort eine UpdateMessage, auf eine Anfrage des Clienten, gesendet, wodurch der Client wusste dass es eine aktuellere Version als seine gibt die er sich dann gleich gedownloadet hat.

Da auch neben den Sockets des Servers eine Flask App läuft kann man sich im Browser angucken welche Clienten es gibt und welche Updates verfügbar sind (mit Downloadlink).

Updates

Hostname	IP	CPU	GPU	RAM[GB]	Datum	Alive
Sorens-MacBook-Air.local	192.168.0.31	Intel(R) Core(TM) i5-5250U CPU @ 1.60GHz	System not Linux or Windows	8	2017-07-23 15:18	False
david-Lenovo-IdeaPad-U530-Touch	192.168.0.42	Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz	['Haswell-ULT Integrated Graphics Controller', 'GK208M [GeForce GT 730M]']	8	2017-07-23 15:22	True
Soeren-pc	192.168.0.14	Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz	['VirtualBox Graphics Adapter']	6	2017-07-23 15:21	False

Strg+S	Clienten
--------	----------

Update	Version	Download
UpdateA.zip	1.0	Download
UpdateAb.zip	1.5	Download
UpdateB.zip	2.0	Download
UpdateC.zip	3.0	Download
UpdateD.zip	4.0	Download
UpdateE.zip	5.0	Download
DynamicUpdate.zip	7.0	Download

Aufrufbar durch die IP des Hostes und auf Port 5000 im lokalen Netzwerk.
In diesem Fall 'http://192.168.0.59:500'.

3.2 Client

Der Client kann auch einfach per aufrufen des Skriptes per Konsole gestartet werden.
Auch hier wird Python 3 benötigt und wichtig ist das man im Skript erst die Server IP anpasst.

SERVERIP= '192.168.0.59'

Der Client versucht sich dann mit dem Server zu verbinden. Ist dieser jedoch Offline oder es existiert bereits eine Verbindung des Clienten mit dem Server kommt eine Fehlermeldung.

```
david@david-Lenovo-IdeaPad-U530-Touch:~/netprog17/Client$ python.py
Trying to connect
No Connection, Server dead or Client already connected
```

Der Client versucht sich dann nach einer bestimmten Zeit (momentan 60 Sekunden) wieder zu verbinden, auch wenn der Client schon verbunden war und der Server dann

down geht , beendet sich der Client nicht sondern versucht in bestimmten Intervallen sich wieder zu verbinden.

```
Trying to connect  
WARNUNG: Sie sollten dieses Programm mit Systemverwalterrechten  
WARNUNG: Ausgabe kann unvollständig oder fehlerhaft werden, Sie  
UpdateRequest
```

Die Warnung in dem Bild resultiert daraus das der Client um seine Grafikkarteninformation herauszufinden ein Shell Kommando aufruft (lshw -c display) anhand dessen Ausgabe die Grafikkarte ermittelt wird.

Da man diese eig. mit sudo für alle Informationen ausführen sollte kommt diese Warnung, welche hier nicht weiter relevant ist da auch mit dem Befehl ohne sudo die nötigen Informationen erlangt werden.

Ist der Client verbunden, schickt er eine Anfrage an den Server mit seinen Informationen welche Version er gerade hat und welche Checksum sein installiertes Paket hat. Ist die Version tiefer als die Max Version des Servers bzw. stimmt die Checksum nicht mit der aktuellen Version des Servers überein schickt der Server einen JSON-String mit Informationen des neuen Updates inklusive URL zum Downloaden und einem Script Befehl welcher ausgeführt werden muss für die Installation.

```
DynamicUpdate.zip  
Archive:  DynamicUpdate.zip  
extracting: downloads/DynamicUpdate.txt  
Updated  
UpdateRequest
```

Der Client downloadet dann die ZIP Datei von der URL und führt den Script Befehl auf sie aus.

Danach wird die UpdateInfo des Clienten noch aktualisiert und die Installation ist beendet.

Der Client wird weiterhin nach einem festgelegtem Zeitintervall nach neuen Updates fragen.