

רשתות מחשבים

פרק 2- תכנות socket

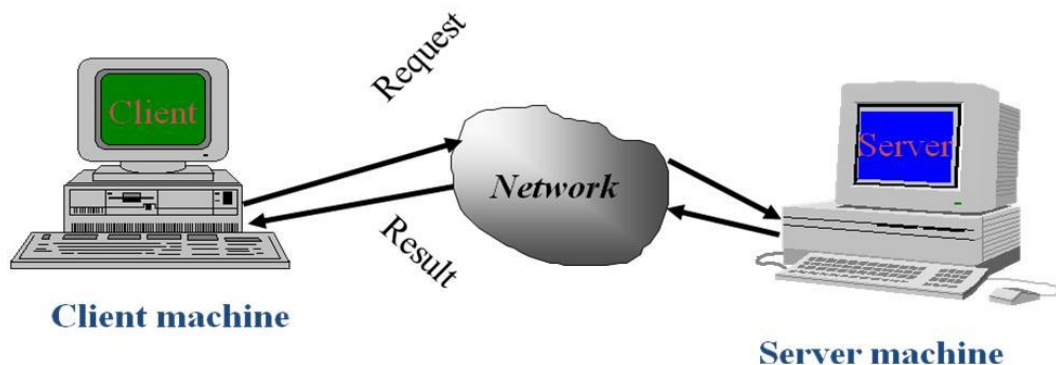
ברק גונן

מבוסס על ספר הלימוד "רשתות מחשבים" מאת

עומר רוזנבוים

מה נלמד בפרק זה?

- ▶ מהי תקשורת שרת - לקוח
- ▶ מהו socket
- ▶ נכתוב שרת ולקוח בשפת python, בעזרת socket
- ▶ השרת והלקוח יתקשרו ביניהם:
 - שליחת הודעות
 - העברת קבצים
 - תמונות
 - וכו'



תקשורת שרת-לקוח

- ▶ שרת - server, לקוח - client
- ▶ צורת התקשורת הנפוצה ביותר באינטרנט
- ▶ ה-server מספק שירות כלשהו
- ▶ ה-client פונה ל-server כדי להשתמש בשירות
- ▶ התקשורת בין השרת והלקוח מתבצעת על ידי socket



מהו socket?

- ▶ Socket הוא נקודת קצה של חיבור בין שני רכיבים
 - אם רוצים להעביר מידע בין מחשבים, צריך לקשר ביניהם
 - גם העברת מידע בין תוכנות שרצות על אותו מחשב מצריכה קישור
 - נקודות הכניסה והיציאה של המידע מכונות socket
- ▶ אפשר להמשיל socket לצינור:
 - ב socket-זרם מידע-זרם של בתים
 - זרימת המידע היא דו כיוונית
 - יש ל socket-נקודות התחלה וסיום



כתובות של socket

▶ כדי ש-socket יוכל לשמש להעברת מידע, צריך להגדיר את נקודות הקצה שלו

▶ הגדרת נקודת קצה מתבצעת ע"י שני מזהים:

◦ מזהה הרכיב- עם איזה מחשב מתקיימת התקשורת?

◦ מזהה התהליך- על המחשב שאיתו מתקיימת התקשורת רצות

מספר תוכנות. עם איזו תוכנה מתקיימת התקשורת?

◦ Socket הוא צירוף של 2 נקודות קצה- שרת ולקוח

▶ מזהה הרכיב: כתובת IP

▶ מזהה התהליך: מספר פורט Port

◦ מספר בטווח 0-65535

▶ Socket מוגדר על ידי צירוף IP ו-Port



מזהה הבניין:
הרצל 1, תל אביב

המחשה - IP ו-Port

- ▶ שרת שכתובת ה-IP שלו היא 1.2.3.4 מאחסן דפי אינטרנט שונים
 - השרת תומך בהעברת דפים רגילה HTTP או מאובטחת HTTPS
- ▶ שרת שכתובת ה-IP שלו היא 5.6.7.8 מספק מספר שירותים
 - השרת תומך בהעברת קבצים, יודע לקשר בין כתובות דומיין לכתובות IP ויודע לטפל באימיילים
- ▶ לאיזה צירוף של IP ו-Port צריך לפנות לקוח שרוצה לבצע תשאול DNS? גלישה מאובטחת לאתר אינטרנט?

תהליכים:

1. גלישה לדפי אינטרנט HTTP - פורט 80
2. גלישה לדפי אינטרנט מאובטחים HTTPS - פורט 443



שרת 1.2.3.4

תהליכים:

1. העברת קבצים FTP - פורט 20
2. מענה לבקשות DNS - פורט 53
3. שליחה וקבל אימיילים SMTP - פורט 25



שרת 5.6.7.8



- ▶ על השרת שכתובתו 1.2.3.4 הותקן דפדפן. תלמיד רוצה לגלוש מהשרת אל אתר אינטרנט, שמאוחסן על אותו השרת. לאיזה IP צריך לפנות?
- ▶ רמז: ה-IP אינו 1.2.3.4
- ▶ תשובה: 127.0.0.1. משמעות IP זה הינה "כתובת הבית", כלומר, התוכנה מולה נפתח ה-socket נמצאת באותו המחשב

תכנות sockets

- ▶ הלקוח מנסה להתחבר לשרת – IP, Port
- ▶ אם השרת מאזין בפורט הנ"ל, נוצר קשר חדש דו כיווני

AF_INET:
Internet
protocol (IP)

SOCK_STREAM:
Connection
type (TCP)

```
first_client.py x
1  import socket
2
3  my_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  my_socket.connect(("127.0.0.1", 8820))
5
6  my_socket.send("Hello".encode())
7  data = my_socket.recv(1024).decode()
8  print("The server sent " + data)
9
10 my_socket.close()
```




תרגיל כיתה - 2.2 לקוח לשרת הדים

▶ הורידו את השרת מהכתובת

▶ https://data.cyber.org.il/networks/echo_server.pyc

▶ שימרו את הקובץ למיקום

C:\networks\work\echo_server.pyc

▶ מה-command line הריצו את הפקודה:

python C:\networks\work\echo_server.pyc

▶ כעת כיתבו לקוח ש:

◦ מתחבר אל השרת (פורט 8820)

◦ שולח הודעה אל השרת

◦ מקבל את תשובת השרת ומדפיס אותה על המסך

תרגיל כיתה - 2.3 כתיבת שרת

```
first_server.py x
1 import socket
2
3 server_socket = socket.socket()
4 server_socket.bind(("0.0.0.0", 8820))
5 server_socket.listen()
6 print("Server is up and running")
7
8 (client_socket, client_address) = server_socket.accept()
9 print("Client connected")
10
11 data = client_socket.recv(1024).decode()
12 print("Client sent: " + data)
13
14 reply = "Hello " + data
15 client_socket.send(reply.encode())
16
17 client_socket.close()
18 server_socket.close()
```

כיתבו שרת
שמקבל את
פניית הלקוח
ועונה לו
Hello

תרגיל כיתה - 2.3 כתיבת שרת

```
first_server.py x
1  import socket
2
3  server_socket = socket.socket()
4  server_socket.bind(("0.0.0.0", 8820))
5  server_socket.listen()
6  print("Server is up and running")
7
8  (client_socket, client_address) = server_socket.accept()
9  print("Client connected")
10
11 data = client_socket.recv(1024).decode()
12 print("Client sent: " + data)
13
14 reply = "Hello " + data
15 client_socket.send(reply.encode())
16
17 client_socket.close()
18 server_socket.close()
```

0.0.0.0:
Listen to all
IP's on this
computer

כיתבו שרת
שמקבל את
פניית הלקוח
ועונה לו
Hello

תרגיל כיתה - 2.3 כתיבת שרת

```
first_server.py x
1 import socket
2
3 server_socket = socket.socket()
4 server_socket.bind(("0.0.0.0", 8820))
5 server_socket.listen()
6 print("Server is up and running")
7
8 (client_socket, client_address) = server_socket.accept()
9 print("Client connected")
10
11 data = client_socket.recv(1024).decode()
12 print("Client sent: " + data)
13
14 reply = "Hello " + data
15 client_socket.send(reply.encode())
16
17 client_socket.close()
18 server_socket.close()
```

כיתבו שרת
שמקבל את
פניית הלקוח
ועונה לו
Hello

Wait for client
connection

תרגיל כיתה - 2.3 כתיבת שרת

```
first_server.py x
1 import socket
2
3 server_socket = socket.socket()
4 server_socket.bind(("0.0.0.0", 8820))
5 server_socket.listen()
6 print("Server is up and running")
7
8 (client_socket, client_address) = server_socket.accept()
9 print("Client connected")
10
11 data = client_socket.recv(1024).decode()
12 print("Client sent: " + data)
13
14 reply = "Hello " + data
15 client_socket.send(reply.encode())
16
17 client_socket.close()
18 server_socket.close()
```

כיתבו שרת
שמקבל את
פניית הלקוח
ועונה לו
Hello

תרגיל כיתה- 2.3 הרצת שרת ולקוח

- ▶ הריצו את השרת והלקוח ב-Pycharm
 - מומלץ כשרוצים לדבג
- ▶ פיתחו שני חלונות command line
 - בראשון הריצו את server.py
 - בשני הריצו את client.py

```
C:\WINDOWS\system32\cmd.exe
C:\Users\barak>cd c:\networks\work\networksbook
c:\Networks\work\NetworksBook>python first_server.py
Server is up and running
Client connected
Client sent: Hello
c:\Networks\work\NetworksBook>
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\barak>cd c:\networks\work\networksbook
c:\Networks\work\NetworksBook>python first_client.py
The server sent Hello Hello
c:\Networks\work\NetworksBook>
```

פרוטוקול תקשורת - תזכורת

- ▶ פרוטוקול Protocol הוא אוסף של חוקים שמאפשר לשתי ישויות או יותר להעביר ביניהן מידע
- ▶ דוגמה:





תרגיל 2.6 – שרת פקודות בסיסי

- ▶ השרת מבצע פקודות שהלקוח שולח ומחזיר תשובה
- ▶ רשימת הפקודות:
 - Time, Name, Rand, Exit (פירוט בספר)
- ▶ טיפים לביצוע התרגיל:
 - הלקוח שולח כמות בתים קבועה - `socket.send(4)`
 - השרת מחזיר כמות בתים משתנה
 - בתחילת תשובת השרת הוסיפו שדה אורך
 - שדה האורך חייב להיות באורך קבוע מראש
 - השרת חייב להיות יציב ולהתמודד עם כל דבר שהלקוח שולח



תרגיל 2.7- שרת פקודות מתקדם

- ▶ בתרגיל זה תכתבו שרת-לקוח של תוכנה שמאפשרת לטכנאי לבצע פעולות שונות על מחשב מרוחק:
 - קבלת צילום מסך
 - הפעלת תוכנות שונות
 - העתקה של קובץ מהשרת
 - הצגת תוכן תיקיה, מחיקת קבצים
- ▶ הדרכה נמצאת בספר הלימוד ובסרטונים הבאים:
 - [הנחיות לתרגיל 2.7](#)
 - [תרגיל 2.7 דוגמת הרצה](#)
- ▶ טיפים לעבודה- בשקף הבא

סיכום- מה למדנו?

- ▶ מודל עבודה שרת - לקוח
- ▶ כתיבת socket
- ▶ פיתוח שרת ולקוח ב-python
- ▶ תכנון פרוטוקול תקשורת

