

תרגיל בית מעשי מס' 3

להגשה עד 14.1.2013

שימו לב: באתר הקורס תוכלו למצוא מסמך עם הנחיות מפורטות לגבי הגשת תרגילים מעשיים. על כל התרגילים המעשיים שתגישו הסמסטר לעמוד בהנחיות הללו.

מטרה

בתרגיל זה נרחיב את הפרוטוקול שהגדרנו בתרגיל 1 ונהפוך אותו ליותר מבוצר.

השרת

השרת מחזיק רשימה של שמות קבצים, ואצל אילו מהלקוחות שלו אפשר למצוא כל קובץ. כשלקוח א' מבקש קובץ השרת דואג לקשר אותו ללקוח ב' שהקובץ נמצא אצלו, כך שהם יוכלו להעביר את הקובץ ביניהם. שימו לב שישנן דרכים רבות לממש התנהגות זו, וה – design המדוייק נתון לבחירתכם.

אתחול השרת מתבצע בשורת הפקודה

```
file_server [port]
```

הפורט להקשבה הוא פרמטר אופציונלי עם ערך ברירת מחדל 2511.

לאחר מכן השרת ממתין ומשרת לקוחות. לשם פשטות, מרגע שהשרת מתחיל לרוץ הוא אינו מסיים את ריצתו. בכל רגע נתון הוא יכול לטפל במספר לקוחות. לשם פשטות, כל לקוח יכול להיות מטופל בצורה סינכרונית (כלומר, לקוח שמבצע פעולה יחכה עד לסיימה לפני שהוא יוכל להכניס את הפקודה הבאה), אבל הסדר בין לקוחות הוא א-סינכרוני: לקוח א' לא צריך להמתין לסיימה של פעולה חוסמת של לקוח ב' לפני שהוא יכול לבצע פעולה משלו.

כשלקוח חדש מתחבר לשרת השרת שולח לו ברכה כלשהי, שהיא שורת טקסט לבחירתכם. למשל:

```
Welcome! I am simple-file-server.
```

לאחר מכן הוא ממתין ללקוח שיתחבר ויכניס פקודות.

הלקוח

אתחול הלקוח מתבצע בשורת הפקודה:

```
file_client dir [hostname [port]]
```

כאשר hostname ו-port הם פרמטרים אופציונליים. ערך ברירת המחדל הוא hostname = localhost, port = 2511. לא ניתן לספק port ללא hostname. הפרמטר hostname יכול להיות שם או כתובת IP.

dir היא הספרייה שבה הלקוח שומר את הקבצים שהוא מקבל מהשרת, וקבצים ממנה הוא יכול לשלוח לשרת. אפשר להניח לגבי הפרמטר הזה את אותן הנחות שהנחנו לגביו בתרגיל 1.

רשתות תקשורת מחשבים, סמסטר א' 2012/13
ביה"ס למדעי המחשב, אוניברסיטת ת"א

הלקוח יתחבר לשרת לפי הפרמטרים hostname, port. אחרי שהלקוח מקבל את הודעת הברכה, הוא ישלח לשרת רשימה של הקבצים הזמינים בספרייה שלו. כשתהליך זה הלקוח ידפיס למשתמש:

```
list of local files was sent to the server
```

לאחר מכן התוכנה תמתין לפקודות מהמשתמש.

המשתמש יכול להכניס את הפקודות הבאות:

1. DIR

בתגובה, השרת ישלח למשתמש רשימה של כל הקבצים הזמינים להורדה מכל המשתמשים המחוברים כעת. תוכנת הלקוח תדפיס אותם, כל קובץ בשורה חדשה (דוגמת הרצה מצויה בסוף בתרגיל).

2. GET filename

השרת יקשר את הלקוח הנוכחי ('א') ללקוח אחר ('ב') שמחזיק אצלו את הקובץ ששמו filename, על מנת שא' יקבל מב' עותק של הקובץ. לאחר שהעברה הושלמה בהצלחה, השרת יעדכן אצלו שגם ללקוח ב' יש עותק של הקובץ.

לשם פשטות - אם ישנם מספר לקוחות שמחזיקים בעותק של הקובץ הדרוש, אפשר לבחור אחד שרירותי שישלח אותו לא', אין צורך להתחשב בעומסים וכו'.

לאחר שהעברה הושלמה בהצלחה תוכנת הלקוח תדפיס למשתמש:

```
transfer complete
```

3. QUIT

יציאה מהתוכנה. שימו לב שעם היציאה השרת צריך לעדכן אצלו שאי אפשר להפנות לקוחות אחרים ללקוח זה כדי לקבל ממנו קבצים.

תוכנת הלקוח גם היא א-סינכרונית: אם לקוח ב' עסוק בלהעביר קובץ גדול ללקוח א' ואז לקוח ג' מבקש גם הוא קובץ מלקוח ב', אזי ג' לא צריך להמתין עד לסיום ההעברה של א' לפני שיתחיל לקבל שירות.

לשם פשטות אפשר להניח:

- אפשר להניח ששמות הקבצים אינם מכילים whitespaces (רווח, t \ וכו'). והם אינם מוסתרים (מתחילים בנקודה).
- שמות הקבצים הם case-sensitive.
- שם קובץ הוא מזהה ייחודי לתוכן הקובץ: אין צורך לתמוך במקרים בהם יש שני קבצים שונים ברשת עם אותו שם.

רשתות תקשורת מחשבים, סמסטר א' 2012/13
ביה"ס למדעי המחשב, אוניברסיטת ת"א

- הקבצים שנמצאים בתיקייה של לקוח הם רק אלו שהיו שם כשתוכנת הלקוח אותחלה, או הועברו לשם ע"י התוכנה. המשתמש לא משנה בעצמו את תוכן התיקייה הזו, ולא מוחק ממנה קבצים.
- לקוח מתנתק מהרשת רק בצורה מסודרת.

דוגמת ריצה

בצד השרת:

```
file_server /specific/a/home/cc/students/cs/my_username/dir1/  
בצד הלקוח (בפונט ירוק – קלט מהמשתמש):
```

```
file_client /specific/a/home/cc/students/cs/my_username/dir2/  
Welcome! I am simple-file-server.
```

המשתמש שולח לשרת את רשימת הקבצים בתיקייה שלו, נניח שהם `hello_world.txt`,
(`MyExcelSheet.xlsx`, `MyFavoriteSong.mp3`)

```
list of local files was sent to the server
```

כעת מתחבר לקוח שני: (נניח שהתיקייה שלו ריקה בתחילה)

```
file_client /specific/a/home/cc/students/cs/other_user/dir/  
Welcome! I am simple-mail-server.
```

```
list of local files was sent to the server
```

```
DIR
```

```
hello_world.txt
```

```
MyExcelSheet.xlsx
```

```
MyFavoriteSong.mp3
```

```
GET hello_world.txt
```

```
transfer complete
```

(השרת מעדכן אצלו שגם ללקוח 2 יש כעת עותק של `hello_world.txt`.)

(כעת מתנתק לקוח 1 בפקודת QUIT. השרת מעדכן אצלו שהקבצים `MyExcelSheet.xlsx` ו-
`MyFavoriteSong.mp3` אינם זמינים יותר להורדה)

כעת מתחבר לקוח 3:

```
file_client /specific/a/home/cc/students/cs/other_user2/dir/  
Welcome! I am simple-mail-server.  
list of local files was sent to the server  
DIR  
hello_world.txt  
  
GET hello_world.txt
```

(השרת דואג שלקוח 3 יוכל לקבל את הקובץ מלקוח 2 שמחזיק אצלו עותק של הקובץ)

```
transfer complete
```

(השרת מעדכן אצלו שגם ללקוח 3 יש כעת עותק של הקובץ)

דרישות התרגיל

ראשית, עליכם לתכנן פרוטוקול אפליקציה מתאים שיעבוד מעל TCP. לאחר מכן, ממשו אותו כפי שנלמד בתרגול. האפליקציה סינכרונית, כלומר אפשר להשתמש בפקודות חוסמות.

את פרוטוקול האפליקציה יש לתעד בבירור, באופן שיאפשר לכל אדם לממש לקוח או שרת ש"ידברו" עם התוכנות שהגשתם.

הדגש בבדיקת הקוד שתגישו ינתן כמובן למימוש התקשורת ברשת. על המימוש להיות יעיל ורובוסטי (robust). אל תשכחו לבדוק שגיאות בערכי חזרה מפונקציות ולטפל בהם בהתאם.

הפלט של התוכנה חייב להיות בדיוק כפי שמוגדר בתרגיל (בלי הדפסות debug וכו') מפני שיהיו בדיקות אוטומטיות.

למקרה שתבחרו להגיש בסביבת nova: ייתכן שזוגות רבים יבדקו את הפתרון שלהם על nova בו-זמנית. לכן, מומלץ להשתמש בפורט שרת שאינו פורט ברירת המחדל בעת בדיקת הקוד.