


## תקשורת ומחשוב

### תרגיל תכנותי 6

#### הוראות להגשת המטלה:

1. את כל המטלות יש להגיש אך ורק דרך מערכת ה-MoodLearn (המידע האישי <= השיעורים שלי <= לחיצה על ) , לא דרך המייל ולא באף דרך אחרת.
2. כל מטלה יש להגיש בנפרד, עד התאריך שנקבע לה (אין הגשה באיחור).
3. ניתן להגיש מטלות בזוגות.
4. יש לתת שם שמכיל ת.ז. של הסטודנט/ים, לקובץ מכוון.
5. יש לכוון את כל הקבצים באחד מהפורמטים הבאים, ZIP, RAR, 7-ZIP, בלבד.
6. חובה להגיש את התרגיל בצרוף קוד המקור בשפת JAVA בלבד. מי שלא יגיש את קוד המקור בצורה שתפתח בפנקס הרשימות (NOTEPAD) העבודה לא תיבדק וייחשב הדבר כאילו לא הגיש שעורי בית ויגרור הורדה בציון.
7. יש להגיש Java Doc
8. יש להגיש קובץ pcap (רק בתרגילי תקשורת)
9. משקל בציון סופי: 8%.

#### תיאור התרגיל:

בתרגיל זה עליכם לבנות מערכת העברת קבצים פרימיטיבית (בדומה ל-FTP) מבוססת על תקשורת TCP/IP תוך שימוש ב-JAVA. ניתן וכדאי להשתמש בתוכנה שפתחתם בתרגיל 3 כתשתית. ספריות רלוונטיות ניתן למצוא באינטרנט. חובה לציין מקור הספריות אשר בשימוש.

#### תיאור המערכת:

השרת (server) מאתחל את עצמו ו"מקשיב". השרת רץ על גבי שרת IP מסוים ומקשיב לפורט הידוע ללקוחות. עד 2 לקוחות (clients) יכולים "לתקשר" עימו (להיות מחוברים לשרת). בשביל הסדר הטוב נסכם שבתרגיל זה נעשה שימוש בפורטים שמספרים 55000-55010 לקוח: תוכנית java שמאפשרת להתחבר לשרת ולהוריד קובץ מתוך השרת. לאחר שלקוח מתנתק הפורט שהתפנה חוזר להיות משאב פנוי (השרת יכול לתמוך ב 2 לקוחות בו זמנית). כל לקוח יכול ליזום הורדת קובץ מהשרת, תוך ניטור ההעברה באמצעות קשר TCP/IP במקביל לערוץ העברת קובץ. עם השלמת העברת חצי מהקובץ בדיוק, יפסיק השרת להעביר את הקובץ, יציג למשתמש את ערך הבית האחרון שהתקבל וימתין לקבלת אישור מהלקוח. הלקוח חייב לקבל אישור מהמפעיל

(משתמש) על מנת להמשיך בתהליך העברת הקובץ. בעת סיום, יש להוציא הודעה מתאימה, בצרוף ערך הבית האחרון שנשלח. שימו לב שניתן להוריד בו זמנית קובץ משני לקוחות שונים. כלומר, יתכן שבעת המתנה לקבלת אישור להמשך הורדה מלקוח אחד הלקוח השני מבקש ומתחיל לקבל את הקובץ במקביל.

הלקוח מציג תפריט פשוט עם מספר אפשרויות:

#### רשימת הודעות שנשלחות מהלקוח (לשרת):

1. מבקש להתחבר לשרת  
<id> <connect>
2. מבקש להוריד קובץ "exe4" (הקובץ בו נמצא תרגיל זה) או קובץ אחר  
<id> <download> <exe4>
3. המשך בהעברת הקובץ  
<proceed>
4. התנתק  
<disconnect>

#### רשימת ההודעות שמתקבלות מהשרת (ללקוח):

1. התחברות לשרת:  
You are connected: <ID>, <port>  
Error: You can not connect (too many users).
2. לקוח מקבל 50% מהקובץ:  
User <ID> downloaded 50% out of file. Last byte is: xxx. Confirm proceeding
3. לקוח מקבל 100% מהקובץ:  
User <ID> downloaded 100% out of file. Last byte is: yyy.
4. לקוח מתנתק  
User <ID> <IP><port> is now disconnected!

### אפיון טכני

השרת יעלה וימתין לפורט קבוע הידוע ללקוחות – לדוגמה 50,000. לקוח שרוצה להוריד קובץ, יוצר קשר עם השרת בפורט ומבקש להוריד קובץ. השרת יפתח ביוזמתו ערוץ TCP אחר (נניח 50,001) ולאחר הקמת הערוץ יוריד הקובץ ללקוח בערוץ הזה – 50,001. בסיום הורדת הקובץ, עם פירוט המצוין לעיל, ייסגר ערוץ 50,001. התקשורת בין השרת ללקוח צריכה להתנהל בשני ערוצי TCP בו זמנית. באחד יועבר המידע-הקובץ (ערוץ ההורדה) ובאחד (ערוץ הבקרה) יועבר מידע על העברת המידע בערוץ השני. רק לאחר שנוצר קשר בערוץ הבקרה, מוקם ערוץ ההורדה. רק לאחר סיום ההורדה, ינותק ערוץ הבקרה. הבקשה להעברת הקובץ, המידע על העברת 50% או 100% והערכים של הבתים יעברו בערוץ הבקרה. הקובץ עצמו יועבר בערוץ ההורדה. באופן כזה יתאפשר לשני לקוחות לבקש הורדת קובץ. הבידול ביניהם יהיה באמצעות הפורט של הלקוח. למען הפשטות, השרת יכול לקרוא את הקובץ ולהכניסו בתוך שטח פנימי בזיכרון. זו אינה חובה אך אפשרות.

### הערות כלליות:

- בתרגיל זה ניתן לכם חופש החלטה רב בקשר לאופן התקשורת בין הלקוח לשרת, אופן מימוש הלקוח ועוד.

- מותר בהחלט להשתמש בדוגמאות קוד מהאינטרנט (אך לא אחד מהשני) אבל עליכם לשלוח בכל אספקט של הקוד בו אתם משתמשים!
- ניתן להניח הנחות מקלות, אך יש לציין כל הנחה מקלה במפורש!
- את התרגיל יש לעשות ב java.
- זהו לא קורס בגרפיקה ולכן אין צורך שהלקוח יהיה יפה במיוחד, אך באופן טבעי נדרש שימוש בחלונות (גראפיים) וב-threads.

בהצלחה!