מטלת מנחה (ממ"ן) 14

הקורס: תכנות מערכות דפנסיבי - 20937

חומר הלימוד למטלה: יחידה 5 - תקשורת

מספר השאלות: 2 **משקל המטלה:** 4

סמסטר: 2022א **מועד אחרון להגשה:** 9.1.2022

בתרגיל זה נממש תוכנת **שרת** לגיבוי ואחזור קבצים ותוכנת **לקוח** שתעבוד מולו.
השרת יכתב בשפת C++ והלקוח בשפת python.

שרת (50%)

השרת יאפשר לכל לקוח לשלוח אליו קבצים לגיבוי ולשלוף את הקבצים האלו במועד מאוחר יותר.

מאפייני השרת:

א. השרת יתמוך בפרוטוקול חסר מצב (stateless)[1], כלומר, לא ישמור מידע בין בקשה לבקשה (כל בקשה עומדת בפני עצמה).
ב. השרת יתמוך בריבוי משתמשים ע"י תהליכונים (threads).

מספר הפורט יקרא <u>מתוך קובץ</u> בצורה הבאה:

- שם הקובץ: port.info
- מיקום הקובץ: באותה תיקיה של קובץ ההרצה (.exe)
- תוכן הקובץ: מספר פורט
  לדוגמא:
  1234

(אם הקובץ לא נמצא, יש להוציא הודעת שגיאה ידידותית ולהימנע ממצב של תעופת התכנית).

אופן הפעולה של השרת:

1. קורא את הפורט מתוך הקובץ port.info
2. בלולאה אין סופית: ממתין לבקשות
3. בעת קבלת בקשה, יוצר thread חדש ומפענח את הבקשה לפי הפרוטוקול הנתון
4. ממשיך לפעול לפי הבקשה שהתקבלה:
   a. בקשה לשמירת קובץ לגיבוי:
   קבצים הנשלחים ע"י הלקוח ישמרו לתוך תיקיה יעודית של השרת, לכל משתמש תהיה תת-תיקיה ובתוכה הקבצים של אותו משתמש.
   לדוגמא: עבור לקוח מספר 1234 וקובץ בשם mmn14.pdf השרת ישמור את הקובץ בנתיב:
   c:\backupsvr\1234\mmn14.pdf
   b. בקשה למחיקת קובץ:
   מוחק את הקובץ הקיים.
   c. בקשה לרשימת הקבצים הקיימים:
   השרת יצור קובץ טקסט המכיל את רשימת הקבצים עבור לקוח זה.

[1] קראו כאן על פרוטוקול חסר מצב: https://en.wikipedia.org/wiki/Stateless_protocol

שם קובץ הטקסט יהיה אוסף תווים רנדומלי באורך 32 תווים (אותיות גדולות, קטנות באנגלית ומספרים)

d. בקשה לאחזור קובץ:

השרת ישלח כתשובה ללקוח את הקובץ המבוקש

5. אחרי הצלחה השרת יחזיר סטטוס הצלחה בהתאם לפרוטוקול

בכל מצב של שגיאה, השרת יחזיר סטטוס שגיאה בהתאם לפרוטוקול

לקוח (50%)

הלקוח יעבוד מול השרת בהתאם לפרוטוקול.

בתחילת הריצה כל לקוח ייצר מספר אקראי ייחודי בגודל 4 בתים. מספר זה ישמש בכל הבקשות שיישלחו לשרת.

גם כאן, אם הקובץ שאמור להיקרא לא נמצא, יש להוציא הודעת שגיאה ידידותית ולהימנע ממצב של תעופת התכנית.

כתובת השרת והפורט יקראו <u>מתוך קובץ</u> בצורה הבאה:

- שם הקובץ: server.info
- מיקום הקובץ: באותה תיקיה של קובץ פייתון הראשי
- תוכן הקובץ: כתובת IP + נקודותיים + מספר פורט
  לדוגמא:
  127.0.0.1:1234

שמות הקבצים לגיבוי ואחזור יקראו מתוך קובץ בצורה הבאה:

- שם הקובץ: backup.info
- מיקום הקובץ: באותה תיקיה של קובץ פייתון הראשי
- תוכן הקובץ: שמות קבצים בלבד ללא נתיב (הקבצים יהיה באותה תיקיה של קובץ פייתון הראשי) .
  לדוגמא:
  mmn14.pdf
  terminator2.avi

כך תראה תיקיה לדוגמא:

```
C:\openu\mmn14>dir /b
mmn14client.py
backup.info
mmn14.pdf
server.info
terminator2.avi

C:\openu\mmn14>type server.info
127.0.0.1:1234
C:\openu\mmn14>type backup.info
mmn14.pdf
terminator2.avi

C:\openu\mmn14>
```

אופן פעולת הלקוח:

1. **יוצר מספר אקראי ייחודי בגודל 4 בתים**

2. **קורא את כתובת השרת והפורט מתוך קובץ server.info**
3. **קורא את שמות הקבצים לגיבוי מתוך קובץ backup.info**
4. שולח בקשה לשרת לקבל את רשימת הקבצים הקיימים בגיבוי
- שרת מחזיר תשובה, יש להציג על המסך את רשימת הקבצים או את הודעת השגיאה שהתקבלה
5. שולח בקשה לשרת לשמירת הקובץ **הראשון** המופיע ב- backup.info
- שרת מחזיר תשובה, יש להציג על המסך את התשובה שהתקבלה (כולל שם הקובץ)
6. שולח בקשה לשמירת הקובץ **השני** המופיע ב- backup.info
- הדפסה של תשובת השרת למסך
7. שולח בקשה לשרת לקבל את רשימת הקבצים הקיימים בגיבוי
- הדפסה של תשובת השרת למסך
8. שולח בקשה לאחזור הקובץ **הראשון** המופיע ב- backup.info
- הדפסה של תשובת השרת למסך ושמירת הקובץ על הדיסק (לצד קובץ פייתון, בשם tmp)
9. שולח בקשה למחיקת הקובץ **הראשון** המופיע ב- backup.info
- הדפסה של תשובת השרת למסך
10. שולח בקשה לאחזור הקובץ **הראשון** המופיע ב- backup.info
- הדפסה של תשובת השרת למסך
11. סיום ויציאה

פרוטוקול התקשורת

עליכם לממש את הפרוטוקול הנתון מעל TCP.

כל השדות המספריים חייבים להיות עם ערכים גדולים מאפס (**unsigned**) ומיוצגים כ- **little endian**

בקשה:

Request	**שדה**	**גודל**	**משמעות**
כותרת (Header)	user id	4 בתים	מייצג את המשתמש
	version	בית	מספר גירסת לקוח
	op	בית	קוד בקשה
	name_len	2 בתים	אורך שם הקובץ
	filename	משתנה	שם הקובץ (ascii) **לא כולל תו מסיים** (null terminated)
תוכן (payload)	size	4 בתים	גודל הקובץ שנשלח
	Payload	משתנה	תוכן הקובץ (בינארי!)

בקשות אפשריות:

Op	**משמעות**	**הערות**
100	שמירה של קובץ לגיבוי	כל השדות מלאים
200	בקשה לאחזור קובץ	שדות size ו- payload לא קיימים
201	בקשה למחיקת קובץ	שדות size ו- payload לא קיימים

202	בקשה לרשימת כל הקבצים של הלקוח	שדות name_len, filename, size, payload לא קיימים

תשובה:

Response	**שדה**	**גודל**	**משמעות**
כותרת (Header)	version	בית	מספר גירסת שרת
	status	2 בתים	סטטוס הבקשה
	name_len	2 בתים	אורך שם הקובץ
	filename	משתנה	שם הקובץ (ascii) **לא כולל תו מסיים** (null terminated)
תוכן (payload)	size	4 בתים	גודל הקובץ שנשלח
	Payload	משתנה	תוכן הקובץ (בינארי!)

תשובות אפשריות:

Status	**משמעות**	**הערות**
210	הצלחה	הקובץ נמצא ושוחזר. כל השדות מלאים
211	הצלחה	רשימת כל הקבצים חזרה ללקוח. כל השדות מלאים
212	הצלחה	גיבוי הצליח או מחיקת קובץ הצליחה. שדה size ו- payload לא קיימים
1001	שגיאה	קובץ לא קיים. שדה size ו- payload לא קיימים
1002	שגיאה	אין קבצים על השרת ללקוח זה. רק שדות version ו- status קיימים
1003	שגיאה	שגיאה כללית. בעיה עם השרת רק שדות version ו- status קיימים

זיכרו! הפרוטוקול הוא בינארי.

כך תיראה לדוגמא בקשה לגיבוי קובץ:

offset					
0	1234	1	100	9	mmn14.pdf
17	29189	25 50 44 46 2D 31 2E 36 ...			

שימו לב!

הפרוטוקול מחייב ולא ניתן לעשות בו שינויים. כפועל יוצא, כל שרת ולקוח המממשים את הפרוטוקול יכולים לעבוד אחד מול השני.

דגשים לקוד שרת:

1. ממשו את התוכנה לפי עקרונות תכנות מונחה עצמים
2. מומלץ (אבל לא חובה) לעשות שימוש בספריות STL
3. ניתן ורצוי להשתמש ביכולות C++11 (לדוגמא פונקציות מסוג למדה, שימוש ב- auto וכו'..).
4. למימוש התקשורת עשו שימוש ב- winsock או בספרית boost
5. שימו לב לייצוג ערכים בזיכרון כ- little-endian או big-endian
6. לקוח יכול לשלוח קובץ בגודל דינמי **גדול**. חשבו על הדרך הנכונה ביותר לקבל כמות מידע גדולה מהלקוח.
7. הקפידו על תיעוד של הקוד (comments)
8. תנו שמות משמעותיים למשתנים, פונקציות ומחלקות. המנעו ממספרי קסם!
9. **אבטחת מידע**

חישבו לאורך כל הדרך על אבטחת מידע. האם בדקתם את הקלט? איך נעשה שימוש בזיכרון דינמי? האם מתבצעת המרת טיפוסים (casting) וכו'..
האם ואיך אפשר לתקוף את השרת? האם השרת יכול לתקוף את הלקוח?

דגשים לקוד לקוח:

1. השתמשו בפייתון **גירסה 3**
2. ממשו את התוכנה לפי עקרונות תכנות מונחה עצמים
3. עשו שימוש בספריות פייתון הסטנדרטיות
4. תוכלו להעזר בספריה **struct** על מנת לעבוד עם נתוני התקשורת בנוחות (בקשות/תשובות)
5. שימו לב לייצוג ערכים בזיכרון כ- little-endian או big-endian
6. השרת מאפשר קבלת קובץ בגודל דינמי **גדול**. חשבו על הדרך הנכונה ביותר לשלוח כמות מידע גדולה לשרת
7. הקפידו על תיעוד של הקוד (comments)
8. **אבטחת מידע**

האם תוכלו לתקף את השרת בצורה כלשהי? האם השרת יכול לתקוף את הלקוח?

הגשה:

1. **שרת**
   א. עליכם להגיש רק את קבצי הקוד (כלומר קבצי h. ו- cpp.).
   שימו לב! על התוכנית להתקמפל ולרוץ בצורה תקינה (ללא צורך בתוספות קבצים ללא קריסות)
   ב. עבודתכם תיבדק במערכת הפעלה חלונות, באמצעות Visual Studio ולכן מומלץ לעבוד עם סביבה זו.
2. **לקוח**
   א. עליכם להגיש רק את קבצי הקוד (כלומר קבצי py.).
   שימו לב! על התוכנית לרוץ בצורה תקינה (ללא צורך בתוספות קבצים, ללא קריסות)
   ב. יש לכלול פונקציה ראשית בשם **main**. פונקציה זו תהיה הפונקציה הראשית של תוכנית הלקוח והיא תעבוד לפי אופן פעולת הלקוח המוצג לעיל.
   טיפ:
   תוכלו להשתמש במנגנון הבא כדי לאפשר עבודה אינטראקטיבית **וגם** הרצה של הקוד

```
if __name__ == "__main__":
```

בהצלחה!