# מטלה מס' 4 – תהליכים וסיגנלים (מימוש שרת לקוח)

## הוראות הגשה:

- את התרגיל יש להגיש למערכת המודול עד ה 20.6 תחת מטלה 4 (עבודה ביחידים).
- אין להדפיס למסך שום דבר מעבר למה שנתבקש בתרגיל.
- יש לוודא שהתרגיל מתקמפל ורץ על שרתי האוניברסיטה ללא שגיאות/אזהרות.
- יש להוסיף במטלות עצמן (כהערה בשתי השורות הראשונות - שורה לכל מגיש) את צירוף ת.ז ושמם המלא של שני המגישים בפורמט:

//Israel Israeli 123456789

- במטלה זו יש להגיש 2 קבצי C בשם **ex4_srv.c, ex4_client.c**.
- חובה לוודא כי כל תוצרי הביניים, כגון קבצים זמניים , שתהליך מייצר לעצמו, ימחקו בהזדמנות הראשונה האפשרית ולכל המאוחר, עם סיום התהליך. בפרט, יש להקפיד שעם סגירת השרת נשאר, אך ורק, עם קבצי ההגשה בלבד (והקבצי out שלהם)

## התרגיל – מערכת שרת לקוח מבוססת תקשורת סיגנלים **והעברת מידע באמצעות קבצים**

1. בתרגיל זה עליכם לבנות שרת שהוא בעצם מחשבון של 4 הפעולות הבסיסיות (חיבור, חיסור כפל וחילוק). השרת מקבל מתוכנת לקוח, זוג מספרים ואת אחת מארבעת פעולות החשבון המבוקשות. על השרת להחזיר כתשובה ללקוח את תוצאת החישוב..
2. אופן פעולת המערכת (השרת והלקוח ביחד):
   a. יש להריץ את תוכנת השרת **ברקע** כלומר, הרצת תהליך שרת, שיוכל לקבל פניות (סיגנלים) מתהלכים אחרים (תהליכי לקוח) . תהליכי הלקוח יאותתו לשרת בכדי לקבל שירותי חישוב מהשרת.
   b. תשתמשו בפקודת **ps** לקבלת הPID של השרת. רשמו לעצמכם מספר זה – הוא יאפשר לכם לפנות לשרת מתוכנת הלקוח בהמשך. כלומר, יאפשר ללקוח לתקשר עם השרת ולשרת להחזיר תשובה ללקוח.
   שימו לב: מספר PID זה, נשאר קבוע, כל עוד השרת רץ והוא בעצם יהווה עבורנו את החלופה לכתובת URL שקיימת בפניית לקוח לשרת סטנדרטי באינטרנט.
   c. יהיה עליכם להריץ במקביל מספר תוכנות לקוח (מספר תהליכי לקוח). כל אחת מהן תתקשר בנפרד עם השרת. כלומר, יפנו לשרת בצורה אסינכרונית.
   d. כל תוכנת לקוח (תהליך לקוח) תקבל, בשורת הפקודה להרצה, 4 פרמטרים נוספים. פקודת הרצת הלקוח תראה כך:
   **ex4 _client.out** P1 P2 P3 P4
   כאשר ה Pi הם הפרמטרים ומוגדרים באופן הבא:
      i. P1 – הוא ה- PID של השרת (מספר מזהה של תהליך השרת – ראה 2.b)
      ii. P2 – הוא הערך המספרי הראשון שעליו יש לבצע את פעולת החישוב

iii. P3 – הוא קוד פעולת החישוב עצמה: 1-חיבור, 2-חיסור, 3-כפל 4-חילוק.

iv. P4 – הוא הערך המספרי השני שאיתו יש לבצע את פעולת החישוב

v. לדוגמא,

אם תכתב, בשורת הפקודה של הטרמינל, הפקודה הבאה:

**ex4 _client.out** 1234 70 2 30

המשמעות היא כי הלקוח יתחבר לשרת שהPID שלו הוא 1234 (תוך שימוש באיתותים וקבצים) ויבקש מהשרת לחשב: 70-30

כלומר 70 "פחות" (קוד פעולה 2 מתורגם ל"חיסור") 30.

על השרת להחזיר ללקוח את התשובה שהיא 40 תוך שימוש באיתותים וקבצים.

e. על הלקוח להמתין לקבלת תשובה מהשרת. יש להוסיף TIMER עבורTIMEOUT. כלומר אם לאחר 30 שניות לא התקבלה תשובה מהשרת הלקוח ידפיס: printf("Client closed because no response was received from the server for 30 seconds")

ויסגור את עצמו. שימו לב: בכל המתנה לתשובה מהשרת, יש לדרוך את הTIMER ועם קבלת התשובה מהשרת, יש לסגור את הטיימר.

f. עם קבלת התשובה מתוכנת השרת, הלקוח ידפיס את התשובה על המסך ויפסיק את פעולתו (יסגר באופן מוחלט ולא ישאיר קבצים זמניים שנפתחו לטובת העברת המידע).

3. יש להשתמש בפונקציית TIMER בתוך השרת. על השרת להכיל TIMER אשר אם אין במשך 60 שניות ברציפות, מאז תחילת הריצה של השרת או מאז הפניה האחרונה של לקוח (המאוחר מבין השניים), אין פניה מאף לקוח, השרת מפסיק את הריצה שלו ונסגר עם ההודעה: printf("The server was closed because no service request was received for the last 60 seconds\n"). שימו לב כל פניית לקוח דורכת מחדש את הTIMER ל60 שניות נוספות.

4. עליכם לכתוב שתי תוכניות נפרדות שירוצו מאותה התיקייה:

a. תוכנית מס' 1 – השרת **ex4_srv.c**

b. תוכנית מספר 2 – הלקוח **ex4_client.c**

5. אלגוריתם התקשורת הנדרש:

a. השרת והלקוח יאותתו אחד לשני באמצעות סיגנלים בלבד.

b. השרת והלקוח ישתמשו בשם קובץ קבוע בשם "to_srv".

c. ראשית יופעל תהליך השרת תוך מחיקת קובץ "to_srv" אם קיים ויכנס להמתנה לקבלת סיגנלים מלקוח כלשהו.

d. בהפעלת תהליך הלקוח, הלקוח **ייצר** את "to_srv" יכתוב לתוכו את הPID של עצמו (של הלקוח – לא של השרת) ובנוסף יעתיק את הפרמטרים האחרים משורת הפקודה (המספרים לחישוב והפעולה בינהם: P2 P3 P4) לתוך "to_srv".

בסיום הכתיבה לקובץ "to_srv" הלקוח ישלח סיגנל לשרת.

e. **סעיף חובה:** מכיוון שיתכן שכאשר תוכנת לקוח מתחילה לרוץ, יש מצב שיש כבר תוכנת לקוח אחרת שכבר רצה והיא כבר משתמשת בקובץ "to_srv" אזי על לקוח זה להמתין עד שהקובץ יתפנה לשימוש. כלומר, אם הקובץ "to_srv" כבר קיים, הלקוח ימתין זמן רנדומלי בין שניה ל5 שניות ואז ינסה בשנית לייצר את הקובץ וכך במשך 10 פעמים ואם לאחר 10 פעמים הוא יכשל הוא יוציא הודעת שגיאה ויסגור את הלקוח תוך סגירה ומחיקת קבצים שהוא פתח ככול שהיו כאלו. פונקציית מערכת לקבלת מספר אקראי **המחויבת שתופיע בקוד שלכם** היא **getrandom()**. (לקבלת מספר אקראי בין 0ל5 אפשר לבצע מודולו 6 על הערך הרנדומלי).

f. עם קבלת הסיגנל אצל השרת, **השרת יפתח תהליך ילד**, שתפקידו יהיה לטפל בלקוח ולבצע עבורו את החישוב המבוקש. במקביל, תהליך האבא יחזור להמתנה לסיגנלים נוספים מלקוחות אחרים.

g. תהליך הילד של השרת, יקרא מתוך ה- "to_srv" את כל הפרמטרים שכתב תהליך הלקוח ששלח את הסיגנל ואז **ימחק את הקובץ "to_srv"!**!!!!

שימו לב: עד ש"to_srv" לא ימחק אף תהליך לקוח אחר לא יוכל להתקשר עם השרת לכן יש למחוק אותו מוקדם ככול האפשר.

h. תהליך הילד ימשיך, וייצר קובץ זמני ייחודי ששמו יהיה "to_client_xxxxxx" כאשר xxxxxx הוא המספר המזהה של תהליך הלקוח שהגיש את הבקשה לחישוב.

i. תהליך הילד של השרת ימשיך ויבצע את פעולת החישוב המבוקשת לפי הפרמטרים שהוא קרא מתוך "to_srv". את תשובת החישוב הוא יכתוב בתוך קובץ "to_client_xxxxxx". ואז תהליך הילד של השרת ישלח סיגנל לתהליך הלקוח כי הוא סיים לבצע את החישוב ויסגור את עצמו.

j. עם קבלת הסיגנל, מתהליך הילד של השרת אצל תהליך הלקוח, הלקוח יקרא את תוצאת החישוב מתוך "to_client_xxxxxx" וידפיס אותה על המסך. **כמו כן, שימו לב כי ללקוח ישנה אחריות:** על הלקוח בשלב זה למחוק את הקובץ "to_client_xxxxxx".

k. הערה: קבצי התקשורת ימוקמו באותה תקייה שבה נמצאים תוכנת השרת ותוכנת הלקוח.

6. **סעיף עקרי בתרגיל – סעיף חובה:**
בשרת, פעולת החישוב, יתבצע בתהליך נפרד!!! – על השרת לפתוח תהליך נפרד לכל לקוח שמתחבר לשרת ומבקש לקבל את תוצאת החישוב. כלומר, החישוב בשרת, תמיד יתבצע במקביל. לא חשוב כמה לקוחות התחברו בו זמנית.

7. הנחת מוצא היא כי יתכנו מספר לקוחות הפונים בו זמנית לשרת ולכן יש לקחת זאת בחשבון - בעיקר בתכנון קובצי התקשורת (למשל יש לקחת בחשבון מצב שבו שתי לקוחות פונים לאותו הקובץ בו זמנית ומפריעים אחד לשני.

8. יש להגן ולבדוק את כל השגיאות המוחזרות (למשל חוסר בפרמטרים בשורת הפקודה או אי הצלחה לפתוח קובץ). בכל המקרים של שגיאה יש להדפיס למסך **printf(ERROR_FROM_EX4\n)** **בלבד** ולסיים את התהליך. אנחנו נכניס שגיאות יזומות ותמיד נרצה לקבל הדפסה זו בלבד ויציאה מהתוכנית.

9. נא להקפיד כי בסיום כל תהליך עליו לנקות ולמחוק את **כל** המשאבים (**לוודא כי לא נשאר כזומבי**, מחק את כל המידע הזמני שיצר כולל קבצים וכ"ו.