משימת תכנות 1

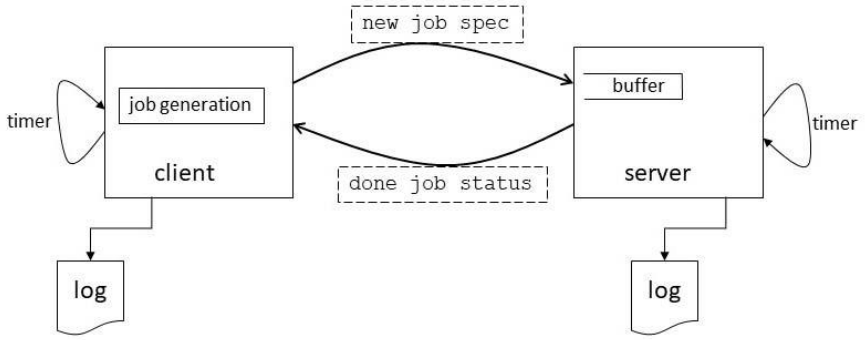
מבוא לתקשורת מחשבים

פרופ' פת-שמיר 26.3.23

הגשה: 18.4.23, במודל. ראו הוראות מפורטות בנספח. נקודות יורדו על הגשה שאינה על פי ההנחיות.

תאור כללי

מטרת התרגיל הינה לרכוש הכרות עם תור M/M/1, וכתיבת אמולציה פשוטה מבוססת TCP וזמן.



במערכת שתבנו יש שרת ולקוח, המחוברים בערוץ TCP. הלקוח מייצר ג'ובים בתהליך הסתברותי (פואסוני), והשרת משרת את הג'ובים בזמן ריצה הסתברותי (התפלגות מעריכית). כאשר ג'וב נוצר אצל הלקוח, נשלחת שורה אל השרת דרך ערוץ ה TCP. כאשר ג'וב מגיע אל השרת, הוא מוכנס לתור FIFO (אם יש מקום - ראו להלן); בכל רגע נתון, הג'וב הראשון בתור מקבל שרות. באשר שרות של ג'וב מסתיים, נשלחת הודעה מהשרת חזרה ללקוח.
במידה ומגיע ג'וב כאשר התור מלא, הג'וב נזרק ונשלחת הודעה מתאימה מהשרת ללקוח.
גם השרת וגם הלקוח ייצרו יומנים (logs) בהם ירשמו זמני השרות של החבילות, תפוסת התור, ומספר חבילות שנזרקו. נתונים אלה יסוכמו בגרפים המראים אספקטים של התנהגות המערכת בתנאים שונים.

תאור מפורט

עליכם להגיש את קוד מקור אשר יצור שתי תכניות בנות-ביצוע (executable programs), שתיהן console applications:

הלקוח (client)

מקבל כפרמטרים בשורת הפקודה (command line arguments) את הפרמטרים הבאים:

- serv_ip: כתובת ה-IP של המחשב בו רץ השרת. ערך: $x_1.x_2.x_3.x_4$, כאשר $x_i \in [0..255]$.
- serv_port: מספר הפורט לו מאזין השרת. ערך: שלם בטווח $[0..65535]$.
- seed: זרע למחולל המספרים הפסבדו-אקראים. ערך: שלם בטווח $[0..32767]$.
- run_id: מספר סידורי של ההרצה (יש לציין מספר זהה ללקוח ולשרת באותה הרצה). ערך: שלם חיובי.
- λ: קצב ממוצע של ייצור ושליחת ההודעות ע"י הלקוח בהודעות לשנייה. ערך: ממשי חיובי.
- T: זמן הפעולה הכולל של המערכת בשניות. לאחר זמן זה לא מיוצרות הודעות נוספות, אך הודעות קיימות עדיין מטופלות. ערך: ממשי חיובי.

בזמן הריצה, הלקוח ראשית מתחבר לשקע של השרת (את כתובתו קיבל כארגומנט), ואז מייצר ג׳ובים עפ״י תהליך פואסוני עם פרמטר λ (גם ניתן כארגומנט). כל ג׳וב נשלח כהודעה לשרת. השרת יבשר ללקוח כאשר הג׳וב סיים ״ביצוע״ או כאשר הג׳וב הופל בגלל תור מלא (שימו לב: ההודעות החוזרות אינן בהכרח לפי סדר ההודעות היוצאות!). בזמן T, הלקוח יסגור את השקע לכתיבה (ע״י קריאה ל shutdown()), ובכך יאותת לשרת על סיום משלוח הג׳ובים. (הלקוח צריך להמשיך ולהמתין לקבלת הודעות על סיום כל הג׳ובים.)

על הלקוח ליצור קובץ client_[run_id].log המכיל שורה לכל חבילה. פורמט כל שורה יהיה:

gen_time end_proc_time total_time

כאשר השדות מציינים, משמאל לימין, את זמן יצור הג׳וב, זמן סיום הג׳וב וההפרש בינם, הוא זמן השהייה במערכת. דיוק הזמן שבו נשתמש הוא מילישניה. אם החבילה נזרקה, יש להדפיס 0 בשדות end_proc_time ו total_time –.

מלבד שורה לכל חבילה, client_[run_id].log יכיל שורת כותרת בשורה הראשונה, לפי הפורמט הבא:

client_[run_id].log: seed = [seed], lambda = [λ], T = [T], total_ pkts = [total_ pkts], total_drops = [total_drops]

כאשר במקום כל סוגריים מרובעים יודפס ערך הפרמטר האמור. במקרה זה, total_ pkts מייצג את מספר החבילות שיוצרו ע״י הלקוח, ואילו total_drops מייצג את מספר החבילות שנזרקו ע״י השרת.

הלקוח מסיים את עבודתו לאחר שזמן T עבר ולאחר שקיבל חיווי חוזר מהשרת עבור כל ההודעות שנשלחו.

לפני סיום העבודה, יציג על גבי ה-stderr את השורה הראשונה של קובץ ה - client_[run_id].log.

תעתיק ריצה (execution transcript) עשוי להראות כך:

```
> client 132.66.16.33 6342 111 5 9.8 150.0
client_5.log: seed=111, lambda=9.8, T=150, total_ pkts=1513,  total_drops=173
>
```

<u>השרת (server)</u>

מקבל כארגומנטים בשורת הפקודה (command line arguments) את הפרמטרים הבאים:

- serv_port: מספר הפורט לו יאזין השרת. ערך: שלם בטווח [0..65535].
- seed: זרע למחולל המספרים הפסבדו-אקראים. ערך: שלם בטווח [0..32767].
- run_id: מספר סידורי של ההרצה (יש לציין מספר זהה ללקוח ולשרת באותה הרצה). ערך: שלם חיובי.
- μ: קצב שירות ממוצע. ערך: ממשי חיובי.
- QSize: נפח האחסון של השרת (בג׳ובים). ערך: שלם חיובי.

כאשר השרת עולה, הוא מקים שקע ומאזין למספר פורט אותו קיבל כארגומנט. לאחר שהלקוח התחבר לשקע, השרת ממתין על השקע לקבלת ג׳ובים (המתוארים ע״י שורת טקסט). כאשר מגיע ג׳וב (שורה), השרת מכניס אותו לתור אם יש מקום (כלומר, אם תפוסת התור הנוכחית קטנה מ QSize), ואחרת, הג׳וב ייזרק.

כל עוד התור לא ריק, השרת משרת את ההודעה הראשונה מהתור. השרות הוא רק אמולציה: למעשה, השרת רק יבחר זמן אקראי t לפי התפלגות מעריכית (אקספוננציאלית) עם פרמטר μ, וימתין t זמן (כזכור, רזולוציית הזמן בה נשתמש היא מילישניות). כאשר הזמן עבר, ההודעה תסולק מהתור והשרת ישלח חיווי ללקוח (דרך שקע הTCP) , המציין כי שירות ההודעה הזאת הסתיים. אם התור לא ריק, השרת יעבור לבצע את ההודעה הבאה.

כאשר הודעה אינה מתקבלת עקב תור מלא, תישלח ללקוח הודעה מיידית המציינת זאת.

על השרת ליצור קובץ server_[run_id].log המכיל שורה לכל שינוי בגודל התור. פורמט כל שורה יהיה:

time q_len

מלבד מידע על גודל התור, server_[run_id].log יכיל שורת כותרת בשורה הראשונה, לפי הפורמט הבא:

server_[run_id].log: seed=[seed], mu=[μ], QSize=[QSize]

כאשר במקום כל סוגריים מרובעים יודפס ערך הפרמטר הרלוונטי.

השרת מסיים את עבודתו לאחר שקיבל מהלקוח את ההודעה המציינת סוף ייצור ג׳ובים, ולאחר שסיים לשרת את אחרונת החבילות.

לפני סיום העבודה, יציג על גבי ה-stderr את השורה הראשונה של קובץ ה - server_[run_id].log.

יש לבצע סיום תוכנית מסודר.

תעתיק ריצה (execution transcript) עשוי להראות כך:

```
> server 6342 511 3 5.0 15
server_3.log: seed=511, mu=5.0, queue size=15
>
```

<u>הניסוי</u>

מקבל כארגומנטים בשורת הפקודה (command line arguments) את הפרמטרים הבאים:

- serv_port: מספר הפורט לו יאזין השרת. ערך: שלם בטווח [0..65535].
- seed: זרע למחולל המספרים הפסבדו-אקראים. ערך: שלם בטווח [0..32767].
- run_id: מספר סידורי של ההרצה (יש לציין מספר זהה ללקוח ולשרת באותה הרצה). ערך: שלם חיובי.
- μ: קצב שירות ממוצע. ערך: ממשי חיובי.
- QSize: נפח האחסון של השרת (בג׳ובים). ערך: שלם חיובי.

כאשר השרת עולה, הוא מקים שקע ומאזין למספר פורט אותו קיבל כארגומנט. לאחר שהלקוח התחבר לשקע, השרת ממתין על השקע לקבלת ג׳ובים (המתוארים ע״י שורת טקסט). כאשר מגיע ג׳וב (שורה), השרת מכניס אותו לתור אם יש מקום (כלומר, אם תפוסת התור הנוכחית קטנה מ QSize), ואחרת, הג׳וב ייזרק.

כל עוד התור לא ריק, השרת משרת את ההודעה הראשונה מהתור. השרות הוא רק אמולציה: למעשה, השרת רק יבחר זמן אקראי t לפי התפלגות מעריכית (אקספוננציאלית) עם פרמטר μ, וימתין t זמן. כאשר הזמן עבר, ההודעה תסולק מהתור והשרת ישלח חיווי ללקוח (דרך שקע הTCP) , המציין כי שירות ההודעה הזאת הסתיים. אם התור לא ריק, השרת יעבור לבצע את ההודעה הבאה.

כאשר הודעה אינה מתקבלת עקב תור מלא, תישלח ללקוח הודעה מיידית המציינת זאת.

על השרת ליצור קובץ server_[run_id].log המכיל שורה לכל שינוי בגודל התור. פורמט כל שורה יהיה:

time q_len

מלבד מידע על גודל התור, server_[run_id].log יכיל שורת כותרת בשורה הראשונה, לפי הפורמט הבא:

server_[run_id].log: seed=[seed], mu=[μ], QSize=[QSize]

כאשר במקום כל סוגריים מרובעים יודפס ערך הפרמטר הרלוונטי.

השרת מסיים את עבודתו לאחר שקיבל מהלקוח את ההודעה המציינת סוף ייצור ג׳ובים, ולאחר שסיים לשרת את אחרונת החבילות.

לפני סיום העבודה, יציג על גבי ה-stderr את השורה הראשונה של קובץ ה - server_[run_id].log.

יש לבצע 10 הרצות שונות עם ערך run_id מ – 0 עד 9, כשבכל הרצה i יתקיימו התנאים המפורטים בטבלה הבאה:

run_id	λ	μ	QSize	T
0	4.55	5	15	150 sec
1	4.60	5	15	150 sec
$\ldots (i)$	$\ldots (4.55 + 0.05i)$	5	15	150 sec
9	5.00	5	15	150 sec

בסיום 10 ההרצות יהיו ברשותכם 20 קיבצי log (10 של הלקוח ו 10 של השרת). יש להשתמש בנתונים אלה, כדי לייצר את הגרפים הבאים:

1. אורך התור כפונקציה של הזמן, לכל ערך של ρ (להזכירכם, $\rho = \mu/\lambda$). כלומר, עשר עקומות שונות באותו הגרף, עקומה לכל ריצה.
2. תפוסת תור מקסימאלית ותפוסת תור ממוצעת כפונקציה של ρ. כלומר, שתי עקומות שונות באותו הגרף, שני ערכים לכל ריצה.
3. total_drops כפונקציה של ρ.
4. זמן שהיה (total_time) מינימאלי, זמן שהיה ממוצע וזמן שהיה מקסימאלי (רק עבור החבילות שלא נזרקו) כפונקציה של ρ. כלומר, שלוש עקומות שונות באותו הגרף, שלושה ערכים לכל ריצה.

טיפים למימוש:

- להמתנה בחלונות ברזולוציה של מילישניה, השתמשו ב Sleep(). (לידיעתכם: בלינוקס הפונקציה המתאימה היא usleep(), שנותנת רזולוציה של מיקרושניה.)
- למציאת זמן בחלונות, השתמשו ב clock() (בלינוקס, הפונקציה התאימה היא gettimeofday()).
- זכרו ש read() ו write() אינם מבטיחים קריאה וכתיבה מלאה בשקע TCP . כדי לוודא כתיבה או קריאה מלאים, ניתן להשתמש בקוד של סטיבנס (lib/readline.c ו lib/writen.c) .
- יש להשתמש באופצית TCP_NODELAY (בעזרת פונקציית setsockopt()) על מנת להבטיח סנכרון במערכת. (נסו לראות מה קורה ללא הדלקת האופציה!)
- את הגרפים אפשר לייצר בגליון אקסל, במטלב, ב gnuplot, או בכל כלי אחר שאתם מכירים.
- יש להשתמש במספר פורט הגדול מ – 5000, כדי לא להשתמש בפורט שבשימוש. רשימת מספרי הפורטים השמורים לשרותים שונים מופיעה ב- http://www.iana.org/assignments/port-numbers/.
- הסבר על ייצור משתנה מעריכי, והסבר על תהליך פואסוני.
- את הערכים של תפוסת התור המקסימאלית ותפוסת התור הממוצעת וכן הערכים של זמן השהיה (total_time) המינימאלי, זמן השהיה הממוצע וזמן השהיה המקסימאלי ניתן לחשב מהנתונים שבקבצי היומן (log files) לאחר ההרצה, או לחשב בזמן ההרצה ולהדפיסם בשורה הראשונה של קובץ היומן המתאים.

בהצלחה!

נספח: הוראות הגשה.

- **התרגיל יוגש רק ע"י אחד השותפים.**
- ההגשה דרך מודל בלבד.
- המימוש חייב להיות במערכת Windows. יש לשלוח קובץ ZIP המכיל את כל ה project של Visual C/C++. יש למחוק את ספריית DEBUG למעט exe.
- בדקו את ההגשה: שלחו אותה לעצמכם בדואר ובנו את האפליקציות על מחשב אחר (למשל במעבדה). נקודות יורדו על הגשה שלא נצליח לבנות בקלות.
- תנו שמות בעלי משמעות לקבצי ההרצה (למשל sender.exe).
- כתבו קוד קריא, כולל שמות בעלי משמעות למשתנים, פונקציות וכו'.
- יש לתעד את הקוד בקובץ README.txt או README.pdf. התיעוד יכיל את הפריטים הבאים:

1. **שמות שני המגישים ומספרי התלמיד שלהם.**
2. הוראות שימוש (כיצד להריץ), במידה ושונות מהתיאור למעלה.
3. תיאור מבנה הקוד באופן כללי (מעין מפת דרכים למימוש), כולל הסבר איזה קוד לא נכתב על ידכם. (מותר ומומלץ להשתמש בספריות. **אסור** ליצור רושם כאילו כתבתם קוד שאחרים יצרו.)
4. באגים ידועים ומגבלות המימוש.
5. אופציונלי: תיאור תוספות המזַכות בבונוס, וגם תוספות אחרות אם יש.