TEL AVIV UNIVERSITY אוניברסיטת תל-אביב

רשתות תקשורת מחשבים
סמסטר א' 2020/21

תרגיל בית מעשי מס' 2

להגשה עד 5.12.2020

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

בצבע אדום, נסמן את כל השינויים מהתרגיל המעשי הראשון. ניתן לדלג על הטקסט השחור, לבחירתכם.

מטרות התרגיל

מימוש תוכנית לא חוסמות באמצעות ריבוב (select). הרחבה קלה של הפרוטוקול מתרגיל 1.

בונוס: שימוש בתהליכונים.

מבוא

בתרגיל זה נרחיב את אפליקציה ונממש אפליקציית רשת במבנה שרת/לקוח. האפליקציה תממש גרסה פשוטה של המשחק Nim:

http://en.wikipedia.org/wiki/Nim

פתרון התרגיל מורכב משתי תוכנות. תוכנית הלקוח (client) – מאפשר משחק אינטראקטיבי לשחקן בודד.

השרת (server) – מנהל את המשחק, ומשחק מול לקוחות רבים.

הלקוח

תוכנית הלקוח זהה לזה שהיה בתרגיל הראשון, מלבד כמה דברים קטנים:

1. השרת משחק מול כמה לקוחות. הוא שומר **רשימת המתנה** של הלקוחות שאינם משחקים, הממוין לפי סדר ההתחברות שלהם. רשימת ההמתנה ממומשת לפי תור, כלומר ברגע ששחקן פעיל יוצא מהמשחק, השחקן הראשון ברשימת ההמתנה יתחיל לשחק מול השרת.
2. רשימת ההמתנה הינה מוגבלת. לכן כל לקוח יכול או לשחק מול השרת, להיות ברשימת ההמתנה או להיות מסורב ע"י השרת.
   כמוכן, הלקוחות הראשונים שהתחברו ישחקו מול השרת, והלקוחות אחריהם היו ברשימת ההמתנה.
   במקרה והלקוח מסורב ע"י השרת, מודפס אצלו:

You are rejected by the server.

במקרה והלקוח נכנס לרשימת המתנה, יודפס:

Waiting to play against the server.

ברגע שהלקוח ישחק מול השרת, יודפס:

Now you are playing against the server!

3. הלקוח יכול לצאת מהמשחק גם כאשר לא מודפסת ההודעה :Your turn. כלומר, בכל רגע נתון תוכנית הלקוח יכולה לקבל את הקלט "Q" מהמשתמש בstdin, ולצאת מהמשחק.
4. בתוכנית הלקוח, אין לקרוא לפונקציות בולמות recv, send (או input()) אלא אם הסוקט שלהן מוכן. ניתן לקרוא לפונקצייה connect שיכולה לבלום עד שהשרת יקבל את הלקוח.

השרת

השרת מנהל משחק בודד. השרת "זוכר" את גדלי הערימות, מקבל צעדי משחק מהלקוח, ומשחק אחרי

כל צעד של הלקוח.

שורת הפקודה להרצה היא:

nim-server.py n_a n_b n_c num-players wait-list-size [port]

- לצורך הבדיקה, כל השחקנים מקבלים את אותם גדלי ערימות n_a, n_b, n_c ברגע שהם מתחילים לשחק מול השרת.
- הפרמטר num-players מציין את הכמות המקסימלית של לקוחות שהשרת משחק נגדם במקביל.
- הפרמטר wait-list-size מציין את הכמות המקסימלית של לקוחות שנמצאים ברשימת ההמתנה של השרת.
- הפורט להקשבה הוא פרמטר אופציונלי עם ערך ברירת מחדל 6444.

למשל:

nim-server.py 8 4 5 2 1 45678

חשוב: אסור לתוכניות השרת לבלום, בקריאות לפונקציות accept, send, recv.

לאחר התחברות לקוח - המשחק מולו מתחיל. השרת מגיב להודעות מהלקוח כפי שפורט בסעיף הקודם.

דרישות התרגיל

- עליכם לרבב בין הפונקציות הבולמות השונות באמצעות select (זכרו שניתן להכניס את stdin לselect).

- אסור לקרוא לפונקציות בולמת (ובפרט, אין לקרוא לsendall\recvall שעשיתם בתרגיל הראשון).
- אלא אם מימשתם את הבונוס, וניתן הדגל multithreading--, אין לפתוח תהליכים\תהליכונים חדשים.
- 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, כך שעל השרת לשאת לקוחות ממחשבים שונים. על המימוש להיות יעיל ורובוסטי (robust).

אל תשכחו לבדוק שגיאות בערכי חזרה מפונקציות ולטפל בהם בהתאם.

ייתכן שזוגות רבים יבדקו את הפתרון שלהם על nova בו-זמנית. לכן, מומלץ להשתמש בפורט שרת שאינו פורט ברירת המחדל בעת בדיקת הקוד (אחרת, פונקציית bind תכשל).

בונוסים (10 נקודות)

ישנם כמה בונוסים אפשריים בתרגיל. בהתאם לבונוסים שמימשתם, שורת ההרצה של השרת תהיה כלדקמן:

nim-server.py n_a n_b n_c num-players wait-list-size [port] [--optimal-strategy] [--multithreading timer]

1. (3 נקודות)- מימושו בצד השרת את האסטרטגיה האופטימלית לניצחון במשחק (כפי שרשום ב https://en.wikipedia.org/wiki/Nim). אם מימשתם את הבונוס, תוכנית השרת צריכה לקבל את הדגל האופציונלי --optimal-strategy . אם הדגל מופיע - השרת תמיד ישחק את האסטרטגיה האופטימלית, אחרת, השרת ישחק באסטרטגיה כפי שתואר בתרגיל הראשון.
2. (7 נקודות- ריבוי תהליכונים) -קראו על multithreading כאן, ובנספח במצגת.
   - אם הדגל multithreading-- מופיע בפקודה - עליכם להשתמש בריבוי תהליכונים במקום select.
   - כמות התהילוכנים שווה לכמות השחקנים (num_players) וכל שחקן היה "משוייך" ללקוח אחר.
   - בנוסף, על השרת לממש timer, ככה שאם לקוח לא שולח הודעה במשך timer שניות מהרגע שהשרת ביקש מהלקוח להזין קלט - השרת מנתק איתו את הconnection. במקרה כזה יופיע אצל הלקוח הודעת disconnected from server מתאימה.

דוגמא ריצה

בצד השרת:

nim-server.py 5 5 5 2 1

לקוח הראשון:

nim.py

Now you are playing against the server!

#Game messages of ex1#

לקוח שני:

nim.py

Now you are playing against the server!

#Game messages of ex1#

לקוח שלישי:

nim.py

Waiting to play against the server.

לקוח רביעי:

nim.py

You are rejected by the server.

אם נניח הלקוח השני לחץ על Q ויצא מהתוכנית, הלקוח השלישי הופך לפעיל. יופיע אצלו ההודעה הבאה:

nim.py

Waiting to play against the server.

Now you are playing against the server!

#Game messages of ex1#

בהצלחה :)