תרגיל 2 במערכות הפעלה למדעי המחשב - סופרים מולקולות

גירסא 2 - 11 במאי 2025

שלב 1 - מחסן אטומים (15 נקודות)

אנחנו מעוניינים לאחסן אצלנו אטומים של פחמן, חמצן, ומימן. אנחנו יכולים לאחסן סדר גודל של 10 בחזקת 18 אטומים מכל סוג.
כתבו שרת בשם atom_warehouse המקבל התחברויות ב-TCP. לאחר שהתקבלה התחברות אנחנו נשתמש ב-IO MUX כדי לטפל בלקוחות המחוברים וכן לקבל התחברויות נוספות.
השרת יקבל בתור ארגומנט ראשון ויחיד (בינתיים) את הפורט עליו הוא מאזין.

פקודות שנוכל לקבל מלקוחות TCP:

ADD CARBON מספר
ADD OXYGEN מספר
ADD HYDROGEN מספר

כל פקודה מתקבלת בתור שורת טקסט מהלקוח (עם ירידת שורה) ומוסיפה אטומים לפי המספר (unsigned int) בפקודה.
לאחר כל קבלה של פקודה יש להדפיס את מספר האטומים מכל סוג במחסן.
בקבלת פקודה בלתי חוקית יש להדפיס הודעת שגיאה.

ממשו לקוח בשם atom_supplier שמתחבר לשרת ושולח בקשות על פי ממשק משתמש לבחירתכם.
הלקוח יקבל בתור ארגומנט ראשון את שם השרת (IP או hostname), ובתור ארגומנט שני את הפורט אליו הוא יתחבר.

שלב 2 - מבקשים מולקולות (15 נקודות)

המחסן שלנו יכול לספק מולקולות של:
מים - H2O
פחמן דו חמצני - CO2
גלוקוז - C6H12O6
אתנול (אלכוהול) - C2H6O

שדרגו את תוכנת המחסן לתוכנה בשם molecule_supplier, כל היכולות מהשלב הקודם נשארות.
השרת יקבל בתור ארגומנט שני את הפורט עליו הוא יקבל בקשות ב-UDP.
שרת המחסן צריך להאזין לפקודות ב-UDP למשל:

DELIVER WATER מספר
DELIVER CARBON DIOXIDE מספר
DELIVER ALCOHOL מספר
DELIVER GLUCOSE מספר

כל פקודה מתקבלת בתור שורת טקסט מהלקוח (עם ירידת שורה), על השרת לענות ללקוח ב-UDP האם הבקשה סופקה או לא.

אם הבקשה סופקה השרת יפחית אטומים כדי לספק מולקולות כנדרש.
בקבלת פקודה בלתי חוקית יש להדפיס הודעת שגיאה.
שימו לב שהשרת יכול לקבל בו זמנית בקשות למולקולות וגם אטומים וצריך להאזין גם ל-TCP וגם ל-UDP.
ממשו לקוח בשם molecule_requester ששולח בקשות לשרת ומדפיס את התשובה.
הלקוח יקבל ארגומנטים כמו atom_supplier ויתקשר על פיהם ב-UDP.

שלב 3 - קונסול (15 נקודות)

המחסן שלנו פותח בר לשירות.
שדרגו את את תוכנת המחסן לתוכנה בשם drinks_bar, כל היכולות מהשלבים הקודמים נשארות.
השרת יאזין גם למקלדת. השרת יכול לקבל מהמקלדת פקודה

GEN SOFT DRINK
GEN VODKA
GEN CHAMPAGNE

אם קיבל את אחת הפקודות הנ"ל השרת יודיע כמה משקאות ניתן לייצר על פי המתכונים:

SOFT DRINK	VODKA	CHAMPAGNE
מים	מים	מים
פחמן דו חמצני	אלכוהול	פחמן דו חמצני
גלוקוז	גלוקוז	אלכוהול

התוכנה תדפיס את מספר המשקאות שניתן לייצר למסך, אין צורך להפחית את מספר האטומים מהמלאי.
בקשות מהמקלדת יתקבלו במקביל לבקשות חיבור ב-TCP, בקשות מלקוחות TCP, ובקשות ב-UDP.

שלב 4 - אופציות התחלה (20 נקודות)

אפשר להתחיל שרת עם מספר ידוע מראש של אטומים מכל סוג בעזרת האופציות. כמו קודם מספר שלא סופק מתחיל מ-0.
בנוסף נממש TIMEOUT שאחריו השרת ירד אם לא קיבל קלט משום ערוץ.
האופציות החדשות הן להלן:

מספר האטומים של חמצן	אופציונלי	-o / --oxygen
מספר האטומים של פחמן	אופציונלי	-c / --carbon
מספר האטומים של מימן	אופציונלי	-h / --hydrogen
בשניות TIMEOUT	אופציונלי	-t / --timeout
פורט TCP	הכרחי	-T / --tcp-port

-U / --udp-port	הכרחי	פורט UDP

שימו לב: אופציה הכרחית משמע שהשרת חייב לקבל אותה עם ערך, אם לא קיבל יש להדפיס שגיאה ולצאת.

בנוסף הלקוחות מעכשיו יקבלו את הארגומנטים בעזרת אופציות:

-h <hostname/IP> -p <port>

שימו לב שהחל משלב זה הארגומנטים מתקבלים בכל סדר שהוא, למשל:

```
./drinks_bar -t 60 -T 5555 -o 12 -U 7777
```

שלב 5 - UDS (15 נקודות)

שדרגו את השרת כך שיתמוך בתקשורת על פני unix domain sockets מסוגי stream ו-datagram לאותם תפקידים כמו TCP ו-UDP בהתאם. לשרת יתווספו אופציות:

-s / --stream-path <UDS stream file path> -d / --datagram-path <UDS datagram filepath>

שדרגו גם את הלקוחות כך שיוכלו לקבל במקום כתובת ופורט ארגומנט:

-f <UDS socket file path>

אם מתקבלים ארגומנטים סותרים (גם כתובת וגם מסלול קובץ) יש להדפיס שגיאה בהתאם.

שלב 6 - מקבול תוכניות (20 נקודות)

אנחנו רוצים שמספר ברמנים יוכלו להתממשק עם השרת בו זמנית.
שדרגו את השרת כך שיתמוך בשמירת המצב (המלאי) בקובץ, מסלול הקובץ יסופק בעזרת האופציה:

-f / --save-file <file path>

אם סופק מסלול לקובץ והקובץ קיים על השרת לטעון ממנו את המלאי הנוכחי (להתעלם מאופציות לאתחול מלאי משלב 4) ולעדכן בו את המלאי כל פעם שיש בו שינוי. אם הקובץ לא קיים יש ליצור אותו עם המלאי ההתחלתי (לפי האופציות משלב 4). (אם לא סופק מסלול יש להתנהג כמו קודם.)
שימו לב שמספר תהליכים יכולים להריץ את השרת בו זמנית ויש לוודא שהם לא מפריעים אחד לשני.
בונוס (5 נקודות): נהלו את כל הממשק מול הקובץ ע"י הגדרת struct למלאי ומיפוי לזיכרון עם מצביע מתאים.

הנחיות נוספות

1. כל שלב צריך להיות בתיקייה בתוך התיקייה הראשית ולהיות מוכל בפני עצמו.
2. מומלץ להשתמש ב-$getopt(3)$ לטיפול בפרמטרים.
3. מומלץ להשתמש ב-$alarm(2)$ למימוש timeout.
4. יש להשתמש ב-$getaddrinfo(3)$ לפתירת hostname בלקוחות.
5. יש לצרף לתרגיל דו"ח (או דו"חות) code coverage.
6. יש לצרף makefile רקורסיבי הבונה את כל השלבים.
7. שימו לב: משקלו של תרגיל זה הוא 10% מהציון הסופי ו-5% הגנה אנא התייחסו בהתאם.