

תרגיל בית 2

ניתן להגיש עד 25.05.2025 עד השעה 23:55

מספר ת"ז	שם:	
מספר ת"ז	שם:	
מספר ת"ז	שם:	

- (1) (18 נקודות) רכיבי מערכת תקשורת כוללים:
- 1 מחשבים
 - 2 רכיבי תקשורת
 - 3 תוכניות שונות

- (א) (3 נקודות) מחשב במערכת מבוזרת יכול להיות
- רק שרת
 - רק לקוח
 - גם שרת וגם לקוח
 - אין תשובה נכונה

- (ב) (3 נקודות) תוכניות של משתמש רצות
- רק על מחשבים
 - רק על רכיבי תקשורת
 - גם על מחשבים וגם על רכיבי תקשורת
 - אין תשובה נכונה

- (ג) (3 נקודות) בקרים הם רכיבים של
- רק מחשב
 - רק רכיבי תקשורת
 - גם מחשב וגם רכיבי תקשורת
 - אין תשובה נכונה

(ד) (3 נקודות) דרייברים הם תוכניות שרצות

- רק על מחשבים
- רק על רכיבי תקשורת
- גם על מחשבים וגם על רכיבי תקשורת
- אין תשובה נכונה

(ה) (3 נקודות) ספק תקשורת מספק

- רק חומרה לתקשורת
- רק תוכנה לתקשורת
- גם חומרה וגם תוכנה לתקשורת
- אין תשובה נכונה

(ו) (3 נקודות) לספקים שונים בדרך כלל

- יש אותה תשתית של תקשורת
- לכל אחד יש תשתית תקשורת שלו
- יש תשתיות שונות וכל ספק בוחר את התשתית שלו
- אין תשובה נכונה

(2) (20 נקודות) נא לבחור תשובה נכונה בכל אחד מהסעיפים הבאים. נא לנמק את תשובתך. תשובה נכונה ללא נימוק נכון אינה מזכה בנקודות.

(א) (5 נקודות) נתבונן ב-RPCs שמחזירות שובות באורך סדר גודל 100KB. האם שימוש ב-UDP במקרה הזה יותר טוב מאשר שימוש ב-TCP.

True / False

100KB גדול מדי לאחסון בחבילה אחת של UDP. מערכת ה-RPC פשוט תצטרך ליישם מחדש חלק ניכר ממה ש-TCP עושה כדי לפצל נתונים אלה לחבילות ולהבטיח שהם מתקבלים בצורה אמינה ויורכבו מחדש.

(ב) (5 נקודות) גם קריאות מקומיות וגם קריאות מרחוק דורשות סמנטיקות חדשות ליישום.

True / False

רק קריאות מרחוק דורשות סמנטיקות חדשות

(ג) (5 נקודות) קריאות מרחוק דורשות סמנטיקות חדשות בגלל שקריאות מרחוק מאופיינות על ידי סוגים אחרים של נפילות ביצוע.

True / False

מערכות RPC עלולות לסבול מכשלים ברשת, ולכן המשתמשים צריכים להתמודד עם זה.

(ד) (5 נקודות) מתכנן גילה שיישום שלו רץ לא מספיק מהר. המתכנן שמע איפשהו שריצות מקבילות זה עתיד של תכנון. לפי כך הוא החליט לחלק את היישום שלו על חלקים רבים שאמורים לרוץ במקביל בתקווה שהוא תמיד ישפר ככה זמני ריצה. האם הוא צדק?

True / False

לא בהכרח. יש רק מספר מוגבל של מעבדים במכונה, והוספת ריצות נוספות לא תהפוך אותה למהירה יותר אם אתם כבר משתמשים בכל המעבדים שלכם. יתר על כן, ישנן עלויות הפעלה ועלויות החלפת הקשר שעשויות להאט אותה. לבסוף, ייתכנו כמה קטעי קוד בסיסיים סדרתיים שלא ניתן להריץ במקביל.

(3) (16 נקודות) נתבונן בהשפעה של תקשורת לחישוב מבוזר.
(א) (8 נקודות) נתבונן בפרוטוקול בקשה-תשובה (request-reply protocol) מיושם על שירות תקשורת עם כשלים כדי לספק סמנטיקה לפחות (at-least-once) של קריאה מרחוק. נא לתת לפחות אתגר אחד שמתכנן חייב לקחת בחישוב ביישום שלו בתסריט הנתון. נא לנמק את האתגר שנבחר. אתגר נכון ללא נימוק נכון אינו מזכה בנקודות.

המתכנן מניח שאם הלקוח מבחין בכשל, הוא לא יכול לדעת האם זה נובע מאובדן הבקשה או מאובדן הודעת התשובה, מקריסת השרת או שתקשורת לקחת זמן רב מהרגיל.
לכן, כאשר הבקשה נשלחת מחדש, הלקוח עשוי לקבל תשובות מאוחרות לבקשה המקורית. המימוש חייב לטפל בכך.

(ב) (8 נקודות) נתבונן ביישום מסירת הודעות אמינה כדי לספק סמנטיקה לכל היותר פעם אחת (at-most-once). היישום משתמש בשעונים מסונכרנים כדי לדחות הודעות כפולות. תהליכים ממקמים את ערך השעון המקומי שלהם ("חותמת זמן") בהודעות שהם שולחים. כל מי שמקבל הודעה שומר טבלה המציגה עבור כל תהליך שליחה את חותמת הזמן הגדולה ביותר שראה. נא להסביר מתי רשאי תהליך להתעלם מהודעה הנושאת חותמת זמן T אם הוא רשם את ההודעה האחרונה שהתקבלה מאותו תהליך כבעלת חותמת זמן?

אם T קטן או שווה זמן שאחריו חובה לשלוח את ההודעה פעם נוספת.

(4) (18 נקודות) נתבונן בפרוטוקולים UDP ו-TCP. נא להשוואות אתגרים של מתכנן שכותב יישום בתקשורת UDP ותקשורת TCP עבור כל אחד מהסוגים של תקשורת:

(א) (6 נקודות) גישה לטרמינל וירטואלי (לדוגמה, Telnet)

משך הזמן הארוך של המשימות, צורך באמינות ורצפי התווים הלא מובנים המועברים הפכים את התקשורת TCP למתאימה ביותר ליישום זה. הביצועים אינם קריטיים ביישום זה, ולכן העלויות הקבועות הן בעלות חשיבות מועטה.

(ב) (6 נקודות) העברת קבצים (לדוגמה, FTP)

קריאות קבצים גורמת העברת כמויות גדולות של נתונים. גישה UDP תהיה בסדר אם שיעורי השגיאות נמוכים וההודעות יכולות להיות גדולות. אך באינטרנט, דרישות אלו אינן מתקיימות, ולכן משתמשים ב-TCP.

(ג) (6 נקודות) גלישת מידע (לדוגמה, HTTP)

ניתן להשתמש גם ב-UDP וגם ב-TCP כמות הנתונים המועבר בכל טרנסקציה יכולה להיות גדול למדי, ולכן בפועל משתמשים ב-TCP.

(5) (12 נקודות) נתבונן בגרסה פשוטה ביותר של אלגוריתם איסוף האשפה המבוזר מפעילה את addRef באתר בו נמצא אובייקט מרוחק בכל פעם שנוצר פרוקסי ואת removeRef בכל פעם שנמחק פרוקסי. נתבונן בכמה ההשפעות האפשריות של כשלים בתקשורת ובתהליך הפעלת האלגוריתם. נא לתאר השלכות של כל אחת מההפצות ונא להציע כיצד להתגבר על כל אחת מההשפעות הללו. נא לנמק את הנכונות של הפתרונות. פתרון נכון ללא נימוק נכון אינו מזכה בנקודות. (א) (4 נקודות) הודעת AddRef נאבדה

האתר שבבעלותו אינו יודע על הפרוקסי של הלקוח ועשוי למחוק את האובייקט המרוחק כאשר הוא עדיין בשימוש. (הלקוח אינו מאפשר כשל זה).

(ב) (4 נקודות) הודעת RemoveRef נאבדה

האתר שבבעלותו אינו יודע שלאובייקט המרוחק יש משתמש אחד פחות. ייתכן שהוא ימשיך לשמור את האובייקט המרוחק כאשר הוא אינו כבר בשימוש.

(ג) (4 נקודות) תהליך המחזיק פרוקסי קורס –

האתר שבבעלותו עשוי להמשיך לשמור את האובייקט המרוחק כאשר הוא אינו בשימוש.

(6) (16 נקודות) נא להביא דוגמא מקורית, לא מועתקות משום מקור (ארבע דוגמאות בסך הכל) לכל אחד מה מקרים המתוארים בטבלה הבאה:

	<i>Time – coupled</i>	<i>Time – uncoupled</i>
<i>Space – coupled</i>		
<i>Space – uncoupled</i>		

כל דוגמא נכונה מתקבלת.

בהצלחה 😊