

מערכות שרת לקוח ותכנות מקבילי 31666

מבחן אמצע, סמסטר ב' תשע"ג, 13/05/2013 (פתרון)

הוראות לנבחן: משך הבחינה 90 דקות. חומרי העזר המותרים הם השקפים של הקורס שנמסרו באתר הקורס. רשום את תשובותיך בגוף הבחינה במקום המתאים לכל שאלה ושאלה. הקפד על כתב יד ברור ומסודר. השימוש בשפה האנגלית מותר. בכדי שתוכל לערוך ולתקן את תשובותיך מומלץ מאוד להשתמש בעיפרון ומחקה. השאלון מכיל 8 שאלות ונפרש על פני 5 עמודים.

שאלה 1

[10%]

תאר בקצרה שלושה ייתרונות עקריים המאפיינים עבודה ברשת מחשבים על פני עבודה על מחשב מבודד

- א. שיתוף משאבים: גישה לשטחי אחסון, מאגרי מידע, ומשאבי עיבוד (מעבדים) נוספים מעבר למחשב הבודד.
- ב. אמינות: כשל במשאבים המקומיים ייפתר על ידי שימוש במשאבי רשת חלופיים (למשל: אם המדפסת המקומית לא פועלת, יש עוד מדפסות רשת שישמשו כתחליף).
- ג. קישוריות חברתית: החלפת הודעות דואר, שיתוף מסמכים
- ד. שיתוף וביזור בסיסי מידע מדעיים: בסיס נתונים, חישוב מבוזר
- ה. ניידות: גישה למידע וקבצים מכל מקום ברשת

שאלה 2

[10%]

מנה בקצרה את סוגי הרשתות השונות שסקרנו בקורס.

1. Personal Area Networks (PAN)
2. Local Area Networks (LAN)
3. Metropolitan Area Networks (MAN)
4. Wide Area Networks (WAN)
5. Wireless Networks (LAN/WiFi)

שאלה 3 [10%]

א. תאר בקצרה את שכבת ה Presentation שבמודל 7 השכבות של OSI
ב. מנה לפחות שלוש דוגמאות של פעולות המתבצעות בשכבה זו.

ו. שכבת ה-Presentation היא השכבה השישית במודל 7 השכבות של OSI. מטרת השכבה הזו היא להמיר מידע (דו-כווני) עם שכבה 7 (שכבת האפליקציה) לפורמט שמוכר לכל המחשבים הקשורים ברשת, ובנוסף לכך גם לאפשר הצפנה ודחיסת מידע למטרות אבטחה ויעילות תעבורה

- ז. 1. הצפנת מידע רגיש העובר ברשתות ציבוריות (אבטחת מידע)
2. דחיסת מידע (לשם הגדלת מהירות תעבורה), או להעברת ספריות ענקיות כקובץ בודד
3. המרה מפורמט ASCII לפורמט EBDIC (מכונות INTEL למכונות IBM), או המרת קוד בינארי מ-"big endian" ל-"little endian", וכדומה
4. המרת קובצי מולטימדיה מפורמטי כמו mp3 ל-wav, וכדומה

שאלה 4 [10%]

תאר בראשי פרקים בלבד את המנגנון הסטנדרטי שבאמצעותו נוצר הקשר (ההתחלתו) שבין שרת ולקוח (תהליך שלפעמים נקרא Hand-Shaking).

- Stage 1: Server starts listening for clients
Stage 2: A client send connection request to server ("HELLO from client")
At the transport layer this is usually a TCP packet with a SYN flag=1
Stage 3: Server acknowledges client request and accepts (or rejects) connection
At the transport layer this is done with a TCP packet with SYN flag=1
Stage 4: If server acknowledges connection request and accepts it, and client accepts Acknowledgement, then the connection is established

שאלה 5 [16%]

לפניך רשימת תוכנות ידועות היטב. סמן ✓ במשבצת המתאימה לצד כל תוכנה כזו באם היא תוכנת שרת או תוכנת לקוח (תשובה שגויה תחסיר נקודה)

- | | | | |
|----------------------------|-------------------------------------|------|-------------------------------------|
| א. MS Internet Explorer | <input type="checkbox"/> | שרת | <input type="checkbox"/> |
| ב. Microsoft Mail Exchange | <input checked="" type="checkbox"/> | שרת | <input checked="" type="checkbox"/> |
| ג. Mozilla Firefox | <input checked="" type="checkbox"/> | לקוח | <input type="checkbox"/> |
| ד. Microsoft Outlook | <input checked="" type="checkbox"/> | שרת | <input type="checkbox"/> |
| ה. Google Gmail | <input checked="" type="checkbox"/> | לקוח | <input type="checkbox"/> |
| ו. Google Chrome | <input checked="" type="checkbox"/> | לקוח | <input type="checkbox"/> |
| ז. Google Drive (Cloud) | <input type="checkbox"/> | לקוח | <input checked="" type="checkbox"/> |
| ח. Dropbox Cloud Storage | <input type="checkbox"/> | לקוח | <input checked="" type="checkbox"/> |

שאלה 6

לפניך רשימה של טענות שונות. לגבי כל טענה, סמן ✓ בתיבה המתאימה אם ורק אם הטענה נכונה. תשובה נכונה מזכה ב-2 נקודות. תשובה שגויה תוריד 2 נקודות. (בכל מקרה הסיכום הכללי לא יהיה שלילי).

- א בכל מערכת שרת לקוח יש בדיוק שני מחשבים: השרת והלקוח
- ב תיתכן סיטואציה בה השרת (server) וכל לקוחותיו (clients) שוכנים באותו המחשב
- ג זה בלתי אפשרי לחלוטין ששרת אינטרנט (web server) וגם שרת דואר (mail server) יפעלו יחדיו באותו מחשב
- ד קיימת אפשרות עקרונית שחבילת IP (IP datagram) תהיה בגודל של 128KB.
- ה קיימת אפשרות עקרונית שראש חבילת TCP (TCP header) יהיה בגודל של 42 Bytes.
- ו ניתן לחבר ערים כמו לונדון וניו-יורק ברשת MAN משותפת
- ז בפרוטוקול TCP/IP: אם חבילת UDP הנשלחת ממחשב A למחשב B הולכת לאיבוד ואינה מגיעה לאחר פרק זמן קצוב, מחשב A ישלח שוב פעם את אותה חבילה למחשב B.
- ח קיימים מחשבים אשר בו-זמנית מחוברים ליותר מרשת אחת
- ט תיתכן סיטואציה שבה תוכנת משתמש אחת תהיה מחוברת בו-זמנית לשני שרתים (servers) שונים בשני מקומות מרוחקים אחד מהשני
- י HTTP הוא פרוטוקול תקשורת שבין לקוח לשרת אינטרנט (web server)
- יא חבילת אתרנט (ethernet frame) תמיד תסיים את דרכה בשכבת האפליקציה (Application Layer)
- יב "אלגוריתם החלון הגולש" (Sliding Window Algorithm) נועד לפתור את בעיית ההרעבה (starvation) בשכבה הפיזית של משלוח חבילות אתרנט (Ethernet Frames).

הערות:

- א. מערכת שרת לקוח תכיל בדרך כלל מספר רב של לקוחות אל מול שרת יחיד (ולפעמים גם מספר שרתים)
- ב. ראינו כבר בכמה הזדמנויות בכיתה דוגמאות לכך שהשרת והלקוח רצים באותו המחשב (אפילו דוגמא לשרת httpd ולקוח explorer שהתחבר אליו). זה אפילו מקובל בהרבה מערכות מורכבות הרצות על מחשב בודד, לחלק יישומים לכמה חלקים קטנים יותר על פי מודל השרת לקוח.
- ג. מקובל מאוד בהרבה אירגונים (קטנים בעיקר) ששרת האינטרנט ושרת הדואר שוכנים ביחד על אותה מכונה. זה אפילו מומלץ וגם עשוי להפחית עלויות.
- ד. על פי מבנה חבילת IP גודל השדה שמציין את גודל החבילה בביתים הוא 16 ביטים. מזה נגזר שהגודל המקסימלי של חבילת IP הוא 64KB.
- ה. ראש של חבילת TCP או חבילת IP חייב להיות כפולה שלמה של מילים (32bit word). או במילים אחרות, הגודל חייב להיות כפולה שלמה של 4Bytes. לכן לא תיתכן חבילת TCP שגודל הראש שלה הוא 42Bytes
- ו. רשת MAN מיועדת עבור שטחים עירוניים (Metropolitan Area Network) ואינה כוללת כבלים תת-ימיים וחומרה נוספת הדרושה לחיבור ערים מרוחקות כמו לונדון וניו-יורק
- ז. בפרוטוקול UDP אין דרישה ובדיקה של הגעת חבילות ליעדיהן. חבילת UDP היוצאת ממחשב A ליעד אחר אינה מנוטרת ואין שום דרך לדעת אם הגיעה ליעדה או לא.
- ח. בוודאי, ללא מחשבים כאלה רשת האינטרנט לא היתה קיימת. בדרך כלל מחשבים כאלה נקראים נתבים (routers) או bridges והם כוללים שניים או יותר כרטיסי רשת, אשר כל אחד מחובר לרשת נפרדת. הנתב (או הגשר) משמש להעברה והמרה של חבילות מרשת אחת לרשת שניה.
- ט. זוהי סיטואציה מאוד שכיחה. למשל דפדפן ה-exploerer עשוי להיות מחובר בו זמנית לשרת אינטרנט שנמצא באוסטרליה ולשרת אינטרנט נוסף שנמצא באוהיוקיטון (בשני טאבים שונים כמובן).
- י. אכן, HTTP, או בשמו המלא: Hyper Text Transport Protocol הוא פרוטוקול האינטרנט שבאמצעותו דפדפן אינטרנט (client) מתחבר ומתקשר עם שרת אינטרנט (server).
- יא. חבילות ethernet פועלות בשכבה הראשונה והשנייה של מודל ה-OSI – השכבה הפיזית ושכבת ה-datalink. חבילת ethernet היא יחידת המידע הבסיסית של רשת ה-Ethernet, והיא נשלחת בין כרטיסי ethernet ברמת החומרה בלבד, ולעולם אינה מגיעה לשכבה השביעית של מודל ה-OSI (שכבת האפליקציה).
- יב. אלגוריתם החלון הגולש (Sliding Window Algorithm) פועל בשכבת ה-Transport ואינו נוגע כלל לשכבה הפיזית או לחבילות ethernet.

רשום תוכנית Python קצרה בשם `address_class(ip)` המקבלת מספר IP בצורה סטנדרטית (כמו למשל: 192.168.2.103) ומחזירה את שם המחלקה אליה שייכת כתובת זו (classes A, B, C, D, E), כפי שהוגדרו בשקפים של הקורס).
למשל:

```
address_class("192.168.2.103") => 'C'
```

תשובה:

שני פתרונות אפשריים:

```
def address_class(ip):
    decimals = ip.split(".")
    d1 = int(decimals[0])
    if d1 < 128:
        return 'A'
    elif d1 < 192:
        return 'B'
    elif d1 < 224:
        return 'C'
    elif d1 < 240:
        return 'D'
    elif d1 < 248:
        return 'E'
    else:
        return 'F'
```

```
def address_class(ip):
    decimals = ip.split(".")
    bits = bin(int(decimals[0]))[2:]
    if bits[0] == '0':
        return 'A'
    elif bits[1] == '0':
        return 'B'
    elif bits[2] == '0':
        return 'C'
    elif bits[3] == '0':
        return 'D'
    elif bits[4] == '0':
        return 'E'
    else:
        return 'F'
```

שאלה 8 [10%]

```
05 32 0a 17
00 00 00 01
00 00 00 00
50 02 07 af
00 00 00 00
62 72 61 75
64 65 00 00
```

נתונה חבילת TCP הבאה (TCP segment) בצורה hexadecimal.
חלץ מתוכה את השדות הבאים (בצורה עשרונית - decimal):

- א. Source Port
- ב. Destination Port
- ג. Window Size
- ד. האם זוהי חבילה סופית (Final) המסמנת סגירת קשר מצד השולח?

תשובה: (רשום תשובות סופיות בלבד)

```
א. 1330
ב. 2583
ג. 1967
ד. False
```

ה. רשום פונקציית Python קצרה בשם winsize() המקבלת חבילת TCP בצורה hexadecimal (מחרוזת נקיה ללא רווחים) ומחזירה את גודל חלון ה-TCP של השולח (Window Size) בצורה עשרונית (decimal). דוגמא:

```
winsize(segment) => 76120
```

```
def winsize(hexstr): # hexstr is a TCP segment in hex format
    window = hexstr[28:32]
    return int(window, 16)
```