

17 כתובות משנה ורשת משנה

[הכול בפרק זה ישתמש ב-IPv4, לא ב-IPv6. הרעיונות הם basically אותו הדבר; זה פשוט יותר קל ללמוד עם IPv4.]

אם אתם צריכים לחזור על פעולות ביטיות, נא לעיין בנספח: פעולות ביטיות.

בפרק הזה:

- ייצוג כתובת
- המרת כתובת מכתובת נקודות ומספרים לערך
- המרת ערך חזרה לנקודות ומספרים
- חידוד מושגי משנה ורשת
- חידוד מושגי מסכת משנה
- מציאת מסכת משנה מתוך ייצוג סלאש
- מציאת הרשת לכתובת IP, בהנחה שהכתובת ומסכת המשנה ידועים

17.1 ייצוג כתובת

זכרו שכתובות IPv4 מוצגות בדרך כלל בכתובת נקודות ומספרים, לדוגמה 198.51.100.10.

וגם זכרו שכל אחד מהמספרים האלה הוא בית, שיכולה להיות לו ערכים בין 0 ל-255, ב-decimal (שזה גם אותו הדבר כמו FF-00 ב-hexadecimal ו-00000000-11111111 ב-binary).

אז בעצם, ייצוג הנקודות והמספרים נמצא שם רק לנוחות שלנו כבני אדם.

עצה: זכרו שבסיסי מספרים שונים כמו binary, hex, ו-decimal הם רק דרכים שונות לרשום ערך. זה כמו שפות שונות לייצוג אותו ערך מספרי.

כשערך מאוחסן במשתנה, כדאי לחשוב עליו כערך מספרי טהור—ללא שום בסיס בכלל. זה רק כשאתם כותבים אותו (בקוד או מדפיסים אותו) שהבסיס חשוב.

לדוגמה, ב-Python כל הדברים מודפסים ב-decimal (בסיס 10) כברירת מחדל. יש לו שיטות שונות להחליף את זה ולהדפיס בבסיס אחר.

נראה כעת את הכתובת 198.51.100.10 ב-hex: c6.33.64.0a.

עכשיו נמזג את כל הבתים האלה למספר hex אחד: c633640a.

אם נמיר את c633640a ל-decimal, נקבל: 3325256714.

אז בכל דרך, כל אחת מהצורות האלה מייצגת את אותה כתובת IP:

- 198.51.100.10
- c6.33.64.0a
- c633640a
- 3325256714

האם זה מובן?

אבל למה?

ובכן, אנחנו עומדים לבצע חישובים על כתובת IP. עכשיו, אנחנו יכולים לעשות את החישובים על כל בית בנפרד וזה יעבוד מצוין.

אבל מסתבר שאם נארוז את כל הבתים האלה למספר אחד, אנחנו יכולים לעשות את החישובים על כל הבתים בבת אחת, וזה הופך את הכל לפשוט יותר. הישארו איתנו!

17.2 המרת כתובת מנקודות ומספרים

המטרה שלנו בפרק זה היא לקחת מיתר נקודות ומספרים כמו 198.51.100.10 ולהמיר אותו לערך המתאים, כמו 3325256714 (ב-decimal).

אפשר לעשות זאת בעזרת מניפולציות מיתר כמו בפרק הקודם, אבל בואו נעשה זאת בצורה ביטית יותר.

נחלץ את המספרים האישיים מכתובת ה-IP (ב-Python אפשר לעשות זאת עם `((split))`).

המיתר:

"198.51.100.10" הופך לרשימה של מיתרים:

["198", "51", "100", "10"] עכשיו, נמיר כל אחד מהם למספרים שלמים. (ב-Python אפשר לעשות זאת עם `int()`, או עם `map()`, או עם `list comprehension`).

[198, 51, 100, 10] עכשיו, אני אכתוב את המספרים האלה ב-hex כי זה יעשה את הצעדים הבאים ברורים יותר. אבל זכרו שהם רק מאוחסנים כערכים מספריים, אז ב-Python לא ידפיסו אותם כ-hex אלא אם תבקשו את זה.

[0xc6, 0x33, 0x64, 0x0a] כדי לבנות את המספר שלנו, נשען על כמה פעולות ביטיות: OR ביטי ו-shift ביטי.

למטרת הדוגמה, נבצע את החישוב בצורה קשיחה:

$(0xc6 \ll 24) | (0x33 \ll 16) | (0x64 \ll 8) | 0x0a$ הריצה בזה ב-Python נותנת לנו את המספר 3325256714 ב-decimal. המרה ל-hex מחזירה אותנו ל-0xc633640a.

אפשר להשתמש בנוסחה הזו להמיר כל קבוצה של 4 בתים למספר ארוז.

יש גם לולאה חכמה שאפשר להריץ כדי לעשות את זה בית אחד בכל פעם. בדקו אם תצליחו להבין את זה כאתגר נוסף! שלחו הודעה למורה אם הייתם חכמים מספיק.

17.3 המרת כתובת לנקודות ומספרים

מה אם יש לכם את הערך מהחלק הקודם, 3325256714, ואתם רוצים להחזיר אותו ככתובת IP?

אפשר להשתמש ב-shift כדי לעשות את זה גם! אבל יש לבצע גם AND Masking כדי לחלץ רק את החלקים של המספר שצריך.

נמיר ל-hex כי כל בית הוא בדיוק 2 ספרות hex וזה קצת יותר קל לראות: 0xc633640a.

עכשיו, נסתכל על המספר הזה לאחר ש-shift אותו ב-0 ביטים, 8 ביטים, 16 ביטים ו-24 ביטים:

```
0xc633640a >> 24 == 0x000000c6
0xc633640a >> 16 == 0x0000c633
0xc633640a >> 8 == 0x00c63364
0xc633640a >> 0 == 0xc633640a
```

אם תסתכלו רק על שתי הספרות בצד ימין, תראו שהן הבתים של המספר המקורי:

```
0xc633640a >> 24 == 0x000000 c6
0xc633640a >> 16 == 0x0000c6 33
0xc633640a >> 8 == 0x00c633 64
0xc633640a >> 0 == 0xc63364 0a
```

אז אנחנו בכיוון הנכון, אלא אם כן תסתכלו למשל על ה-right shift ב-8, אז תקבלו:

```
0xc633640a >> 8 == 0x00c63364
```

אז כן, אני מעוניין בבית 0x64 כפי שמופיע בצד ימין, אבל לא ב-0xc633. איך אפשר לאפס את החלק הגבוה ולהשאיר רק את 0x64?

אפשר להשתמש ב-AND Mask! הפעולה AND הביטית יכולה לעבוד כמו תבנית, מה שמאפשר לחלק מהמספר לעבור ולהאפס את החלקים האחרים. נעשה AND ביטי עם המספר 0xff, שהוא 8 ביטים שהוגדרו ל-1, וכל הביטים שמעל ה-8 הראשונים הם בעלי ערך 0.

0x00c63364

0x000000ff &

0x00000064

היי! 0x64 הוא הבית מכתובת ה-IP שרצינו! רואים איך איפסנו את כל מה שהיה ב-0 בסכמת המסכה והחלקים ש-1 איפשרו לערך לעבור?

עכשיו אפשר לחלץ את המספרים:

```
0xc633640a >> 24) & 0xff == 0x000000c6 == 0xc6)
(0xc633640a >> 16) & 0xff == 0x00000033 == 0x33
(0xc633640a >> 8) & 0xff == 0x00000064 == 0x64
(0xc633640a >> 0) & 0xff == 0x0000000a == 0x0a
```

וזהו! אלה הם הבתים האישיים של כתובת ה-IP.

כדי להגיע מנקודות ומספרים, אפשר להשתמש ב-f-string או ב-join() ב-Python.

17.4 חידוד מושגי משנה ורשת

זכרו שכתובות IP מחולקות לשני חלקים: מספר הרשת ומספר המחשב. חלק מהביטים בצד השמאלי של כתובת ה-IP הם מספר הרשת, והביטים הנותרים בצד הימני הם מספר המחשב.

הנה דוגמה שבה 24 הביטים השמאליים (3 בתיים) הם מספר הרשת וה-8 הביטים הימניים (1 בית) הם מספר המחשב.

רשת | מחשב

|
198.51.100.10

זאת מייצגת מחשב 10 על הרשת 198.51.100.0. (אנחנו מחליפים את ביטי המחשב ב-0 כשמדברים על מספר הרשת).

אבל אמרתי למעלה באופן חד-צדדי ש-24 הביטים של הרשת בכתובת ה-IP. זה לא היה כל כך קצר. לכן הומצא ייצוג הסלאש.

198.51.100.0/24 24 ביטים משמשים עבור הרשת
או שאפשר להשתמש בזה עם כתובת IP:

198.51.100.10/24 מחשב 10 על הרשת 198.51.100.0
בואו ננסה את זה עם 16 ביטים לרשת:

10.121.2.17/16 מחשב 2.17 על הרשת 10.121.0.0
הבנתם?

בדוגמאות האלה השתמשנו בכפולות של 8 כך שהן יהיו תואמות לגבול בית, אבל אין סיבה שלא תוכל להיות לכם תת-רשת עם חלק מבית שמפוצל:

10.121.2.68/28 מחשב 4 על הרשת 10.121.2.64
אם לא הבנתם מאיפה ה-4 וה-64 הגיעו בדוגמה הקודמת, נסו לכתוב את הבתים בבינארי!

17.5 חידוד מושגי מסכת משנה

מה זה מסכת משנה? מדובר בריצה של ביטים עם ערך 1 בבינארי שמעידה איזה חלק בכתובת ה-IP שייך לרשת. אחריה יש ריצה של ביטים עם ערך 0 בבינארי שמעידה על החלק שייך למחשב.

השתמשו בזה כדי לקבוע איזו רשת שייכת לכתובת IP או איזה חלק מכתובת ה-IP מייצג את מספר המחשב.

בואו נצייר את זה בבינארי. נשתמש בכתובת ה-IP הזאת כדוגמה:

198.51.100.10/24 מחשב 10 על הרשת 198.51.100.0
נתחיל בהמרה לבינארי. (יש כאן רמז שמסכת המשנה היא מסכת AND ביטית!)

198.51.100.10 11000110.00110011.01100100.00001010

עכשיו, מעליה, נצייר ריצה של 24 ים (שהם למעשה האזורים של הרשת):
11111111.11111111.11111111.00000000

נכפול את המידע:

כתובת ה-198.51.100.10 IP מסווגת כך על ידי המסכה 255.255.255.0 (למעשה כתובת ה-IP הרשת היא 198.51.100.0).

17.6 מציאת מסכת משנה מתוך ייצוג סלאש

איך כל אחד יכול לפתח מערכת רשת משלו?

המערכת הזו משתמשת בכתובת שמחולקת לפי תת-רשתות. כדי להקים את הרשתות, תצטרך להתאים את מסכת המשנה.

יש לך כתובת כמו "198.51.100.0/24" אבל אל תדאג, זה סיפור קל למצוא את הייצוג.

• הגידול כלפי מעלה: 0.1.0.0, .