מערכות ספירה · בינארי, הקסדצימלי ואוקטלי · הכתובת בסיביות

שבוע 4 · 08/10–04/10 · **לופו משה**

נתבים ומתגים לא מכירים 192. הם מכירים 11000000. כל מה שנעשה בחודשיים הקרובים — מסכות, תת-רשתות, `VLSM`, `ACL` — נשען על היכולת לעבור בין עשרוני לבינארי בלי לחשוב. היום גם הקסדצימלי (`MAC`, `IPv6`) ואוקטלי (כי הבחינה שואלת).

1. בינארי — שמונה סיביות, שמונה משקלים

כל אוקטט בכתובת `IPv4` הוא 8 סיביות. לכל סיבית משקל, מ-128 בשמאל עד 1 בימין. הערך העשרוני הוא סכום המשקלים של הסיביות הדלוקות:

```
128   64   32   16    8    4    2    1
 1    1    0    0    0    0    0    0    = 128+64      = 192
 1    0    1    0    1    0    0    0    = 128+32+8    = 168
 0    0    0    0    1    0    1    0    = 8+2         = 10
 0    0    0    0    1    0    1    1    = 8+2+1       = 11

192.168.10.11 = 11000000.10101000.00001010.00001011
```

טבלת המשקלים היא הכלי היחיד שצריך. בבחינה אין מחשבון שממיר בסיסים — הטבלה חייבת להיות בראש.

שיטה	איך	מתאימה ל...
חיסור מלמעלה	האם 128 נכנס? כן → 1, מחסירים. האם 64? ...	המרה מהירה בראש
חלוקה ב-2	מחלקים ב-2, רושמים שארית, קוראים מהסוף	אימות, מספרים כלשהם

טיפ · לאמת תמיד בכיוון ההפוך: המרתם 168 → 10101000? חברו חזרה: 128+32+8 = 168 ✓. שתי דקות אימות חוסכות שאלה בבחינה.

2. הקסדצימלי — 4 סיביות בספרה אחת

16 ספרות: 0–9 ואז A–F (10–15). כל ספרה הקסדצימלית = בדיוק 4 סיביות, ולכן ההמרה לבינארי היא החלפה ישירה, בלי חישוב.

```
hex  0    1    2    3    4    5    6    7
bin  0000 0001 0010 0011 0100 0101 0110 0111
hex  8    9    A    B    C    D    E    F
bin  1000 1001 1010 1011 1100 1101 1110 1111

0xAB  =  1010 1011  =  171
MAC 00:1B:44:11:3A:B7  =  12 hex digits  =  48 bits
IPv6 2001:0DB8::1      =  32 hex digits  =  128 bits
```

לכן כתובת `MAC` נכתבת ב-12 ספרות הקסדצימליות, ו-`IPv6` ב-32. הקסדצימלי הוא פשוט דרך קצרה לכתוב בינארי.

הסימון 0x (או h בסוף) אומר 'זה הקסדצימלי': 0x0800 בשדה `Type` של מסגרת `Ethernet` = `IPv4`.

3. אוקטלי — 3 סיביות בספרה, לבחינה

8 ספרות: 0–7. כל ספרה = 3 סיביות. הבחינה (2023 ש'1ב) נתנה מספר הקסדצימלי וביקשה בינארי **ואוקטלי**. הדרך הבטוחה: הקסה → בינארי → קיבוץ לשלשות **מימין** → אוקטלי.

```
0xAB
hex -> bin :  A=1010  B=1011   ->  10101011
group by 3 from the RIGHT:      10 | 101 | 011
pad the left group with zeros:  010 | 101 | 011
bin -> oct :  2   5   3         ->  253 (octal)

check via decimal: 0xAB = 171 ;  253 octal = 2*64+5*8+3 = 171  OK
```

הקיבוץ מימין הוא הטעות הנפוצה. מימין!

⚠ **שימו לב ·** בבחינה שואלים 'כתבו בבסיס בינארי ובבסיס אוקטלי'. לא לעבור דרך עשרוני — זה איטי ומועד לטעויות. הקסה↔בינארי↔אוקטלי הוא החלפה של קבוצות סיביות.

4. `AND` — מה שהמסכה עושה

המסכה היא רצף של אחדות (חלק הרשת) ואז אפסים (חלק המארח). `AND` בין הכתובת למסכה משאיר את סיביות הרשת ומאפס את השאר. התוצאה: כתובת הרשת.

```
address   192.168.10.11   = 11000000.10101000.00001010.00001011
mask      255.255.255.0   = 11111111.11111111.11111111.00000000
AND                       = 11000000.10101000.00001010.00000000
network   192.168.10.0

mask 255.255.255.192 = ...11000000  ->  /26  (24 + 2 network bits)
```

1 `AND 1` = 1, כל השאר 0. אורך הקידומת (/26) הוא מספר האחדות במסכה — סופרים בבינארי.

דוגמה פתורה

השאלה

נתונה כתובת 172.16.5.70 עם מסכה 255.255.255.192. כתבו את הכתובת והמסכה בבינארי, מצאו את כתובת הרשת, ורשמו את המסכה כאורך קידומת.

הפתרון צעד אחר צעד

1. **172** = 128+32+8+4 = 10101100 · **16** = 00010000 · **5** = 00000101 · **70** = 64+4+2 = 01000110.
2. **192** = 128+64 = 11000000. המסכה: 11111111.11111111.11111111.11000000.
3. **על האוקטט האחרון**: 01000110 `AND 11000000` = 01000000 = 64. שלושת הראשונים נשארים (`AND` עם 11111111).
4. **כתובת הרשת**: 172.16.5.64.
5. **אורך הקידומת**: 8+8+8+2 = 26 אחדות → /26.

תשובה · 172.16.5.64/26. הבלוק הוא 64 כתובות (64.–127.), כי 2 סיביות נלקחו למספור תת-רשתות ו-6 נשארו למארחים.

תרגילים — פתרו לבד לפני שאתם בודקים בכיתה

1. המירו לבינארי: 200 · 45 · 255 · 129.
2. המירו לעשרוני: 11100000 · 00111100 · 10000000.
3. 0x3F בבינארי, בעשרוני ובאוקטלי?
4. כתובת `MAC` בת 12 ספרות הקסדצימליות — כמה סיביות?
5. מהי כתובת הרשת של 10.1.1.130 עם מסכה 255.255.255.128?
6. כמה אחדות במסכה 255.255.240.0? כתבו כ-/n.

7. למה 11001100 אינה מסכה חוקית?

8. השדה `Type` במסגרת מכיל 0x0800. מה זה בבינארי ומה המשמעות?

דף עבודה — דף תרגול 4 — עשרים המרות

שם: _____ תאריך: _____ חוליה: _____

המטרה: להגיע למצב שבו טבלת המשקלים בראש והמרה לוקחת פחות מדקה.

מהלך — סמנו כל שלב שהושלם

- ☐ רשמו את טבלת המשקלים בראש הדף: 128 64 32 16 8 4 2 1.
- ☐ המירו לבינארי (בשיטת החיסור): 172 · 16 · 5 · 70 · 224 · 193 · 100 · 7 · 250 · 33.
- ☐ אמתו כל תוצאה בחיבור חזרה.
- ☐ המירו הקסה → בינארי: 0x0A · 0xFF · 0x7E · 0xC0 · 0x1F.
- ☐ המירו בינארי → אוקטלי (קיבוץ מימין!): 10101011 · 11000000 · 00111111.
- ☐ `AND`: 192.168.20.77 עם 255.255.255.224 · 10.0.5.200 עם 255.255.255.128.
- ☐ החליפו דף עם שכן/ה ובדקו זה את זו. תקנו בעט אחר.

רישום התוצאות

#	מה למלא	חישוב · הערות · ראיות (פלט פקודה)
1	עשרוני→בינארי (10): _____	_____
2	הקסה→בינארי (5): _____	_____
3	בינארי→אוקטלי (3): _____	_____
4	`AND 1:` רשת _____ / _____ · `AND 2:` רשת _____ / _____	_____
5	טעויות שנמצאו בבדיקה: _____	_____

רפלקציה — שתי שורות

מה היה הכי קשה, ומה פתר את זה?

_____

מה הייתי עושה אחרת בפעם הבאה?

_____

אישור המורה: ☐ שאלון היחידה ב-Moodle (100) ☐ שיעורי הבית (30) · **XP** _____ · תג: ☐

מפענח/ת הבסיסים · חתימה _____

כרטיס יציאה — 3 שאלות, 5 דקות, נמסר ביציאה

1. 192 בבינארי?

2. כמה סיביות בספרה הקסדצימלית? ובאוקטלית?

3. לאן מקבצים לאוקטלי — מימין או משמאל?

עוד לבדיקה עצמית — ענו בעל פה, בלי להסתכל אחורה

1. מה נותן `AND` בין כתובת למסכה?

מילון מונחים

עברית	אנגלית	הסבר קצר
סיבית	`Bit`	0 או 1 — היחידה הקטנה ביותר
אוקטט	`Octet`	8 סיביות · ערך 0–255 · רבע כתובת `IPv4`
בינארי	`Binary`	בסיס 2 — משקלים 1…128
הקסדצימלי	`Hexadecimal`	בסיס 16 · 0–9, A–F · 4 סיביות לספרה
אוקטלי	`Octal`	בסיס 8 · 0–7 · 3 סיביות לספרה
אורך קידומת	`Prefix Length`	מספר האחדות במסכה, /n
לוגי AND	`Logical AND`	1 רק כששתי הסיביות 1

להעמקה

בספר הקורס

- **ITN 5** — Number Systems · עמוד 176

קבצים שנעבוד איתם

- ITN_Module_5.pptx
- ITN_Module_11.pptx
- Subnetting IP Networks.docx
- טבלת חלוקה.xlsx