ארגון המחשב ומערכות הפעלה

אביב תשפ"ד

תרגול 1 – לוגיקה פסוקית ומעברי בסיסים

קצת פרטים

- 5-6 תרגילי בית תקפים
- אפשר להגיש בזוגות או יחידים

קצת על הקורס

- המטרה של הקורס - ללמוד איך מערכות הפעלה עובדות מנקודת המבט של המתכנת
- שפת התכנות תהיה C- אם אתם לא זוכרים, זה הזמן לחזור על החומר!
- ספר הקורס -
  - חלק ראשון- Computer systems - a programmer’s perspective / Bryant and O’hallaron
  - חלק שני- Operating System Concepts / Avi Silberschatz, Peter Baer Galvin, Greg Gagne

למה C?

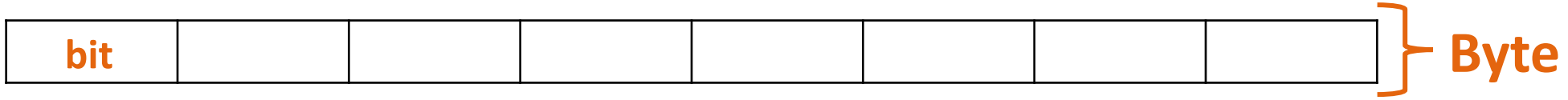
- אם אנחנו רוצים לשלוט על המשאבים של המחשב, אנחנו צריכים שפה שתהיה כמה שיותר קרובה לשפת מכונה.
- כדי לקבל מושג על מערכות הפעלה נעקוב אחרי התכנית הבסיסית hello מזמן היצירה שלה ע"י המתכנת, ההרצה במערכת, הדפסת ההודעה והסיום.

```c
#include <stdio.h>

int main()
{
    printf("hello, world\n");
}
```

תכנית ה- hello מתחילה את חייה בקובץ מקור (source file) שהמתכנת יוצר עם עורך (editor) ושומר את זה בקובץ text שנקרא hello.c

קובץ המקור מורכב מסיביות (bits) שיכולות להיות 0 או 1, ומאורגנות בקבוצות של 8 - נקרא בית (byte).
כל byte מייצג תו (character) בתכנית.



נרצה להבין איך התכנית hello.c מיוצגת בעזרת סיביות (bits)

כדי להבין - נתחיל בלעבור על לוגיקה פסוקית ומעברי בסיסים

לוגיקה פסוקית

- בכל שפה- יחידת התוכן הבסיסית היא המשפט
- במתמטיקה- משפט הוא טענה חד משמעית אמיתית, שיש לה הוכחה
- ניתן לסווג משפטים לכאלו שנכונים וכאלו שלא נכונים
- פסוק: הצהרה שמציינת עובדה שיכולה להיות נכונה או לא נכונה

פסוק

נסמן פסוק באותיות a,b,c...
כל פסוק מורכב מ:

משתנים בוליאניים

יכולים להיות 0 (false) או 1 (true)

קשרים לוגיים

- פעולות שמפעילים על המשתנים
- מאפשרים לשלב כמה פסוקים פשוטים לפסוק מורכב

קשרים לוגיים

- ~ NOT
- & AND
- | OR
- ^ XOR
- → IMPLIES
- ↔ EQUIVALENT TO

כדי להבין איך קשרים לוגיים עובדים- נבין מהי טבלת אמת

זוהי הדרך להציג את כל ההשמות האפשריות לפסוקיות בוליאניות

טבלת אמת מורכבת מ:

1. משתנים בוליאניים- פסוקיות, קשרים
2. שלבי ביניים- אם צריך
3. ביטוי סופי- התשובה

$\sim$ NOT

נגדיר את הפסוק a – בבוקר ירד גשם

הקשר הלוגי NOT על הפסוק a יחזיר את השלילה של הפסוק a:

בבוקר לא ירד גשם. נסמן את הפסוק כך: $\sim a$

טבלת האמת של הקשר NOT:

a	$\sim a$
1	0
0	1

& AND

- נגדיר את **<u>הפסוקים הפשוטים</u>** הבאים:

a: יש חלב בפינת קפה

b: יש קפה בפינת קפה

- נגדיר את **<u>הפסוק המורכב</u>** הבא:

c: יש חלב בפינת קפה וגם יש קפה בפינת קפה

- הפסוק c יהיה פסוק אמת באשר שני תתי הפסוקים a ו-b יהיו אמת
  כלומר- c=1 אמ"מ a=1 וגם b=1

טבלת האמת של הקשר AND

a	b	$a\&b\ (c)$
1	0	0
1	1	1
0	0	0
0	1	0

| OR

- נגדיר את **<u>הפסוק המורכב</u>** הבא:

d: יש חלב בפינת קפה או שיש קפה בפינת קפה

- הפסוק d יהיה פסוק אמת באשר לפחות אחד מתתי הפסוקים a ו-b יהיו אמת

טבלת האמת של הקשר OR

a	b	$a\|b$
1	0	1
1	1	1
0	0	0
0	1	1

^ XOR

- פעולת XOR תהיה אמת רק באשר 2 הפסוקים שונים זה מזה
- A או B אבל לא A וגם B
  - בלומר, הפעולה תוציא אמת רק באשר $A \neq B$

טבלת האמת של הקשר XOR

a	b	$a \hat{} b$
1	0	1
1	1	0
0	0	0
0	1	1

→ IMPLIES

- נגדיר את הפסוקים הפשוטים הבאים:

a - המלכה מתה

b - הבן של המלכה מחליף אותה

- נגדיר את הפסוק המורכב הבא:

e - אם המלכה מתה אז הבן שלה מחליף אותה

- נגיד שהפסוק e אמת באשר מתקיים:
  - אם המלכה מתה והבן שלה מחליף אותה - התקיים התנאי והתקיימה המסקנה (**קיום באופן מלא**)
  - אם המלכה לא מתה - לא התקיים התנאי, אין שום סיבה שנקיים את המסקנה (**קיום באופן ריק**)

טבלת האמת של הקשר IMPLIES

a	b	$a \rightarrow b$
1	0	0
1	1	1
0	0	1
0	1	1

↔ EQUIVALENT TO

- נגדיר את הפסוקים הפשוטים הבאים:

a - אני ארוויח בסף

b - אני אעבוד קשה

- נגדיר את הפסוק המורכב הבא:

f - אני ארוויח בסף אמ"מ אני אעבוד קשה

- הפסוק f הוא אמת רק אם שני תתי הפסוקים שלו זהים
  - ההופכי ל-XOR
  - בלומר, יוציא אמת רק כאשר $a = b$

טבלת האמת של הקשר EQUIVALENT TO

a	b	$a \leftrightarrow b$
1	1	1
0	0	1
1	0	0
0	1	0

תרגיל

הציעו 2 דרכים לייצג את הקשר XOR באמצעות הקשרים: NOT, AND, OR

פתרון

הציעו 2 דרכים לייצג את הקשר XOR באמצעות הקשרים: NOT, AND, OR

<u>פתרון</u>:

1. $a \oplus b = (\sim a \& b) | (a \& \sim b)$
2. $a \oplus b = (a | b) \ \& \sim (a \& b)$

התמרות בין אופרטורים

- חוקי דה מורגן

$$a\&b \equiv \sim(\sim a|\sim b)$$
$$a|b \equiv \sim(\sim a\&\sim b)$$

- רוצים לייצג את הקשר XOR ע"י AND, OR, NOT
  - לפי טבלת האמת של XOR
  - בעזרת חוקי דה מורגן

אחרי שהבנו מה זו לוגיקה פסוקית- עוברים לשלב הבא- מעברי בסיסים

מעברי בסיסים

העולם שאנחנו מכירים עובד בבסיס 10- הספרות 0,1,2,3,4,5,6,7,8,9

בפועל אפשר לעבוד באיזה בסיס שרוצים אבל

- קשה יותר לחשב
- קשה יותר לאגור מידע
- קשה יותר להעביר מידע

רוב המחשבים שאנחנו מכירים עובדים בבסיס הקסדצימלי- בסיס 16

לפני שנבין איך עובדים בבסיס 16, נבין איך עוברים לכל מיני בסיסים

נגדיר את המקומות של הספרות במספר כלשהו

$$\underbrace{1}_{-}\ \underbrace{0}_{-}\ .\ \underbrace{-1}_{-}\ \underbrace{-2}_{-}\ \ldots$$

מעבר מבסיס בלשהו לבסיס 10

נייצג את המספר 241.31 בבסיס 5 לבסיס 10

$$(241.31)_5 = 2 * 5^2 + 4 * 5^1 + 1 * 5^0 + 3 * 5^{-1} + 1 * 5^{-2} = (71.64)_{10}$$

241.31 בבסיס 5 שווה ל-71.64 בבסיס 10

נייצג את המספר 11.01 בבסיס 2 לבסיס 10

$$(11.01)_2 = 1 * 2^1 + 1 * 2^0 + 0 * 2^{-1} + 1 * 2^{-2} = (3.25)_{10}$$

11.01 בבסיס 2 שווה ל-3.25 בבסיס 10

עכשיו נבין איך עוברים מבסיס 10 לבסיסים אחרים

אלגוריתם המעביר מספר שלם k לבסיס B

```
string str=“ ”;
while (k != 0)
{
    int digit=k%B;
    str=digit_to_char(digit)+str;
    k=k/B;
}
return str;
```

שארית החלוקה של k מ-B

נפטרים מהספרה הימנית של k

נעביר את המספר 125 לבסיס 8
הולכים לפי האלגוריתם:
k=125, B=8

השלם k	תוצאת החלוקה ב-B	שארית החלוקה ב-B	str
125			

נעביר את המספר 125 לבסיס 8
הולכים לפי האלגוריתם:
k=125, B=8

K	K//B	K%B	str
125	15	5	5
15	1	7	75
1	0	1	175

נוודא שעשינו נכון

$$(175)_8 = 1 * 8^2 + 7 * 8^1 + 5 * 8^0 = (125)_{10}$$

ייצוג בבסיס 16

Decimal (Base 10)	Binary (Base 2)	Hexadecimal (Base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

- רוב המחשבים עובדים בבסיס 16
- בסיס הקסדצימלי מיוצג ע"י:
  - הספרות 0,1,2,3,4,5,6,7,8,9
  - האותיות A,B,C,D,E,F

דוגמה:

$$(D37B)_{16}$$
$$= 13 * 16^3 + 3 * 16^2 + 7 * 16^1 + 11 * 16^0$$
$$= (54{,}139)_{10}$$

כללי אצבע למעברים בין בסיסים

- דצימלי לכל בסיס אחר: חישוב באמצעות אלגוריתם חלוקה
- כל בסיס לדצימלי: סכימת המכפלות של הבסיס לפי מיקומים
- הקסה לבינארי / בינארי להקסה:

כל ספרה הקסהדצימלית תופסת 4 ביטים. לכן אפשר להמיר ישירות

Hex digit	0	1	2	3	4	5	6	7
Decimal value	0	1	2	3	4	5	6	7
Binary value	0000	0001	0010	0011	0100	0101	0110	0111
Hex digit	8	9	A	B	C	D	E	F
Decimal value	8	9	10	11	12	13	14	15
Binary value	1000	1001	1010	1011	1100	1101	1110	1111

תרגיל

Decimal	Binary	Hexadecimal
0	0000 0000	00
55	0011 0111	37
136		
	1111 0011	
		52
172		
		E7
	1010 0111	
	0011 1110	
		BC

פתרון

Decimal	Binary	Hexadecimal
0	0000 0000	00
55	0011 0111	37
136	1000 1000	88
243	1111 0011	F3
82	0101 0010	52
172	1010 1100	AC
231	1110 0111	E7
167	1010 0111	A7
62	0011 1110	3E
188	1011 1100	BC