

# **INF1: Informatika alapismeretek**

*4. lecke: Az információ mennyisége*

## 1. A lecke tartalma

- A bit és a byte
- Informatikai számrendszerek bemutatása, viszonyuk a tízes számrendszerrel és egymással

## 2. A bit

Az információ, illetve az adat mennyiségét ha meg akarjuk határozni, akkor a legkisebb használható mértékegység a **bit** lesz, de az információmennyiség alap mértékegysége már a **byte**. Nézzük meg ezek közül először, mit jelent a *bit*.

Egy **bit** az információtartalma egy *eldönthető kérdésre adott válasznak*, ha az *igen és a nem* válasz valószínűsége megegyezik. A bit értéke 0 vagy 1 lehet, más jelölés szerint *hamis/igaz, ki/be, igen/nem* lehet. Ez az információmennyiség legkisebb mértékegysége. Jelölése a „bit” vagy a kis „b” betűvel lehetséges.

Tehát a bit két azonos esélyű lehetőség között tud különbséget tenni. Az egyik lehetőséget 0, a másikat 1 értékkel jelölve szoktuk használni. Amennyiben két bit áll rendelkezésre, akkor könnyen belátható, hogy 4 azonos valószínűségű lehetőség között lehet különbséget tenni. Tovább gondolva mindezt, belátható, hogy  $n$  bit  $2^n$  azonos esélyű lehetőség között képes különbséget tenni.

Hogy miért fontos az azonos esély hangsúlyozása, azt könnyen beláthatjuk a következők alapján:

Egy kalapban piros és kék golyók vannak. Kék golyó kettő van, piros golyó pedig 6. Hány bit szükséges ahhoz, hogy megadjuk, milyen színű a kalapból kihúzott golyó?

Ha a fenti példában nem vesszük figyelembe, hogy nem azonos a piros és a kék golyók száma, akkor egy bit elég a két szín megkülönböztetésére. Azonban abban a pillanatban, hogy az is lényeges, hogy melyik golyószínnek mekkora a valószínűsége, már változik a helyzet. Ekkor, mivel 1:3 esélye van az egyik, illetve a másik színnek, valójában már két bit szükséges, amelyeken ábrázolva például 00 érték jelenti a kék színt, míg a 01, 10 és 11 értékek egyaránt a pirost. Ha pedig arra lennénk kíváncsiak, hogy pontosan melyik golyót húzták ki, akkor a nyolc golyót három biten lehetne megkülönböztetni, mind a nyolc golyónak egy-egy számhármast adva.

Mivel a bitek száma és a megkülönböztethető információk száma között a kettő hatványok adnak kapcsolatot, ezért az első néhány kettő hatvány ismerete a számítástechnikában különösen fontos. Íme az első 11 kettő hatvány felsorolva:

1.  $2^0 = 1$
2.  $2^1 = 2$
3.  $2^2 = 4$
4.  $2^3 = 8$
5.  $2^4 = 16$
6.  $2^5 = 32$
7.  $2^6 = 64$
8.  $2^7 = 128$
9.  $2^8 = 256$
10.  $2^9 = 512$
11.  $2^{10} = 1024$

A fenti felsorolásban két érték a fontossága miatt ki van emelve, ezekre még visszatérünk.

Az adatátvitel sebességét általában a másodpercekben továbbított bitek számában szokás mérni, aminek mértékegysége b/sec, azaz bit/másodperc. Természetesen az SI prefixumokkal együtt lehet ezt is használni (pl. kbit/sec, vagy Mb/sec).

### 3. A kettes számrendszer

A bit szó a binary digit rövidítéséből származik, ami arra utal, hogy a 0/1 értékek a kettes számrendszer egy-egy helyiértékén megjelenő számoknak feleltethetőek meg. Ezáltal tehát a több bitből álló kifejezések valójában mint kettes számrendszerbeli számok értelmezhetőek. Ez az értelmezés egy lehetőséget jelent arra, hogy a lehetséges eseteket megsorszámozva egy ábrázolási módot kapjunk.

A ma használatos számítógépek túlnyomó többsége – gyakorlatilag az összes digitális számítógépünk – ez alapján kettes, azaz bináris számrendszerben tárol mindent. Ennek egyik előnye, hogy egy bit ábrázolása akár elektronikusan, akár optikai úton két fizikailag jól elkülöníthető értékkel reprezentálható. Elektronikus esetben kézenfekvő két egymástól távoli feszültség szint (például 0 V és 5 V) használata, optikai esetben a nincs fény/van fény lehet ennek megfelelője.

A kettes számrendszerrel egy másik témakör majd bővebben foglalkozik, itt most elégedjünk meg annyival, hogy ha a biteket egymás mellé tesszük, az egy kettes számrendszerben megadott számnak tekinthető.

Egy számítógép mindig valahány bitet kezel együtt. Ez a bitszám általában a számítógépben állandó. Így kialakult a szó elnevezés a számítógépben együtt kezelt bitsorozatra. Eleinte géptípusonként változott a szóhossz, azaz a szóban található bitek száma. Tipikusnak mondható eleinte a 7, 8, 16 bites szóhossz, de létezett olyan számítógép, amely 60 bites szavakkal dolgozott.

## 4. A byte

Mivel az angol nyelvben használt karakterek (betűk, számok, írásjelek) száma néhány speciális vezérlőjellel együtt 7 biten eltárolható volt, az első szabványos szóhossz az adatátvitelben 7+1 bit volt. 7 biten történt magának a karakternek a továbbítása, a 8. bit pedig egy *ellenőrző bit* volt, ami arra szolgált, hogy a *dekódoló* ellenőrizhesse az érkezett jel helyességét. Erre a 7+1 bitre emlékeztet az ASCII kódrendszer (American Standard Code for Information Interchange), amely 128 különböző karakter kódját egységesítette. Később az európai nyelvek további betűinek tárolására ezt bővítették ki a 8. bit felhasználásával 8 bites kódokra.

A bitek nyolcasával történő csoportosításából alakult ki a „*by eight*” rövidítéseként a **byte**, amely az **információmennyiség alap mértékegységének** számít, és amely tehát 8 bitet jelent együtt. Másik elterjedt elnevezés erre a 8-as csoportosításra az **octet**, amely szintén bit-nyolcast jelent.

A számítógépek tipikusan a 8 többszörösét használják szóhosszként, így szokás arról is beszélni, hogy a szóhossz hány byte méretű.

*A magyar helyesírási szabályzat szerint a byte szót „bájt” formában kellene írni. Azonban ez valójában egy mozaikszó, amelyet a „by eight” rövidítéseként hoztak létre (egészen pontosan Werner Buchholz használta először 1956 júliusában). Emiatt a helyes írásmód az eredeti „byte”, hiába gondolják másként a nyelvészeink (vagy azok egy része).*

## 5. Váltószámok

A bit, illetve a byte többszöröseinek megadásában elég komoly keveredést okoz, hogy egyes adattárolóknál nincs értelme az SI prefixumoknál használt 1000-szeres szorzónak. Ez az eset azokban az egyes byte-onként megcímezhető memóriáknál, amelyek címeit is kettős számrendszerben tárolja a számítógép. Ilyen memóriákat célszerű úgy legyártani, hogy a megcímezhető szavak száma kettőhatvány legyen. Ezért ezek kapacitásánál a váltószámok is célszerűen kettőhatványok.

Emiatt alakult úgy, hogy a kettős tizedik hatványa, azaz 1024 a váltószám, amelyet eredetileg szintén az SI ezerszerest jelentő váltószámaival azonos módon jelöltek.

Ugyanakkor a bináris címzést nem igénylő háttértárak esetében az SI-féle ezerszeres váltószámokat használják a gyártók. Ebből a kettősségből adódott egy jókora káosz, amit azzal próbálnak feloldani, hogy a bináris váltószámokra egy módosított jelölést vezettek be – ami szépen lassan, de kezd elterjedni.

Ez alapján a következők igazak:

- 1 kbyte = 1000 byte, de 1 kabyte = 1024 byte
- 1 Mbyte = 1000 kbyte, de 1 Mibyte = 1024 kabyte
- 1 Gbyte = 1000 Mbyte, de 1 Gibyte = 1024 Mibyte

Kimondva viszont mind az SI, mind a módosított (i-vel megtoldott) prefixumot ugyanúgy mondjuk, ami félreértésekre vezethet, mint ahogyan az is, hogy egyes programok nem szabványos módon keverik a kétféle számítást. Vigyázat: hivatalosan nem keverhető a két jelölés egymással, tehát ha byte-ról kbyte-ra váltottunk, akkor már nem válthatunk tovább 1024-gyel számolva, ahogyan fordítva sem!

A byte rövidíthető „B” formában is.

## 6. További számrendszerek

A kettes számrendszer mellett még két számrendszert szoktak használni a számítástechnikában. Ezek a nyolcas és a tizenhatos számrendszer. Ezek valójában csak reprezentatív számrendszerek, valójában ezekben számolás nem történik. A használatuk célja mindössze az, hogy a bináris értékeket rövidebben lehessen felírni.

Az 1. táblázat mutatja az átváltást a kettes, a nyolcas és a tizenhatos számrendszer között. Az első oszlop az értékeket mutatja tízes számrendszerben.

1. Táblázat: Átváltó táblázat a decimális, a bináris, az oktális és a hexadecimális számok között

Tízes (decimális)	Kettes (bináris)	Nyolcas (oktális)	Tizenhatos (hexadecimális)
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Két dolgot célszerű megfigyelni a táblázatban: Az első, hogy egy oktális számjegy kettes számrendszerben három jegyű, míg egy hexadecimális számjegy kettes számrendszerben négy jegyű. Ennek egyszerű az oka: az oktális számrendszer alapszáma, a nyolc a kettő harmadik hatványa, míg a hexadecimális számrendszer alapszáma a tizenhat éppen a negyedik hatványa a kettőnek. Így valójában az átírás csupán a bináris szám rövidebb felírását jelenti, de valódi átszámításra nincs szükség.

A másik megfigyelhető, hogy a tizenhatos számrendszerben a nem létező utolsó hat számjegyet az ábécé első hat betűjével pótoljuk. Ezek írhatók kis- és nagybetűként is, de egy dokumentumon – de egy számon belül feltétlenül – illik ragaszkodni vagy a nagybetűkhöz vagy a kisbetűkhöz. Mint látható, egy byte tartalma felírható két hexadecimális számjeggyel.

## 7. További internetes források a témában

- [http://en.wikipedia.org/wiki/Octet\\_%28computing%29](http://en.wikipedia.org/wiki/Octet_%28computing%29)
- <http://en.wikipedia.org/wiki/Bit>
- [http://en.wikipedia.org/wiki/Units\\_of\\_information](http://en.wikipedia.org/wiki/Units_of_information)
- [http://en.wikipedia.org/wiki/Binary\\_prefix](http://en.wikipedia.org/wiki/Binary_prefix)

