

Кодирање података коришћењем бинарног бројевног система

Рачунари све податке записују коришћењем бројева. Међутим, из техничких разлога бројеве у меморији рачунара није могуће записати на уобичајен начин, помоћу десет различитих цифара (тј. уз помоћ **декадног система**). Меморија рачунара је на неки начин низ елемената који може бити у једном од два различита стања, та два стања обележавамо различитим цифрама 0 и 1, тј. **бинарним системом записа бројева**.

Бројевни системи и бинарни бројеви

Бинарни бројевни систем

01

Декадни бројевни систем

0123456789

Октални бројевни систем

01234567

Хексадекадни бројевни систем

0123456789ABCDEF

Декадни бројевни систем. Овај бројни систем има 10 цифара (од 0 до 9). Основа овог система је 10.

Декадни бројни систем је тежински – то значи да свака цифра у декадном броју има одређену тежину.

Пример броја 4594 у тежинском облику

$$4 \cdot 10^3 + 5 \cdot 10^2 + 9 \cdot 10^1 + 4 \cdot 10^0 = 4594$$

Прва цифра 4 и последња цифра 4 немају исту тежину!!! Прва цифра има тежину 4000 а последња има тежину (вредност) 4.

Бинарни бројевни систем. Овај бројни систем има само две цифре (два стања) цифру "0" (у математици и програмирању означавамо као FALSE) и цифру "1" (у математици и програмирању означавамо као TRUE). Основа овог бројног система је 2.


Зашто је овај бројни систем погодан за представљање података у рачунару? Зато што се са једноставним колима (прекидачима) може реализовати пренос/представљање података.

Стање логичке нуле би представљало отворен прекидач (нема струје), а стање логичке јединице представљено је затвореним прекидачем (има струје).

Конверзија декадног броја у бинарни

Декадни број делимо са 2 (основа бинарног система) и поред пишемо остатак при дељењу, а испод резултат дељења (целобројно дељење)

| | | |
|------|-----|---|
| 25 : | 2 = | 1 |
| 12 | 2 | 0 |
| 6 | 2 | 0 |
| 3 | 2 | 1 |
| 1 | 2 | 1 |



Делимо до краја када нам остане 1 при дељењу са два има остатак 1 а цео део 0 и ту је крај!

Запис бинарног броја иде од дна табеле ка врху

$$25_{(10)} = 11001_{(2)}$$

Конверзија бинарног броја у декадни

Бинарни бројевни систем је такође тежинског типа, тако да ћемо користити исту логику коју смо користили када смо представљали декадне бројеве

$$\begin{aligned}1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 &= \\1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 &= \\16 + 8 + 1 &= 25\end{aligned}$$

Тежински распоред прва цифра 1 у овом броју има тежину 16 друга 8, а последња 1.

$$11001_{(2)} = 25_{(10)}$$

Хексадекадни бројевни систем. Овај бројни систем има 16 цифара које иду од 0 до 9, а преостале се представљају словима A(10), B(11), C(12), D(13), E(14) и F(15). Ово је урађено ради лакшег и јаснијег записа. Како је бинарни систем гломазан хексадекадни нам даје могућност краћег записа бинарних цифара. Основа хексадекадног система је 16.

нпр: $9A_{(16)}$ - хекса запис $154_{(10)}$ - декадни(децимални) запис $10011010_{(2)}$ - бинарни запис

Конверзија хексадекадног броја у декадни

Основа хексадекадног бројевног система је 16 и користимо исту логику као до дас представљамо га у тежинском облику

$$\begin{aligned}F1 &= F \cdot 16^1 + 1 \cdot 16^0 = \\15 \cdot 16 + 1 \cdot 1 &= 241\end{aligned} \quad F1_{(16)} = 241_{(10)}$$

Конверзија бинарног броја у хексадекадни

Конверзија бинарног броја у хекса је јако брза и без математике. Групишете бинарне бројеве од краја по четири бинарне цифре. Са четири бинарне цифре се представља једна хекса цифра. У сличају да недостају цифре дописују се нуле.

Пример конвертовати бинарни број 11101011110010 у хекса

$$11101011110010 = 0011 \ 1010 \ 1111 \ 0010 = 3 \ A \ F \ 2$$

$$11101011110010_{(2)} = 3AF2_{(16)}$$

Конверзија хексадекадног броја у бинарни

Свака хекса цифра се представља са четири бинарне цифре

$$FF1 = 1111 \ 1111 \ 0001$$

$$FF1_{(16)} = 111111110001_{(2)}$$

| | | | | | | | | | | | | | | | | |
|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| DEC | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| HEX | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| BIN | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

Октални бројевни систем. Овај бројни систем има 8 цифара које иду од 0 до 7, основа овог бројног система је 8.

Конверзија окталног броја у декадни

$$432_{(8)} = 4 \cdot 8^2 + 3 \cdot 8^1 + 2 \cdot 8^0 = 4 \cdot 64 + 3 \cdot 8 + 2 \cdot 1 = 282_{(10)}$$

Конверзија бинарног броја у октални

Групишете бинарне бројеве од краја по три бинарне цифре. У сличају да недостају цифре дописују се нуле.

Пример конвертовати бинарни број 10110 у октални

$$10110_{(2)} = 010\ 110 = 26_{(8)}$$

Конверзија окталног броја у бинарни

$$347_{(8)} = 011\ 100\ 111 = 011100111_{(2)}$$

Конверзија из декадног у бинарни

$$\begin{array}{l} 26:2 = 13\ (0) \\ 13:2 = 6\ (1) \\ 6:2 = 3\ (0) \\ 3:2 = 1\ (1) \\ 1:2 = 0\ (1) \end{array}$$

$$26_{(10)} = 11010_{(2)}$$

Конверзија из декадног у октални

$$\begin{array}{l} 137:8 = 17\ (1) \\ 17:8 = 2\ (1) \\ 2:8 = 0\ (2) \end{array}$$

$$137_{(10)} = 211_{(8)}$$

Конверзија из декадног у хексадекадни

$$\begin{array}{l} 949:16 = 59\ (5) \\ 59:16 = 3\ (11_{(10)} = B_{(16)}) \\ 3:16 = 0\ (3) \end{array}$$

$$949_{(10)} = 3B5_{(16)}$$

Јединице за мерење количине података

Бит је минимална количина за мерење информације. Бит може регистровати само једну од две вредности 0 и 1. Бајт је већа јединица за мерење количине информација. Бајт се састоји од 8 битова.



ДЕФ: Најмања количина података која се може регистровати је **бит** (једна бинарна цифра). Основна јединица мере количине података је **бајт**.

Веће јединице од бајта су:

$$1\text{ KB} = 1024\text{ B} = 2^{10}\text{ B}$$

$$1\text{ MB} = 1024\text{ KB} = 2^{20}\text{ B}$$

$$1\text{ GB} = 1024\text{ MB} = 2^{30}\text{ B}$$

$$1\text{ TB} = 1024\text{ GB} = 2^{40}\text{ B}$$

У овим јединицама мери се **капацитет** меморије рачунара и величина фајлова (датотека).

Кодирање карактера

У рачунарима се све представља бројевима, па и текст. Текст је у рачунару представљен као низ **карактера**. **Катактери** су сви елементи од којих се састоји текст (слова, цифре, интерпункцијски знаци и специјални знаци). Да би се текст представио бројевима, врши се **кодирање карактера**. Сваком карактеру се придружује јединствен број – његов **код**. Што је прописано међународним стандардима тј. **кодним схемама**.

ASCII код

За представљање катактера (знакова, слова) на рачунару прво се појавио ASCII код (*American Standard Code for Information Interchange*). Са ASCII кодом се приказују сви знаци видљиви, они који имају графички симбол, слова нпр “А” или “а”, интерпункција и специјални знаци “&”, као и “невидљиви” знаци пример је “знак” ENTER или TAB. ASCII код има 256 знакова то је постигнуто са 8 бит-а.

| Бинарно | Декадно | Хекса | Знак у табели |
|-----------|---------|-------|---------------|
| 0100 0001 | 65 | 41 | A |
| 0100 0010 | 66 | 42 | B |
| 0100 0011 | 67 | 43 | C |
| 0100 0100 | 68 | 44 | D |
| 0100 0101 | 69 | 45 | E |
| 0100 0110 | 70 | 46 | F |
| 0100 0111 | 71 | 47 | G |
| 0100 1000 | 72 | 48 | H |
| 0100 1001 | 73 | 49 | I |
| 0100 0000 | 64 | 40 | @ |

Табела ASCII кода

Универзални код (Unikod)

Проблем код ASCII кода је да не може да представи наша слова (ћирилицу и латиницу), као и сва остала регионална писма. Тај проблем је првобитно решен применом универзалног кода **UCS-2** који за представљање свих писама користи 2 бајта. Односно уз помоћ њега можемо да кодирамо (представимо) $2^{16} = 65536$ знакова, што је сасвим довољно да се представе сви карактери различитих писама.

Применом Unikoda је за неки текст потребно дупло више меморије неко коришћењем ASCII кода, јер нам је код кодирања ASCII кодом потребно 8 бита, док коришћењем Unikoda користимо 16 бита за кодирање једног знака. Ово може да буде проблем за пренос података. У већини случајева нама није потребно да користимо више писама (арапски, енглески, руски...). па је овај проблем је решен коришћењем Unikod трансформационе шеме.

Уникод трансформациона шема (UTF -8)

Примена UTF -8 нам даје могућност да поједине знакове кодирамо са један, два или три бајта. UTF -8 користимо у Word-у. Погодан је “Markup” језике HTML, XML... То нам даје могућност да “куцамо” команде и уносимо текст за приказ на матерњем језику, нпр web странице. UTF -8 је подржан од стране неколико стандардних фонтова (Times New Roman, Helvetica...). UTF -8 се такође користи у развоју база података (подаци у бази унети на адекватном језику – нпр. ћирилица).

Кодирање (компресија) слике, звука, видео записа...



Декодирање (декомпресија)

