Računarski sistem, predstavljanje podataka u računaru

predavač: Nadežda Jakšić

# Programiranje programski jezik C

# Računarski sistem

**Hardver**

- ulazni uređaji
- procesor
- unutrašnja, operativna, RAM memorija – u njoj se nalazi program koji se trenutno izvršava
- izlazni uređaji
- spoljne memorije

**Softver**

- sistemski programi
- aplikativni ili korisnički programi

# Podaci u računaru

- memorija se sastoji od velikog broja najčešće elektronskih komponenti koje mogu da memorišu odnosno čuvaju dva stanja (diskretne vrednosti) određene fizičke veličine tj. napona
- bit (nula ili jedan)
- bajt (osam bitova)
- kilobajt (KB=$2^{10}$ bajtova=1024 bajta)
- megabajt (MB=$2^{20}$ bajtova=1024 KB)
- gigabajt (GB=$2^{30}$ bajtova=1024 MB)
- terabajt (TB=$2^{40}$ bajtova=1024 GB)

# Kodiranje

- zapis podataka - način na koji će podaci bliski čoveku (alfanumerički i numerički) biti prevedeni u binarnu formu koju prepoznaje računar
- univerzalna metodologiju prevođenja sa jednog jezika na drugi kao rezultat ima da za svaki podatak postoji odgovarajuća kombinacija bitova (0 i 1) koja taj podatak predstavlja u računarskom sistemu
  - 2 bita, $2^2 = 4$ kombinacije 00,01,10,11
  - 8 bitova, $2^8 = 256$ kombinacija, ASCII kod
  - Unicode, 16 bitova, 65536 kombinacija
  - UTF-8, verzija Unicoda, različitim brojem bajtova se kodiraju podaci

# ASCII cod – deo

| DEKADNI | BINARNI | OKTALNI | HEKSADEKADNI | ASCII |
|---|---|---|---|---|
| 32 | 010 0000 | 040 | 20 | (space) |
| 33 | 010 0001 | 041 | 21 | ! |
| 34 | 010 0010 | 042 | 22 | " |
| 35 | 010 0011 | 043 | 23 | # |
| 36 | 010 0100 | 044 | 24 | $ |
| 37 | 010 0101 | 045 | 25 | % |
| 38 | 010 0110 | 046 | 26 | & |
| 39 | 010 0111 | 047 | 27 | ' |
| 40 | 010 1000 | 050 | 28 | ( |
| 41 | 010 1001 | 051 | 29 | ) |
| 42 | 010 1010 | 052 | 2A | * |
| 43 | 010 1011 | 053 | 2B | + |
| 44 | 010 1100 | 054 | 2C | , |
| 45 | 010 1101 | 055 | 2D | - |
| 46 | 010 1110 | 056 | 2E | . |
| 47 | 010 1111 | 057 | 2F | / |
| 48 | 011 0000 | 060 | 30 | 0 |
| 49 | 011 0001 | 061 | 31 | 1 |
| 50 | 011 0010 | 062 | 32 | 2 |

# Brojni sistemi

- mi koristimo:

**dekadni**, osnova 10, cifre 0,1,2,3,4,5,6,7,8,9

- u računaru se koristi:

**binarni**, osnova 2, cifre 0,1

- zbog lakšeg predstavljanja binarnih brojeva koriste se:

**oktalni**, osnova 8, cifre 0,1,2,3,4,5,6,7
**heksadekadni**, osnova 16, cifre 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

- prirodni brojevi **N** = {1, 2, 3, . . .}
- celi brojevi **Z** = {. . . , −2, −1, 0, 1, 2, 3, . . .}
- racionalni brojevi **Q** – mogu da se prikažu u obliku razlomka, konačni dekadni brojevi, npr. -3/5; 8/15; 16/1;
- iracionalni brojevi **I** – ne mogu da se prikažu u obliku razlomka, npr. π; $\sqrt{2}$; $-\sqrt{5}$; $\sqrt[3]{3}$; …
- realni brojevi **R** – unija skupova racionalnih i iracionalnih brojeva, npr. 6; 3,56; -1/5; π; …

# Binarni brojni sistem

- **iz prirodnog dekadnog broja u binarni**
  - podeliti dekadni broj brojem 2
  - zapisati ostatak deljenja (0 ili 1)
  - postupak se ponavlja dok se ne dobije nula
  - ostaci deljenja predstavljaju binarni broj koji se čita u obrnutom smeru

| | |
|---|---|
| 87 : 2 = 43 | ostatak 1 |
| 43 : 2 = 21 | ostatak 1 |
| 21 : 2 = 10 | ostatak 1 |
| 10 : 2 = 5 | ostatak 0 |
| 5 : 2 = 2 | ostatak 1 |
| 2 : 2 = 1 | ostatak 0 |
| 1 : 2 = 0 | ostatak 1 |

$87_{10} = 1010111_2$

# Binarni brojni sistem

- **dekadni brojevi koji su manji od 1**
  - pomnožiti dekadni broj brojem 2
  - ako je dobijeni proizvod veći od 1, u binarnom broju se piše 1
  - ako je dobijeni proizvod manji od 1, u binarnom broju se piše 0
  - samo ako je dobijeni proizvod veći od 1, od njega se oduzima 1
  - postupak se ponavlja dok se ne dobije nula
  - dobijene binarne cifre treba čitati normalno, sa nulom i zarezom ispred

| | |
|---|---|
| 0.3125 x 2 = 0.625 | piše se 0 |
| 0.625 x 2 = 1.25 | piše se 1 |
| 0.25 x 2 = 0.5 | piše se 0 |
| 0.5 x 2 = 1.0 | piše se 1 |

$0.3125_{10} = 0.0101_2$

# Binarni brojni sistem

- **iz binarnog u dekadni -** zbir proizvoda svake cifre binarnog broja sa odgovarajućim stepenom broja dva

$101101_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45_{10}$

- **realan binarni u realan dekadni**

$101011.1011 = 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-4} = 32 + 8 + 2 + 1 + 0.5 + 0.125 + 0.0625 = 43.6875$

# Oktalni brojni sistem

- **iz dekadnog u oktalni**

127 : 8 = 15    ostatak 7
15 : 8 = 1    ostatak 7
1 : 8 = 0    ostatak 1
$127_{10} = 177_8$

- **dekadni brojevi manji od 1** u oktalni
  - pomnožiti dekadni broj brojem 8
  - ako je dobijeni proizvod veći od 1, u oktalnom broju se piše cifra jedinica i taj broj se oduzima od dobijenog proizvoda
  - ako je dobijeni proizvod manji od 1, u oktalnom broju se piše 0
  - ponavlja se postupak, sve dok se ne dobije nula

0.3125 x 8 = 2.5  piše se 2
0.5 x 8 = 4.0    piše se 4
$0.3125_{10} = 0.24_8$

# Oktalni brojni sistem

- **iz oktalnog u dekadni:**

$37_8 = 3 \times 8^1 + 7 \times 8^0 = 24 + 7 = 31_{10}$

- **iz binarnog u oktalni:** grupišu se po 3 cifre, sa desne u levu stranu, dodaju se vodeće nule sa leve strane - svaka grupa od tri broja menja se jednom oktalnom cifrom

| | |
|---|---|
| 000 | 0 |
| 001 | 1 |
| 010 | 2 |
| 011 | 3 |
| 100 | 4 |
| 101 | 5 |
| 110 | 6 |
| 111 | 7 |

$1101101111_2 = 001\ 101\ 101\ 111 = 1557_8$

- **iz oktalnog u binarni:** svaka oktalna cifra se prikazuje sa 3 bita

$76543_8 = 111\ 110\ 101\ 100\ 011 = 111110101100011_2$

# Heksadekadni brojni sistem

- **iz dekadnog u heksadekadni**

943 : 16 = 58   ostatak 15 tj. F

58 : 16 = 3   ostatak 10 tj. A

3 : 16 = 0   ostatak 3

$943_{10} = 3AF_{16}$

- **dekadni brojevi manji od 1** u heksadekadni

0.015625 x 16 = 0.25   piše se 0

0.25 x 16 = 4.0   piše se 4

$0.015625_{10} = 0.04_{16}$

- **iz heksadekadnog u dekadni**

$3B_{16} = 3 \times 16^1 + 11 \times 16^0 = 48 + 11 = 59_{10}$

- **iz heksadekadnog u binarni**

$AF3_{16} = 1010\ 1111\ 0011 = 101011110011_2$

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

# Predstavljanje brojeva

binarni brojevi mogu da budu predstavljeni kao:

- **neoznačeni** – nemaju oznaku za znak broja – zapis je identičan njihovoj reprezentaciji u binarnom brojnom sistemu
- **označeni** – imaju konačan i nepromenljiv broj bitova (8,16 ili 32), a predstavljaju se kao:
  - **znak i apsolutna vrednost** (na vodećem mestu se dodaje cifra 0 ako je broj pozitivan ili 1 ako je broj negativan) – nedostaci: dva različita binarna zapisa za nulu; za otkrivanje eventualnog prekoračenja prilikom računskih operacija potrebno je ispitivati znak i apsolutnu vrednost oba operanda
  - **nepotpuni komplement** – pozitivnim brojevima se ne radi komplement, kod negativnih se znak minus zamenjuje brojem 1, a ostale cifre se konvertuju (nule u jedinice, a jedinice u nule) – efikasnija aritmetika ali takođe dva različita binarna zapisa nule što otežava operacije poređenja sa nulom
  - **potpuni komplement** – odredi se nepotpuni komplement i na mestu najmanje težine se doda jedan, prekoračenje se ignoriše

# Predstavljanje brojeva

| Dekadna vrednost | Znak i apsolutna vrednost | Nepotpuni komplment | Potpuni komplement | Uvećanje 128 |
|---|---|---|---|---|
| +127 | 01111111 | 01111111 | 01111111 | 11111111 |
| +64 | 01000000 | 01000000 | 01000000 | 11000000 |
| +32 | 00100000 | 00100000 | 00100000 | 10100000 |
| +16 | 00010000 | 00010000 | 00010000 | 10010000 |
| +15 | 00001111 | 00001111 | 00001111 | 10001111 |
| +10 | 00001010 | 00001010 | 00001010 | 10001010 |
| +9 | 00001001 | 00001001 | 00001001 | 10001001 |
| +8 | 00001000 | 00001000 | 00001000 | 10001000 |
| +7 | 00000111 | 00000111 | 00000111 | 10000111 |
| +6 | 00000110 | 00000110 | 00000110 | 10000110 |
| +5 | 00000101 | 00000101 | 00000101 | 10000101 |
| +4 | 00000100 | 00000100 | 00000100 | 10000100 |
| +3 | 00000011 | 00000011 | 00000011 | 10000011 |
| +2 | 00000010 | 00000010 | 00000010 | 10000010 |
| +1 | 00000001 | 00000001 | 00000001 | 10000001 |
| +0 | 00000000 | 00000000 | 00000000 | 10000000 |
| -0 | 10000000 | 11111111 | --- | --- |
| -1 | 10000001 | 11111110 | 11111111 | 01111111 |
| -2 | 10000010 | 11111101 | 11111110 | 01111110 |
| -3 | 10000011 | 11111100 | 11111101 | 01111101 |
| -4 | 10000100 | 11111011 | 11111100 | 01111100 |
| -5 | 10000101 | 11111010 | 11111011 | 01111011 |
| -6 | 10000110 | 11111001 | 11111010 | 01111010 |
| -7 | 10000111 | 11111000 | 11111001 | 01111001 |
| -8 | 10001000 | 11110111 | 11111000 | 01111000 |
| -9 | 10001001 | 11110110 | 11110111 | 01110111 |
| -10 | 10001010 | 11110101 | 11110110 | 01110110 |
| -15 | 10001111 | 11110000 | 11110001 | 01110001 |
| -16 | 10010000 | 11101111 | 11110000 | 01110000 |
| -32 | 10100000 | 11011111 | 11100000 | 01100000 |
| -64 | 11000000 | 10111111 | 11000000 | 01000000 |
| -127 | 11111111 | 10000000 | 10000001 | 00000001 |
| -128 | --- | --- | 10000000 | 00000000 |

# Binarna aritmetika

- sabiranje

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 sa prenosom 1

```
  1    1 1 1 1  1
  10010111.1101
+    10101.11
  -------------
  10101101.1001
```

```
1 1 1   1 1 1 1  1
   1110111.1101
+    10101.11
  -------------
  10001101.1001
```

- oduzimanje

```
pozicija  6 5 4 3 2 1
          1 0 0 1 1 1
        - 0 0 1 0 1 1
          -----------
          0 1 1 1 0 0
```

```
  1 1 0
- 0 1 1
  -----
  0 1 1
```

sa pozicije 6 se pozajmljuje jedinica i pozicija 5 dobija vrednost 10, tj. 2, pozajmljuje se jedinica sa te pozicije i na poziciji 5 ostaje vrednost jedan, a na pozicji 4 je pozajmljena vrednost 10

# Znak i apsolutna vrednost

- posebno se analizira znak, a posebno apsolutna vrednost
- ako su brojevi istog znaka tada i njihov zbir ima isti znak - apsolutna vrednost zbira je jednaka zbiru njihovih apsolutnih vrednosti

  +61 + 14<br>
  +61 = 0|0111101<br>
  +14 = 0|0001110<br>
  +75 = 0|1001011
- ako su brojevi različitog znaka tada njihov zbir ima znak sabirka čija je apsolutna vrednost veća - apsolutna vrednost zbira je jednaka razlici njihovih apsolutnih vrednosti pri čemu se oduzima manja apsolutna vrednost od veće

-61 –(+ 14) = -61+ (-14)<br>
-61 = 1|0111101<br>
-14 = 1|0001110<br>
-75 = 1|1001011

+61 –(+ 14) = +61+ (-14)<br>
+61 = 0|0111101<br>
-14 = 1|0001110<br>
+47 = 0|0101111

-61 + 14<br>
-61 = 1|0111101<br>
+14 = 0|0001110<br>
-47 = 1|0101111

# Znak i apsolutna vrednost

-61 - 75
-61 = 1|0111101
-75 = 1|1001011
*** = 1|10001000

+120 + 10
+120 = 0|1111000
+10 = 0|0001010
*** = 0|10000010

- prekoračenje - nemoguće izvršiti sabiranje navedenih brojeva u osmobitnom cifarskom sistemu

- do prekoračenja prilikom sabiranja binarnih brojeva može doći samo ukoliko su brojevi istog znaka - ako se sabiraju pozitivni brojevi i dobije se negativan rezultat ili ako se sabiraju negativni brojevi i dobije se pozitivan rezultat

# Potpuni komplement

- zove se i komplement dvojke - u savremenim računarima se označeni celi brojevi zapisuju u potpunom komplementu
- prekoračenje se ignoriše
- oduzimanje se svodi na sabiranje uz prethodnu promenu znaka umanjiocu

+61 + 14
+61 = 0|0111101
+14 = 0|0001110
+75 = 0|1001011

+61 –(+ 14) = +61+ (-14)
+61 = 0|0111101
-14 = 1|1110010
+47=**1**0|0101111

-61 - 75
-61 = 1|1000011
-75 = 1|0110101
*** = **1**0|1111000

prekoračenje - znak zbira se razlikuje od znaka sabiraka - cifra se odbacuje i ne ulazi u vrednost rezultata

-61 –(+ 14) = -61+ (-14)
-61 = 1|1000011
-14 = 1|1110010
-75 = **1**1|0110101

-61 + 14
-61 = 1|1000011
+14 = 0|0001110
-47 = 1|1010001

# Celi brojevi u računaru

| Dekadna vrednost | 8-bitna reč | 16-bitna reč | Zapis |
|---|---|---|---|
| +127 | 01111111 | 0000000001111111 | znak i aps. vred. |
| +5 | 00000101 | 0000000000000101 | znak i aps. vred. |
| +0 | 00000000 | 0000000000000000 | znak i aps. vred. |
| -0 | 10000000 | 1000000000000000 | znak i aps. vred. |
| -5 | 10000101 | 1000000000000101 | znak i aps. vred. |
| -127 | 11111111 | 1000000001111111 | znak i aps. vred. |

| Dekadna vrednost | 8-bitna reč | 16-bitna reč | Zapis |
|---|---|---|---|
| +5 | 00000101 | 0000000000000101 | nepotpuni komp. |
| -5 | 11111010 | 1111111111111010 | nepotpuni komp. |
| +9 | 00001001 | 0000000000001001 | potpuni kompl. |
| -9 | 11110111 | 1111111111110111 | potpuni kompl. |

# Realni brojevi

- realni ili brojevi se u računaru predstavljaju kao:
  - brojevi sa **fiksnim zarezom** (fixed) - unapred se zna broj pozicija za razlomljeni deo
  - brojevi sa **pokretnim zarezom** (float) - moguće je imati više ili manje pozicija za razlomljeni deo, dele se na brojeve sa:
    - **jednostrukom tačnošću** (float)
    - **dvostrukom tačnošću** (double) - ostavljeno je više bitova za razlomljeni deo

# Fiksni zarez

**m** pozicija se koristi za razlomljeni deo (plavo), **n** za celobrojni deo broja (roze) i **1** za znak - broj pozicija za ceo i razlomljeni deo varira u zavisnosti od konkretnog računara ili rešenja u jeziku u kome se programira

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

primer predstavljanja pozitivnog binarnog broja 1011010,010010101

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# Pokretni zarez - float

- eksponencijalni zapis se koristi za zapis vrlo velikih i vrlo malih brojeva
- **$R = m \times b^e$** gde je: **R** - vrednost broja, **m** -mantisa, **b** - osnova brojnog sistema, **e** - eksponent

    dekadni broj 1200000000000 = $1.2 \times 10^{12}$

    dekadni broj 0.0000000378 = $3.78 \times 10^{-8}$

    binarni broj 11010000000 =$1.101 \times 2^{10}$
- **normalizovana** notacija podrazumeva položaj decimalnog zareza u mantisi neposredno iza prve nenulte cifre
- prilikom zapisa negativnih celih vrednosti eksponenta u potpunom komplementu javlja se problem prilikom poređenja dve vrednosti, zato se eksponent ponderiše, tj. na stvarnu vrednost eksponenta se dodaje konstanta:

    127 za float

    1023 za double

# Pokretni zarez

$7.6875_{10}=111.1011_2=1.111011 \times 2^2$
S=0
$E'=2+127=129_{10}=10000001_2$
F=11101100...00

$-17.5_{10}$
S=1
$E'= 4+127=131_{10}=10000011_2$
F=000110000...00

po standardu IEEE754 za **float**:
mantisa - 23 bita
eksponent - 8 bitova
jedan bit za znak broja
ukupna dužina - **32 bita**

za **double**:
mantisa - 52 bita
eksponent - 11 bitova
jedan bit za znak broja
ukupna dužina **- 64 bita**