MASARYKOVA UNIVERZITA

Reprezentace dat v informačních systémech

Jaroslav Šmarda

Reprezentace dat v informačních systémech

- Reprezentace dat v počítači
- Datové typy
- Proměnná
- Uživatelské datové typy
- Datové struktury: pole, zásobník, seznam, binární strom

Reprezentace dat v počítači

- Paměť počítače:
  - hlavní paměť (RAM)
  - mezipaměť v procesoru (cache)
    - registr (segment mezipaměti)
  - trvalé úložiště (např. pevný disk)
    - Data a instrukce zůstanou uložena i po vypnutí počítač

Paměť počítače

- Nejmenší jednotka paměti:
  - Bit (Binary Digit) – 0 nebo 1
- Základní jednotka paměti:
  - Byte = 8 bitů
  - 0 1 0 0 1 1 1 0
  - 2x2x2x2x2x2x2x2 = 2^8 = 256 možností

Paměť počítače

- 4 bity (1/2 bytu)
  - 0 1 1 1
  - 2x2x2x2 = 16 možností
  - šestnáctková soustava (hexadecimální)

Paměť počítače

7D 30 FF 4A

0111 1101
0011 0000
1111 1111
0100 1010

4 bity	Hexadecimální číslice
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

Reprezentace dat v počítači

- Data
  - posloupnost bytů
  - celé číslo, reálné číslo, text, zvuk, fotka, video
  - jsou vždycky nějakého typu
- Datový typ
  - určuje množství paměti potřebné k uložení a druh dat
  - velikost paměti pro určitý datový typ závisí na typu počítače (Intel, iPhone, …)
  - velikost paměti pro určitý datový typ může záviset na programovacím jazyce

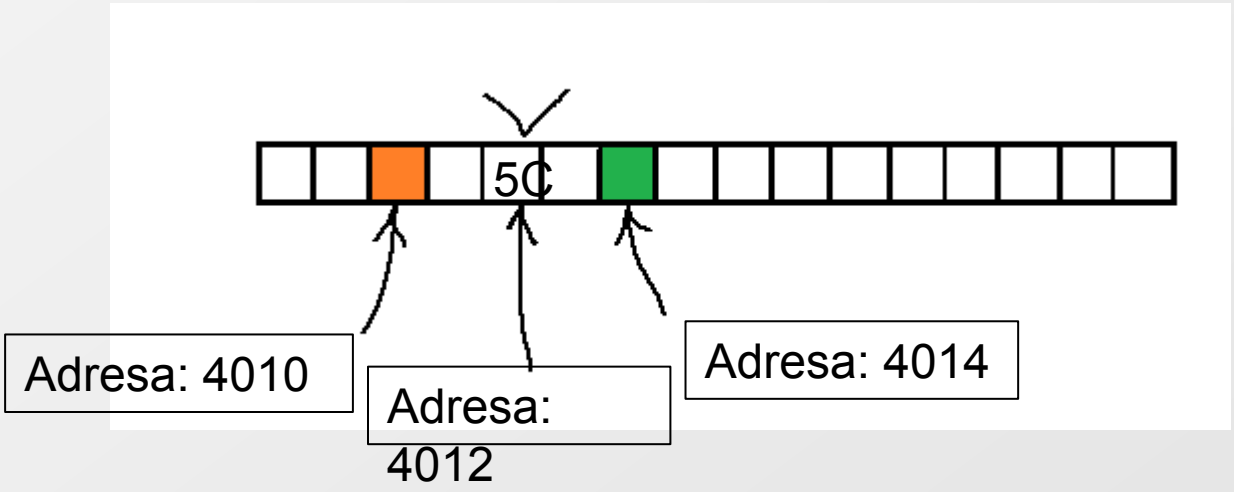
Skupiny datových typů

- celočíselné (byte, short, int, long)
- s plovoucí desetinnou čárkou (float, double)
- znakové (char)
- s logickou hodnotou (boolean)

Datový typ	Velikost v bitech	Rozsah	Skupina
short	16 bitů	-32 768 až 32 767	celé číslo (short integer)
long	64 bitů		celé číslo (long int)
double	64 bitů	$1,7^{-308}$ až $1,7^{308}$	číslo s pohyblivou desetinnou čárkou (dvojnásobná přesnost)
boolean	1 bit		logická hodnota (1=pravda, 0=nepravda)

Paměť počítače

- paměť s přímým přístupem (RAM – Random Access Memory)
- všechna paměťová místa kdykoliv dostupná přes adresu



Proměnná

- Abstrakce paměťového místa
- Je určitého datového typu
- V programovacím jazyce má symbolické jméno (název proměnné)
- Pozn.: paměťové místo má adresu

Deklarace proměnné

- Deklarace proměnné v programovacím jazyce:

  int cisloStudenta;

  int pocetStudentu;

  float teplotaVenku;

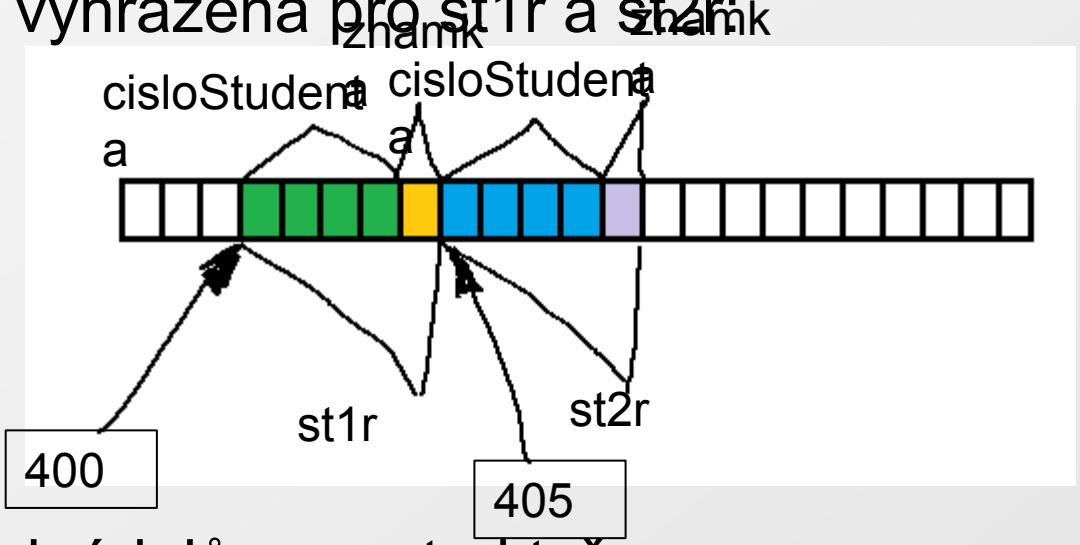
  <datový typ> <název proměnné>;
- Název proměnné:
  - začíná (a..z, A..Z) a obsahuje (a..z,A..Z,0..9)
- Deklarace = vyhrazení paměti v počítači
- Definice = jen určení nového datového typu (bez konkrétního vyhrazení paměti)

Uživatelské datové typy

- Definice uživatelského datového typu:
  - struct ZaznamOStudentovi {
    int cisloStudenta;
    char znamka;
    };
- Deklarace proměnné uživatelského datového typu ZaznamOStudentovi:
  - struct ZaznamOStudentovi st1r;
    struct ZaznamOStudentovi st2r;

Uživatelské datové typy

- paměť vyhrazená pro st1r a st2r:



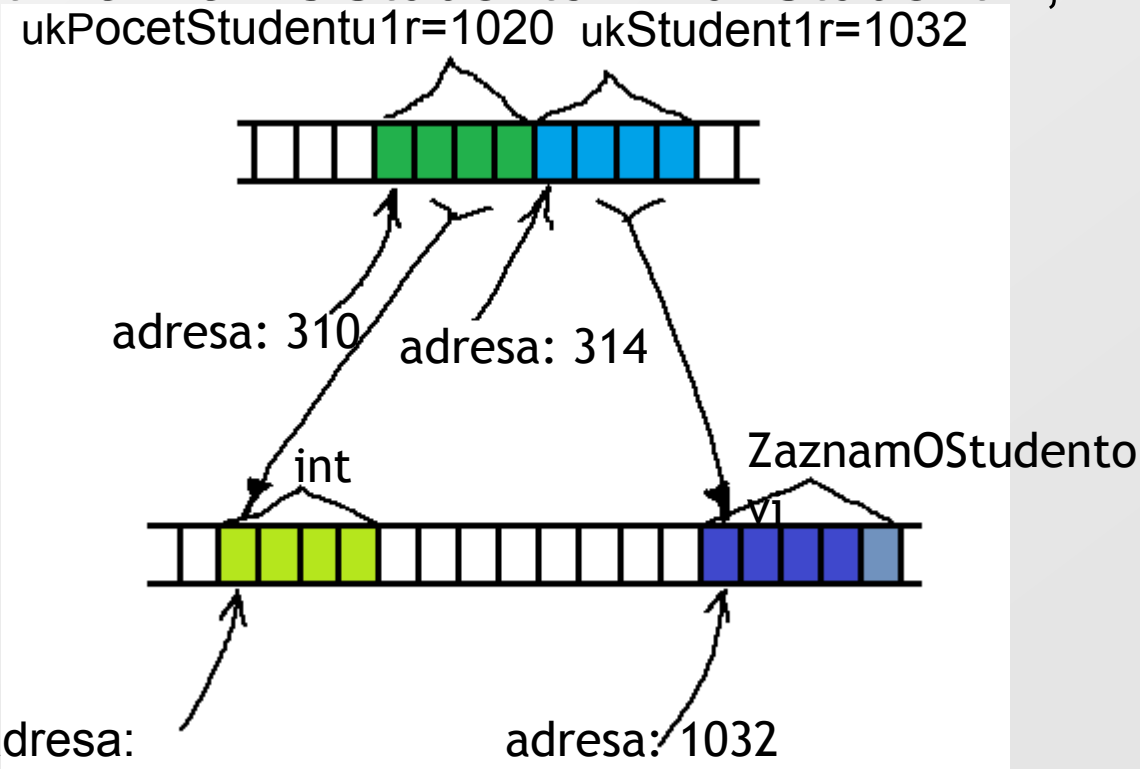
- přístup k údajům ve struktuře:
  - st1r.cisloStudenta
    st1r.znamka

Ukazatel (pointer)

- Abstrakce paměťové adresy
- Velikost ukazatele: 4 byty

Ukazatel

- Deklarace ukazatelů:
  - int *ukPocetStudentu1r;
    struct ZaznamOStudentovi *ukStudent1r;



Proměnná uživatelského datového typu

- Definice uživatelského datového typu:

```
struct ZaznamOStudentovi {
        int cisloStudenta;
        char znamka;
};
```

- Deklarace proměnné uživatelského datového typu ZaznamOStudentovi:

```
struct ZaznamOStudentovi st;
```

- Odkaz na atribut proměnné:

```
st.cisloStudenta=137246;
st.znamka="A";
```

Datová struktura: pole

- Potřebuji několik (? 10, 50, 1000,…) proměnných stejného typu – všichni studenti ročníku
- Pole:
  - množina homogenních prvků
  - přístupný kdykoliv kterýkoliv prvek

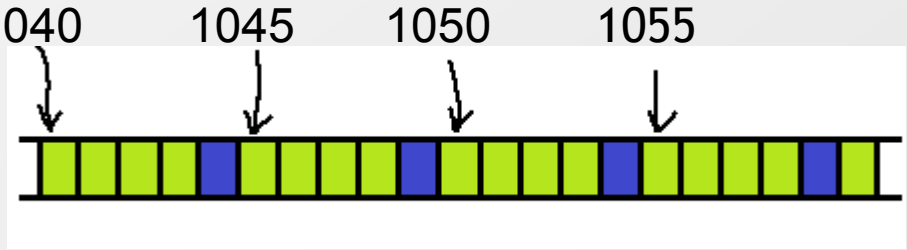
Datová struktura: pole

- Deklarace pole:

  struct ZaznamOStudentovi st[1000];

Index prvku pole

- První prvek pole: st[0] , poslední prvek: st[999]



- Odkaz na atribut prvku pole:

  st[315].cisloStudenta=128753;
  st[315].znamka="A";

- adresa_prvku_i =

  adresa_začátku_pole + (i x délka_prvku_pole)

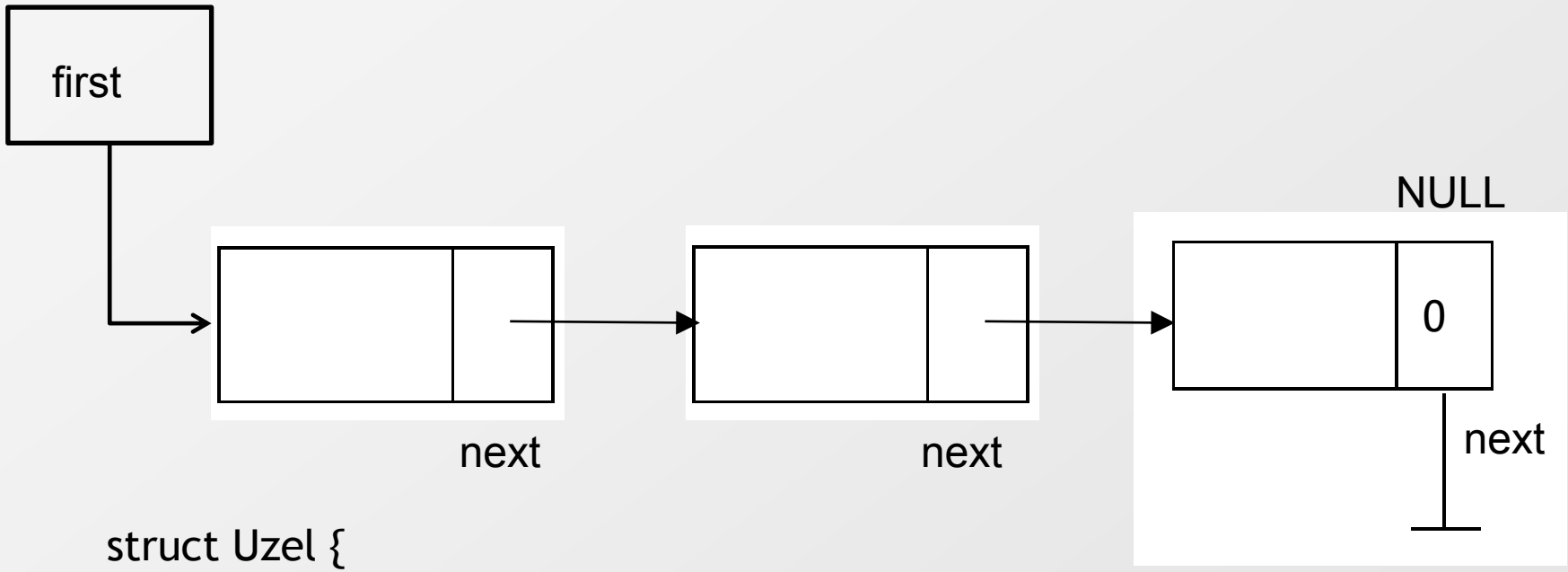
Datová struktura: pole

- Vícerozměrné pole:
  - char znamka [60] [5];
  - znamka[20,2]=“B”;

Datová struktura: spojový seznam

- Dynamická datová struktura
- Položka spojového seznamu: uzel
- Usnadňuje změnu uspořádání dat

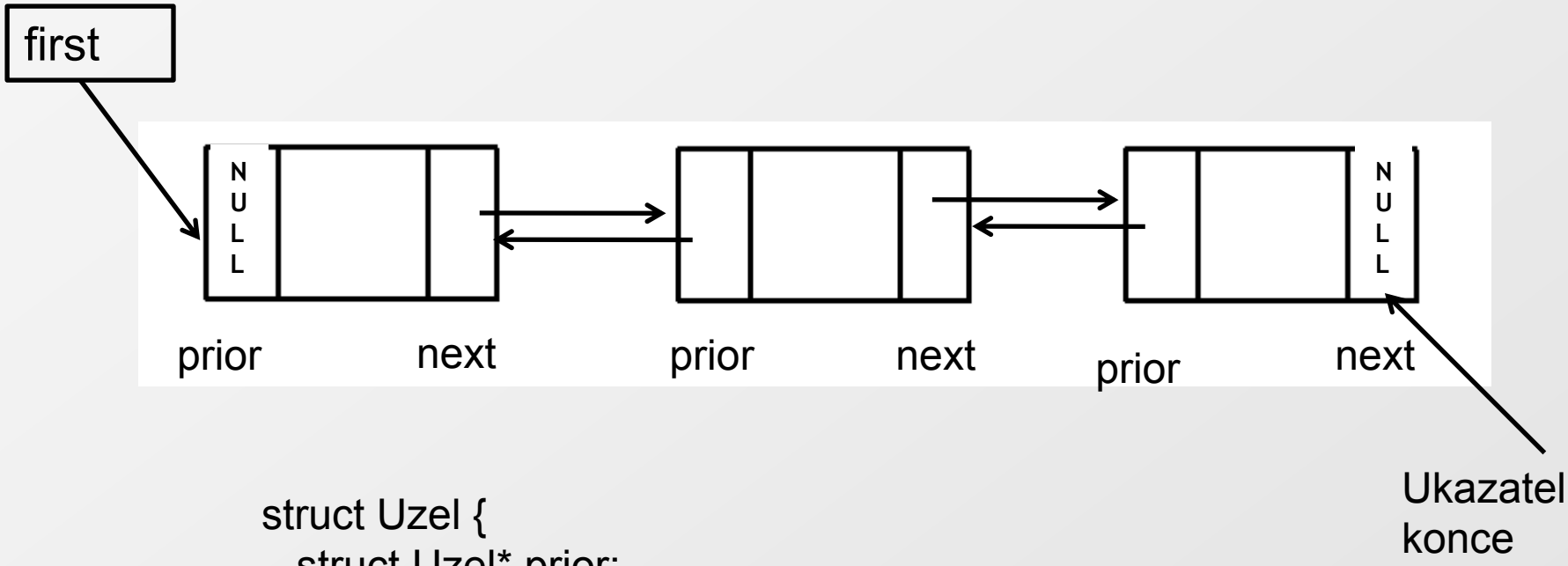
Seznam – jednosměrný spojový seznam




```
struct Uzel {
        struct Tobsahu o;
        struct Uzel* next;
}


struct Uzel* first;
```

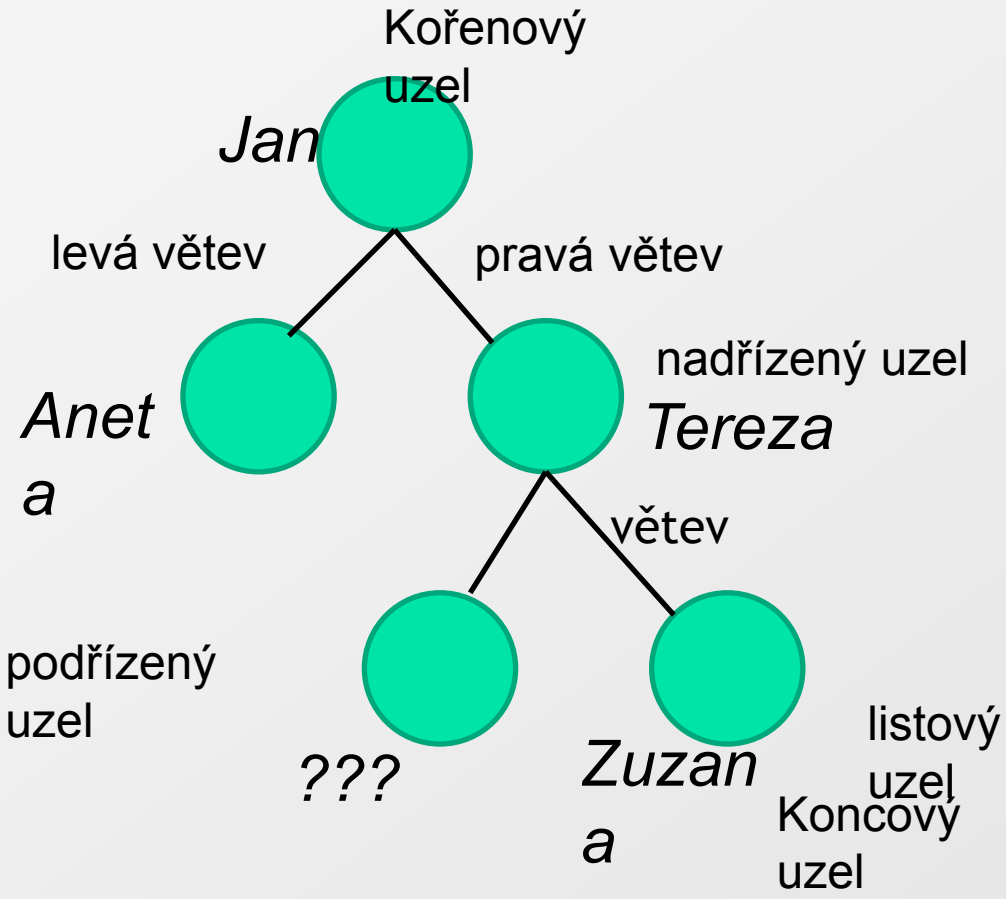
Seznam – obousměrný spojový seznam




```
struct Uzel {
   struct Uzel* prior;
   struct Tdata data;
   struct Uzel* next;
}

struct Uzel* first;
```

Datová struktura: binární strom



Datová struktura: binární strom

```
struct BinTree {
   struct BinTree* left;
   struct Tdata data;
   struct BinTree* right;
}

struct BinTree* korenovyUzel;
```