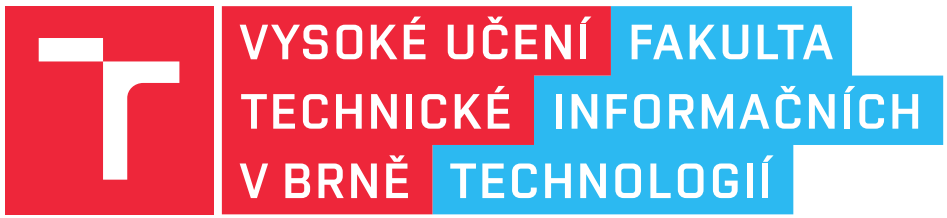
Datové štruktúry
Jednosmerne viazaný zoznam

Filip Novák
xnovakf00

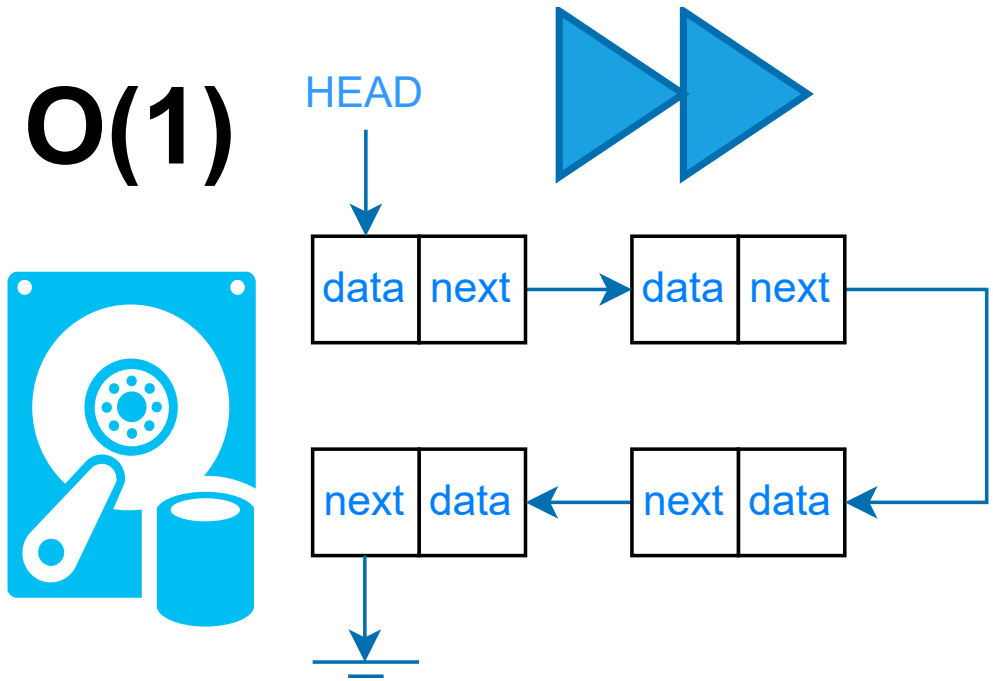
Vysoké učení technické v Brně
Fakulta informačních technologií



8. mája 2024

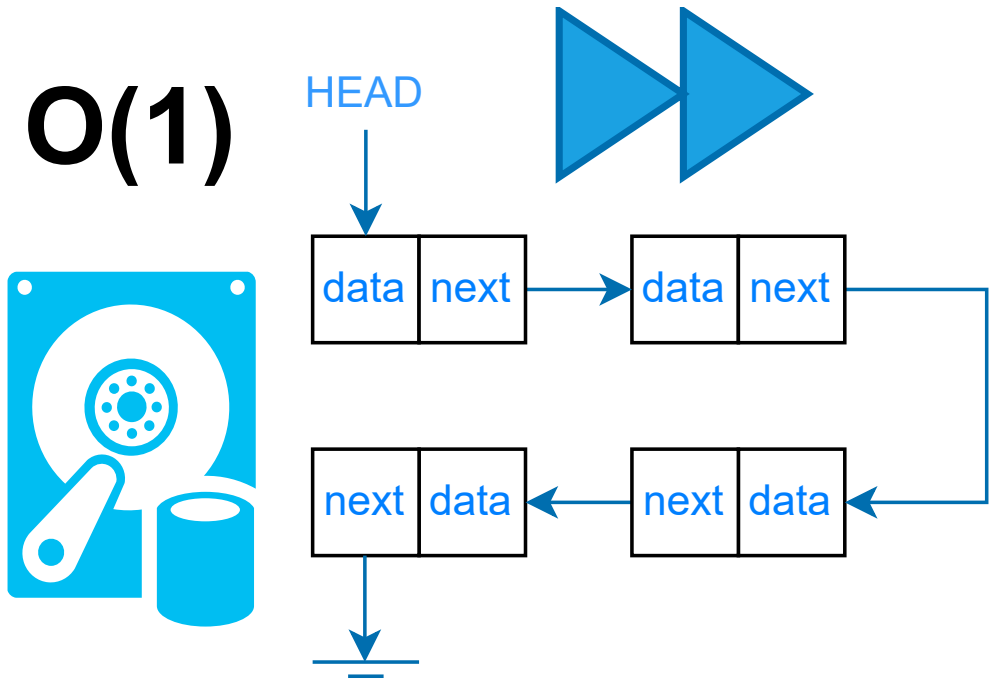
Prečo jednosmerne viazané zoznamy?

- jednoduchá implementácia



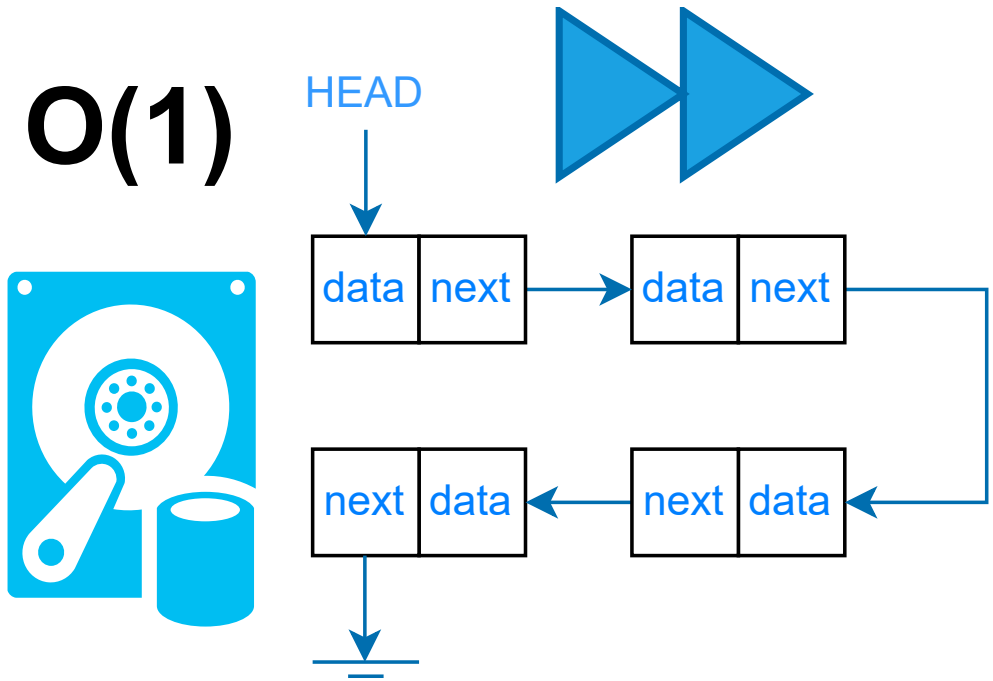
Prečo jednosmerne viazané zoznamy?

- jednoduchá implementácia
- dynamická veľkosť zoznamu



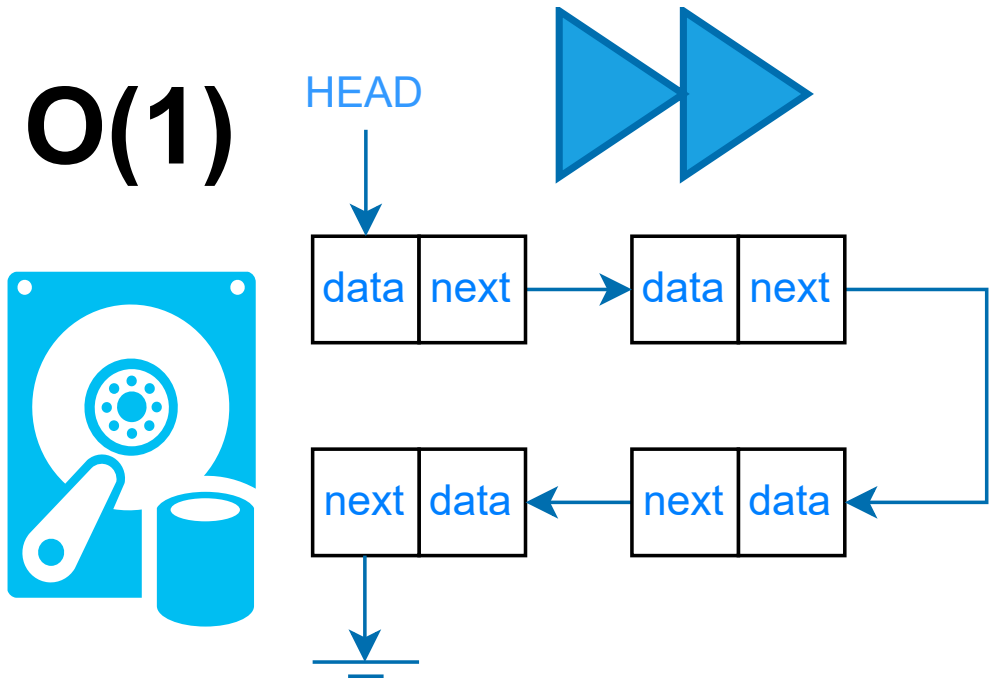
Prečo jednosmerne viazané zoznamy?

- jednoduchá implementácia
- dynamická veľkosť zoznamu
- nízka pamäťová náročnosť



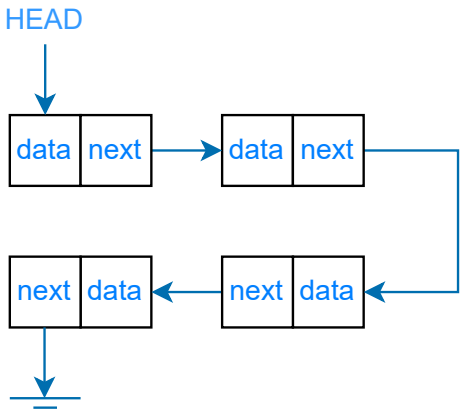
Prečo jednosmerne viazané zoznamy?

- jednoduchá implementácia
- dynamická veľkosť zoznamu
- nízka pamäťová náročnosť
- efektivita operácií



Definícia 1/2

- Jednosmerne viazaný zoznam je dynamická datová štruktúra skladajúca sa z **uzlov** prepojených odkazmi[1].

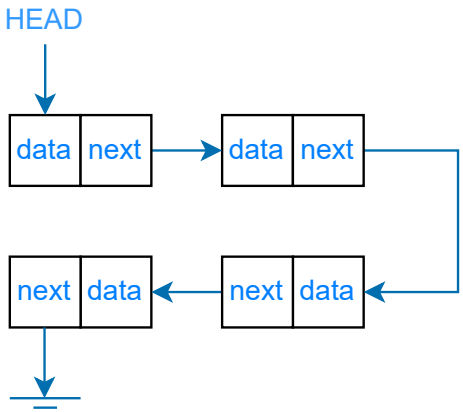


Obr.: Jednoduchá schéma

[1] Jednotlivé uzly tak nemusia byť v pamäti fyzicky za sebou.

Definícia 1/2

- Jednosmerne viazaný zoznam je dynamická datová štruktúra skladajúca sa z **uzlov** prepojených odkazmi[1].
- Na prvý prvok ukazuje špeciálny ukazateľ **HEAD**.

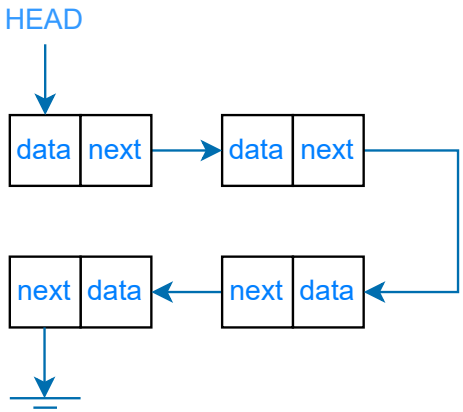


Obr.: Jednoduchá schéma

[1] Jednotlivé uzly tak nemusia byť v pamäti fyzicky za sebou.

Definícia 1/2

- Jednosmerne viazaný zoznam je dynamická datová štruktúra skladajúca sa z **uzlov** prepojených odkazmi[1].
- Na prvý prvok ukazuje špeciálny ukazateľ **HEAD**.
- Nutné časti jednotlivých uzlov:

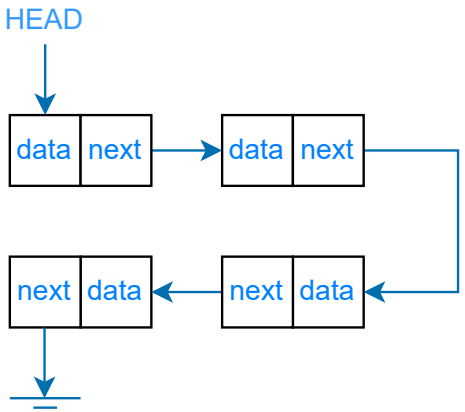


Obr.: Jednoduchá schéma

[1] Jednotlivé uzly tak nemusia byť v pamäti fyzicky za sebou.

Definícia 1/2

- Jednosmerne viazaný zoznam je dynamická datová štruktúra skladajúca sa z **uzlov** prepojených odkazmi[1].
- Na prvý prvok ukazuje špeciálny ukazateľ **HEAD**.
- Nutné časti jednotlivých uzlov:
  - dáta
  - odkaz na ďalší uzol

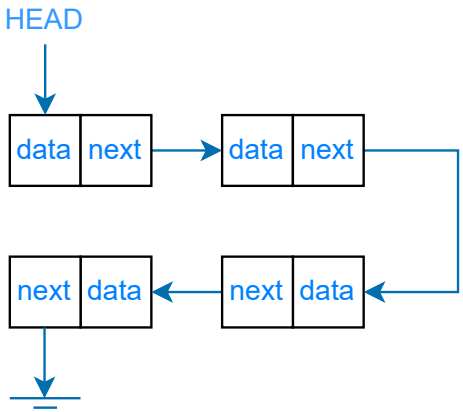


Obr.: Jednoduchá schéma

[1] Jednotlivé uzly tak nemusia byť v pamäti fyzicky za sebou.

Definícia 1/2

- Jednosmerne viazaný zoznam je dynamická datová štruktúra skladajúca sa z **uzlov** prepojených odkazmi[1].
- Na prvý prvok ukazuje špeciálny ukazateľ **HEAD**.
- Nutné časti jednotlivých uzlov:
  - dáta
  - odkaz na ďalší uzol
- Posledný uzol ukazuje na **NULL**.



Obr.: Jednoduchá schéma

[1] Jednotlivé uzly tak nemusia byť v pamäti fyzicky za sebou.

```
struct uzol
{
    data_t data;
    uzol *next;
}
```

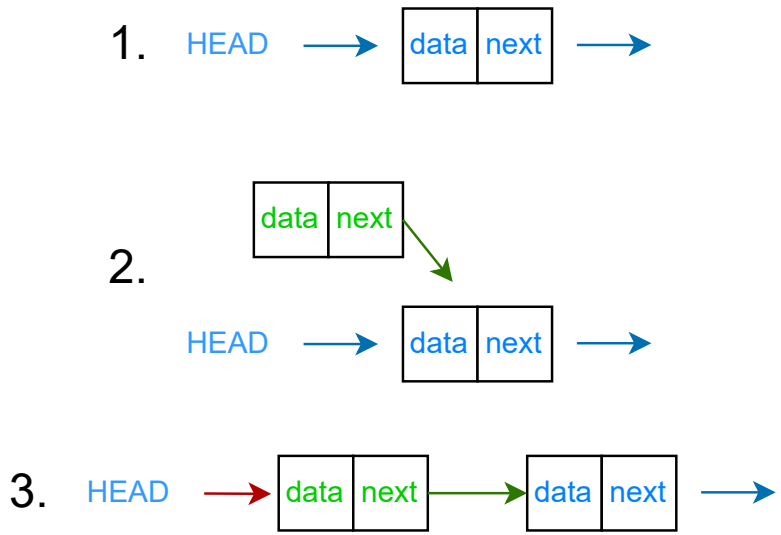
Pseudokód 1: Definícia štruktúry uzol

```
int main()
{
    uzol *head = NULL; // prazdny zoznam
    return 0;
}
```

Pseudokód 2: Inicializácia zoznamu

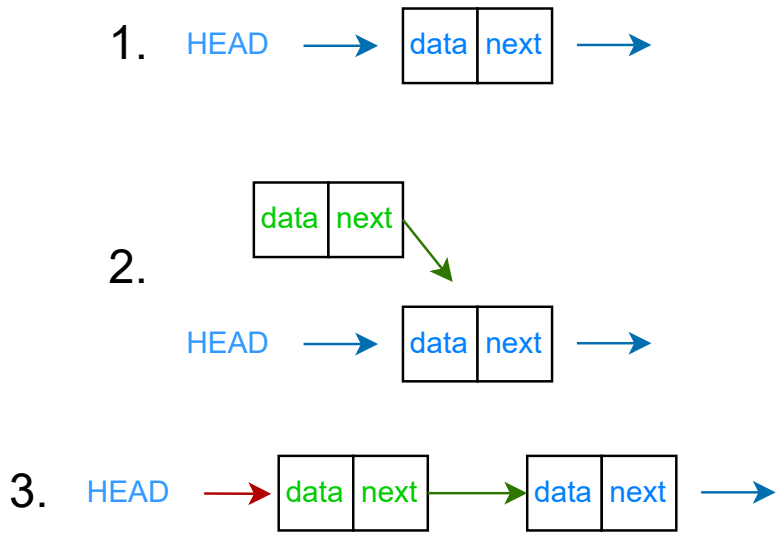
Operácie 1/5

- Pridávanie nových uzlov na začiatok zoznamu je najdôležitejšia operácia.



Obr.: Schéma pridania nového prvku na začiatok

- Pridávanie nových uzlov na začiatok zoznamu je najdôležitejšia operácia.
- Časová komplexita je **O(1)** – **next** v novom prvku sa nastaví na **next HEAD-u** a **HEAD** sa nastaví na nový prvok.



Obr.: Schéma pridania nového prvku na začiatok

```
void pridaj_na_zaciatok(uzol *head, uzol *novy)
{
    novy->next = head;
    head = novy;
    return;
}
```

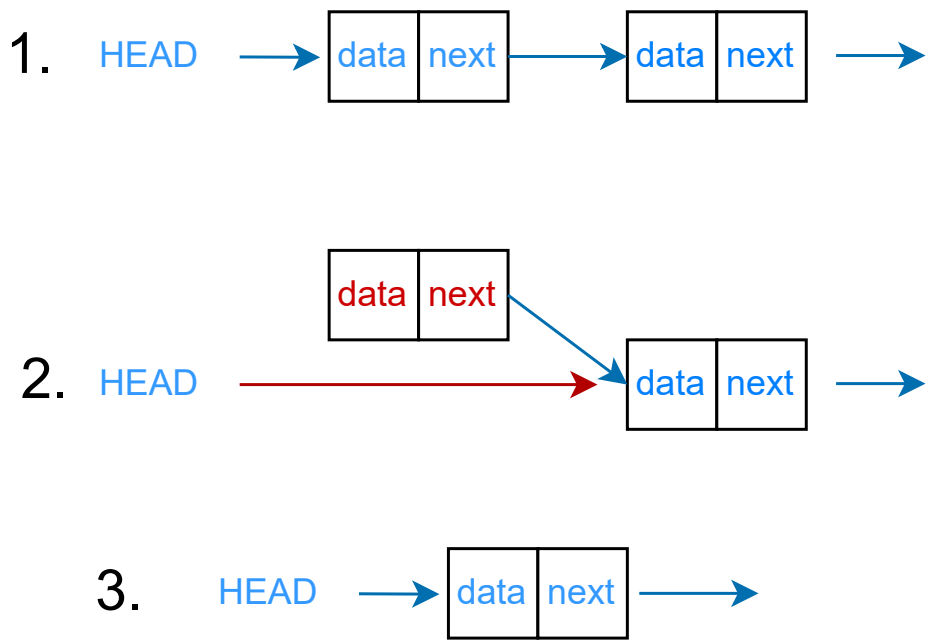
Pseudokód 3: Pridanie na začiatok zoznamu

- Prechádzanie poľom je ďalšia z dôležitých operácií.
- **O(n)** – aktuálny uzol sa nastavuje na **next**, až kým nie je **NULL**, teda koniec zoznamu.

```
void prechadzaj(uzol *head)
{
    uzol *aktualny = head;
    while(aktualny != NULL)
    {
        // nieco s uzlom sprav
        aktualny = aktualny -> next;
    }
    return;
}
```

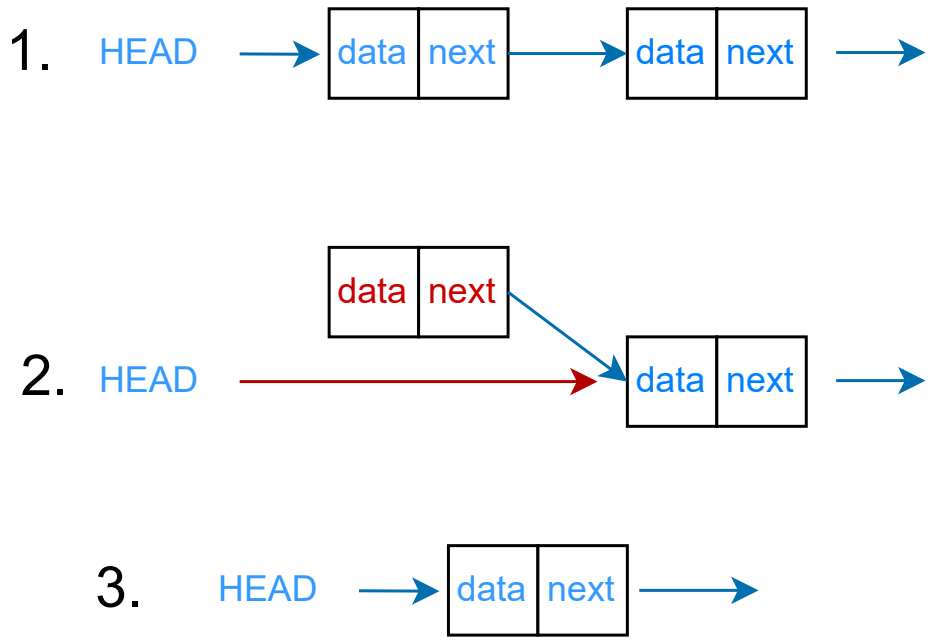
Pseudokód 4: Prechádzanie zoznamom

- Poslednou popísanou operáciou v tejto prezentácii je odstránenie prvého uzla.



Obr.: Schéma odstránenia prvého uzlu

- Poslednou popísanou operáciou v tejto prezentácii je odstránenie prvého uzla.
- **O(1)** – **HEAD** sa nastaví na **next** odstraňovaného a odstraňovaný sa uvoľní.



Obr.: Schéma odstránenia prvého uzlu

```
void odstran_prvy(uzol *head)
{
    uzol *temp = head -> next;
    head = temp -> next;
    free(*temp);
    return;
}
```

Pseudokód 5: Odstránenie prvého uzla

Zhrnutie

- V tejto prezentácii bola vysvetlená dynamická datová štruktrúra jednosmerný viazaný zoznam.
- Uvedené boli pseudokódy deklarácie a inicializácie tejto štruktúry.
- Tiež boli vysvetlené základné operácie nad zoznamom nákresom i pseudokódom.
- Pri každej operácii bola spomenutá aj časová komplexita.

Zdroje

- Lineární seznam
  `https://cs.wikipedia.org/wiki/Lineární_seznam`
- Linked list
  `https://en.wikipedia.org/wiki/Linked_list`
- Obrázky boli prebrané z voľne dostupnej knižnice aplikácie draw.io

Ďakujem za pozornosť!