

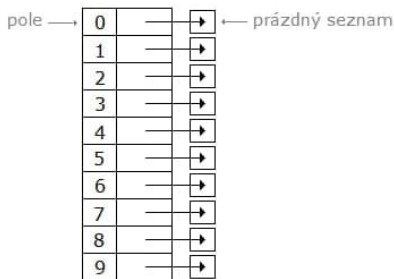
Hashovací Tabulka

Andrei Shchapaniak

VUT FIT

16. srpna 2021

Co je to hashovací tabulka?



Prázdná tabulka

- Hashovací tabulka je vyhledávací datová struktura, která asociuje hašovací klíče s odpovídajícími hodnotami.

Kde se používá?

- Implementace tabulky symbolů v překladači nějakého jazyka
- Indexování databáze
- Asociativní pole
- Objektová reprezentace
- Bezpečné třídění IP adres

Inicializace struktur hashovací tabulky

```
1 // // Define the Hash Table Item here
2 typedef struct Ht_item {
3     char* key;
4     char* value;
5 } Ht_item;
6
7 // Define the Hash Table here
8 define struct HashTable {
9     // Contains an array of pointers
10    // to items
11    Ht_item** items;
12    int size;
13    int count;
14 } HashTable;
```

Inicializace prvku hashovací tabulky

```
1  Ht_item* create_item(char* key, char* value) {  
2      // Creates a pointer to a new hash table item  
3      Ht_item* item = (Ht_item*) malloc (sizeof(Ht_item));  
4  
5      item->key = (char*) malloc (strlen(key) + 1);  
6      item->value = (char*) malloc (strlen(value) + 1);  
7  
8      strcpy(item->key, key);  
9      strcpy(item->value, value);  
10  
11     return item;  
12 }  
13
```

Inicializace hashovací tabulky

```
1  HashTable* create_table(int size) {  
2      // Creates a new HashTable  
3      HashTable* table = (HashTable* )malloc (sizeof(  
HashTable));  
4  
5      table->size = size;  
6      table->count = 0;  
7  
8      table->items = (Ht_item**) calloc (table->size,  
sizeof(Ht_item*));  
9  
10     for (int i=0; i<table->size; i++)  
11         table->items[i] = NULL;  
12  
13     return table;  
14 }  
15
```

Vyčištění alokované paměti

```
1  void free_item(Ht_item* item) {
2      // Frees an item
3      free(item->key);
4      free(item->value);
5      free(item);
6  }
7
8  void free_table(HashTable* table) {
9      // Frees the table
10     for (int i=0; i<table->size; i++) {
11         Ht_item* item = table->items[i];
12         if (item != NULL)
13             free_item(item);
14     }
15
16     free(table->items);
17     free(table);
18 }
19
```

Výhody a nevýhody hashovací tabulky

- Hashovací tabulka je velmi efektivní vyhledávací metoda
- Pokud uvažujeme ideální případ, hashovací tabulka je rychlejší než binární vyhledávací strom
- Neposkytuje intervalové dotazy, tj. vyhledání všech klíčů v daném intervalu
- Hashovací tabulka umožňuje pouze dotazy na jednotlivé klíče

- Wikipedie [▶ Link](#)
- Práce s hashovací tabulkou [▶ Link](#)
- Kod [▶ Link](#)