

Tema 0 HashTable

Termen de predare: **07.03.2011**

Termen de predare hard: **16.03.2011**

- Last modified: 20.02.2011 22:35
- Last modified: 20.02.2011 23:30
- Last modified: 21.02.2011 00:45
- Last modified: 21.02.2011 22:35
- Last modified: 23.02.2011 19:00
- Last modified: 27.02.2011 15:45
- Last modified: 02.03.2011 16:15

Scopul temei

- Recapitularea lucrului cu funcțiile ANSI C:
 - lucrul cu fișiere
 - alocare dinamică de memorie
- Folosirea pointerilor

Enunț

Să se implementeze **în C** o tabelă de dispersie (hashtable) ce va conține cuvinte.

Operațiile ce trebuie suportate de această tabelă sunt următoarele:

- `add` *adauga cuvântul la hashtable (nu se permit dubluri)*
- `remove` *șterge cuvântul din hashtable (nu e obligatoriu să existe cuvântul)*
- `clear` *golește tabela*
- `find []` *caută cuvântul în hashtable scrie True sau False pe o linie nouă în fișierul specificat sau la consolă dacă acest parametru lipsește*
- `print_bucket []` *scrie cuvintele din bucketul respectiv, pe o singură linie și separate de spațiu în fișierul specificat sau la consolă dacă acest parametru lipsește, index_bucket este valid*
- `print []` *printează toate bucket-urile pe linii diferite, începând cu bucketul 0, în fișierul specificat sau la consolă dacă acest parametru lipsește*
- `resize double` *dublează dimensiunea hash-ului (bucket-urile vor fi parcurse în ordine și cuvintele sunt redistribuite)*
- `resize halve` *înjumătățește dimensiunea hash-ului (bucket-urile vor fi parcurse în ordine și cuvintele sunt redistribuite, memoria în surplus va fi dealocată)*

Aceste comenzi se vor regăsi una per linie.

Programul va primi o serie de parametri:

- Primul parametru este lungimea inițială a hashului
- Următorii parametri pot lipsi și reprezintă o listă de fișiere din care se face citirea. Dacă aceștia lipsesc citirea se face de la STDIN

Exemplu:

```
./tema0 256 hash1.in hash2.in
./tema0 100 < hash1.in
```

```
hash1.in:
add tema
add hash
print hash.out
find tema hash.out
remove tema
find tema hash.out
print hash.out
resize double
print
```

```
print_bucket 185 hash2.out
```

Hashtable-ul implementat va conține SIZE bucketuri. Fiecare bucket va conține cuvintele în ordinea în care ele au fost introduse.

Pentru operația de resize bucketurile vor fi parcurse în ordine și redistribuite. Cuvintele din bucket vor fi parcurse începând cu cel mai vechi și terminand cu cel mai recent.

Precizări generale

- Valorile introduse în hashtable sunt cuvinte [A-Za-z].
- Un tablou **nu** poate conține duplicate.
- Nu există limitări pentru lungimea unui bucket sau a unui cuvânt.
- Inserarea într-un tablou (bucket) se face la finalul acestuia.
- Funcția de hash ce **trebuie** folosită (în întreaga temă) este definită în *hash.c*. Nu poate fi folosită altă funcție
- Programul trebuie să execute comenzile în ordine, așa cum au fost primite citite din fișier(e).
- În fișiere se va scrie în modul append
- Dacă dimensiunea hash-ului este impară ($2k+1$), după înjumătățire dimensiunea lui va fi k
- Lungimea hash-ului și a unui cuvânt vor fi reprezentate număr pe 32 de biți (cu semn)
- Șirul vid nu este valid
- Dimensiunea hash-ului va fi întotdeauna pozitivă
- Executabilul generat va purta numele tema0
- Comportamentul dorit la resize este unul echivalent cu urmatorul: se creează un nou hash, se iterează prin bucketurile din vechiul hash și se adaugă în noul hash

Precizări VMChecker

Arhiva temei va fi încărcată de două ori pe vmchecker (linux și windows). Arhiva trimisă trebuie să fie aceeași pe ambele platforme (se vor compara cele două trimise).

Arhivele trebuie să conțină sursele temei, README și două fișiere makefile care conțin target-urile build și clean:

- makefile-ul de linux se va numi GNUmakefile
- makefile-ul de windows se va numi Makefile

Executabilul rezultat din operația de compilare se va numi tema0

Punctare

- Tema va fi punctată cu maxim 0.5 puncte dacă aceasta compilează pe o platformă și cu 1 punct dacă ea compilează pe ambele platforme, fără a folosi directive de preprocesare:

```
#ifdef LINUX
[...]
```

```
#ifdef WIN32
[...]
```

- **Atenție** Depunctarea pentru această temă este de 1 punct/zi. Depunctarea **nu** se oprește la nota 7

Nota mai poate fi modificată prin depunctări suplimentare:

- [Lista generala de depunctari](#)
- -0.1 diverse alte probleme constatate în implementare
- -4 alocare statică hash

Materiale ajutătoare

Arhiva cu funcția de hash [hash.zip](#)

[ANSI C reference](#)

[Hash table](#)

Teste:

- [linux](#)
- [windows](#)

FAQ

- **Q:** Tema 0 se poate face în C++?
 - **A:** Nu.
- **Q:** "Valorile introduse în hashtable sunt cuvinte [A-Za-z]" - trebuie verificate cuvintele la introducere?
 - **A:** Nu.
- **Q:** Se pot folosi directive de preprocesare de tipul #define?
 - **A:** Da.

Lista de discuții

Pentru întrebări sau nelămuriri legate de temă puteți [căuta](#), [consulta](#) sau [trimite un mail pe](#) lista de discuții (trebuie să fiți [înregistrați](#)).

From:
<http://elf.cs.pub.ro/so/wiki/> - Sisteme de Operare

Permanent link:
<http://elf.cs.pub.ro/so/wiki/teme/tema-0>

Last update: 2011/03/07 09:53