

Primo modulo

Stefano Guerrini

Nicola Galesi

A.A. 2004/05

1 Scopo del modulo

Lo scopo del primo modulo del progetto è quello di creare una tavola contenente tutti i nodi associati agli ipertesti presenti nella base di ricerca (nella directory in cui si vogliono eseguire le ricerche).

Dato che nelle successive fasi del progetto occorrerà scorrere nuovamente tutti gli ipertesti, la tavola da creare sarà formata da due parti:

1. una tavola hash, per poter accedere in modo diretto ai nodi mediante il nome del file corrispondente;
2. una lista dei nodi presenti nella tavola hash, per potervi accedere in modo sequenziale.

Dato che l'implementazione dei successivi moduli del progetto richiederà nuovamente l'utilizzo di tavole per memorizzare informazioni, le tavole sviluppate in questo modulo saranno disegnate in modo di poter memorizzare generici elementi che hanno una stringa come chiave.

2 Specifiche del modulo

L'header file `table.h` (riportato in appendice B) contiene la definizione delle seguenti strutture:

`struct tablem` : elemento di una tavola.

La stringa nel campo `key` è la chiave dell'elemento.

Il campo `info` è un puntatore alle informazioni associate alla chiave. Questo puntatore è di tipo `void *` in modo di poter implementare la tavola indipendentemente dal tipo dei dati associati alla chiave.

`struct nodelist` : lista di elementi di una tavola.

`struct table` : tavola di elementi.

Il campo `htable` contiene la tavola hash corrispondente all'implementazione vera e propria della tavola.

Il campo `ls` contiene la lista di tutti gli elementi nella tavola; l'ordine in cui gli elementi sono memorizzati in questa lista non è specificato.

L'header file `table.h` contiene inoltre i prototipi delle seguenti funzioni:

`void table_init(struct table *t)` : inizializza la tavola puntata da `t`, inizializzando la hash table e la lista degli elementi.

`int table_full(struct table *t)` : verifica se la tavola puntata da `t` è piena, ritornando true (un valore non 0) se la tavola è piena, 0 altrimenti.

`struct node *table_ins(struct table *t, char *key)` : inserisce nella tavola puntata da `t` la chiave `key`, ritornando il puntatore all'elemento inserito o NULL in caso di errore (ad esempio, tavola piena) o chiave già presente.

`struct node *table_search(struct table *t, char *key)` : cerca in una tavola (quella puntata da `t`) l'elemento della chiave `key`, ritornando il puntatore all'elemento cercato, se questo è presente nella tavola, o NULL, altrimenti.

L'header file `nodes.h`, oltre alle definizioni dei tipi per la memorizzazione del grafo, contiene il prototipo della funzione:

`struct table *create_node_table(char *dirname)` : che crea una nuova tavola le cui chiavi degli elementi sono i nomi dei file ipertesto contenuti in `dirname`. Ritornando il puntatore alla nuova tavola creata, o NULL in caso di errore. I campi `info` degli elementi della nuova tavola creata saranno lasciati a NULL; le informazioni associate a ciascun nodo saranno determinate e memorizzate successivamente.

2.1 Cosa si deve implementare

Per completare il modulo si devono implementare i seguenti file:

1. `hashtab.h`
che definisce il tipo di dato `HASTABLE` usato in `struct table`;
2. `hashtab.c`
che implementa il tipo di dato `HASTABLE`;
3. `table.c`
che implementa le funzioni specificate in `table.h`;
4. `modulo1.c`
che implementa la funzione `create_node_table` contenuta in `nodes.h`.

2.2 Le tavole hash

Il file `hashtab.h`, oltre alla dichiarazione di tipo di `HASTABLE`, deve contenere i prototipi di tutte le funzioni di interfaccia necessarie all'uso di questa struttura dati (inserimento e ricerca di elementi, test tavola piena e inizializzazione).

A `table.h`

```
/*
 * Laboratorio di Programmazione - A.A. 2004-05
 * Canali A-D (Nicola Galesi), P-Z (Stefano Guerini)
 *
 * table.h
 */

#include "hashtab.h";

/*
 * Elemento di una tavola.
 *
 * Il campo key e' la chiave dell'elemento, una stringa.
 * Il campo info e' un puntatore alla struttura contenente i dati
 * associati alla chiave. Per poter implementare la tavola indipendentemente
 * dal tipo dei dati associati alla chiave, questo campo e' di tipo void *.
 */
struct tabelem {
    char *key; /* chiave dell'elemento */
    void *info; /* informazioni associate alla chiave */
};

/*
 * Lista di elementi di una tavola.
 */
struct telist {
    struct tabelem *el; /* elemento della tavola */
    struct telist *next; /* coda della lista */
}

/*
 * Tavola di elementi.
 *
 * La tavola e' formata da
 * - una tavola hash nella quale gli elementi vengono inseriti e ricercati
 * per mezzo della chiave
 * - il puntatore alla lista di tutti igli elementi nella tavola
 */
```

```

* NB La lista degli elementi serve a scorrere in modo sequenziale
*   gli elementi presenti nella tavola.
*/
struct table {
    HASHTABLE htable; /* tavola hash */
    struct telist *ls; /* lista di tutti i nodi */
};

/*
* Verifica se una tavola e' piena.
*
* Ritorna true (non 0) se la tavola e' piena, 0 altrimenti.
*/
int table_full(struct table *t);

/*
* Inizializza una tavola.
*/
void table_init(struct table *t);

/*
* Inserisce in una tavola un nuovo elemento.
*
* struct table *t: e' la tavola in cui inserire l'elemento
* char *key: la chiave dell'elemento da inserire
* Ritorna il puntatore all'elemento inserito o NULL in caso di errore
* (eg, tavola piena) o elemento gia' presente. Il campo info del nuovo
* elemento inserito viene inizializzato a NULL.
*/
struct talem *table_ins(struct table *t, char *key);

/*
* Cerca in una tavola un elemento.
*
* struct table *t: e' la tavola in cui ricercare l'elemento
* char *key: la chiave dell'elemento da cercare
* Ritorna il puntatore all'elemento cercato, se questo e' presente nella
* tavola, o NULL, nel caso in cui l'elemento non e' presente.
*/
struct talem *table_search(struct table *t, char *key);

/*
* Cerca in una tavola un elemento.
*
* struct table *t: e' la tavola in cui ricercare l'elemento
* char *key: la chiave dell'elemento da cercare
* Ritorna il puntatore all'elemento cercato, se questo e' presente nella
* tavola, o NULL, nel caso in cui l'elemento non e' presente.
*/
struct talem *table_search(struct table *t, char *key);

```

B nodes.h

```

/*
* Laboratorio di Programmazione - A.A. 2004-05
* Canali A-D (Nicola Galesi), P-Z (Stefano Guerini)
*
* nodes.h

```

```
* Versione per il primo modulo.
*/

/*
* Per i successivi moduli questo header files conterra' tutte le dichiarazioni
* relative ai tipi e alle funzioni necessarie alla gestione del grafo degli
* ipertesti.
*
* Per il primo modulo contiene solo il prototipo di create_node_table
*/

/*
* Crea una nuova tavola le cui chiavi degli elementi sono i file ipertesto
* contenuti in "dirname".
*
* Ritorna il puntatore alla nuova tavola creata, o NULL in caso di errore.
*/
struct table *create_node_table(char *dirname);
```