CSC161 2010F *Imperative Problem Solving*

Assignment 9: Hash Tables

Assigned: Friday, 19 November 2010
Due: 11:00 p.m., Wednesday, 24 November 2010

Summary: In this assignment, you will extend our implementation of hash tables.

Purposes: To give you more experience with pointers and structures.

Expected Time: Three to four hours.

Collaboration: I encourage you to work in groups of two or three students. However, you may work on your own or in groups of up to size four. You may discuss the assignment with anyone you wish, provided you clearly document such discussions.

Submitting: Email me a tarball of your important files (your `.c` files, your `.h` files, your `Makefile`, a typescript of a sample session, and anything else you deem appropriate).

Warning: So that this assignment is a learning experience for everyone, I may spend class time publicly critiquing your work.

Background

In class, we developed a simple implementation of hash tables, based on the code in K&R. In particular, we

- Specified the data structure and three associated functions (create a new table, insert a key/value pair, get by key).
- Built two testing applications: One that permitted interactive testing and one that did some straightforward unit tests.
- Implemented the functions.

We now have a limited hash table library. But many clients will want a richer library, one with more functions. In this assignment, you will add functions.

Assignment

Add the following functions to the hash table implementation and update the two test programs to work with them.

`int contains_key (struct hash *hashtab, char *key)`
: Determine if `hashtab` has an entry with key `key`.

`char *find_key (struct hash *hashtab, char *value)`
: Given a value, find a key that maps to that value. If no such key exists, return `NULL`.

`void delete (struct hash *hashtab, char *key)`

Given a key, remove the associated key/value pair from the hash table. If the key/value pair is not in the hash table, do nothing.

`void iterate (struct hash *hashtab, void (*fun)(char *key, char *value))`

Call `fun` on every key/value pair in the hash table.

`void free_hash (struct hash *hashtab)`

Free all of the memory associated with `hashtab`.