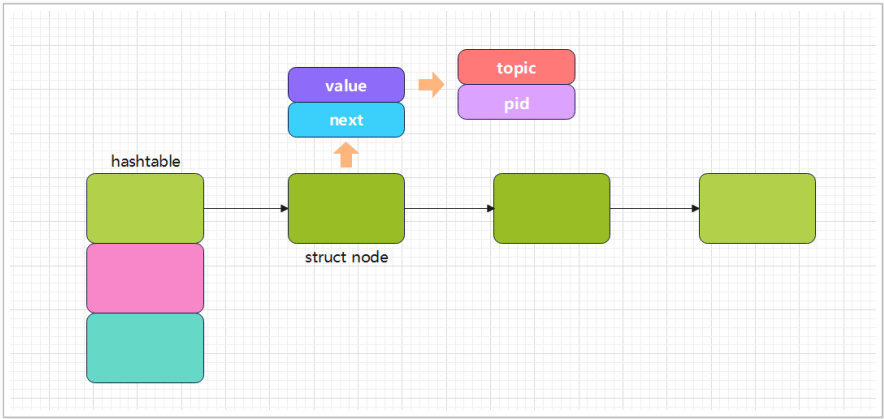


4.5 消息订阅功能实现 (四)-hashmap 模块_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 4.5 消息订阅功能实现 (四)-hashmap 模块涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。



- 在 hashmap 的数据结构中，主要是涉及两个结构体：
 - 第一个是链表节点：

```
typedef struct node{
    datatype_t data;
    struct node *next;
}hashtable_t;
```
 - 第二个是数据域

```
typedef struct datatype{
    char topic[32];
    pid_t pid;
}datatype_t;
```
- 这里的数据域是根据项目进行了定制，主要是保存主题与订阅进程的 id

```
hashtable_t **create_hashtable()
{
    int i;
    hashtable_t **h = NULL;
    h = (hashtable_t **)malloc(HTABLE_LEN * sizeof(hashtable_t *));

    memset(h,0,sizeof(HTABLE_LEN * sizeof(hashtable_t *)));

    for (i = 0;i < HTABLE_LEN;i++){
        h[i] = NULL;
    }

    return h;
}
```

- 根据项目需要，针对数据结构课程中的添加函数进行修改
 - 修改一：key 不设置为整数，而是 字符串
 - 修改二：增加字符串转换 hashcode, 在通过 hashcode 计算相应的 hash 表索引具体实现如下:
- ```
int hashcode(const char *string)
{

```

```

int hashCode = 0;

while (*string){
 hashCode += *string++;
}

return hashCode;
}

```

- 这里的 hash 算法比较简单, 直接对字符串进行求和

- 具体实现如下:

```

void insert_data_hash(hashtable_t **h, char * key, datatype_t value)
{
 hashtable_t *temp = NULL;
 hashtable_t **p = NULL;

 int index = 0, i = 0;
 int code = 0;

 code = hashCode(key);

 index = code % HTABLE_LEN;

 for (p = &h[index]; *p != NULL; p = &((*p)->next)){
 if ((*p)->data.pid > value.pid)
 break;
 }

 temp = (hashtable_t *)malloc(sizeof(hashtable_t));
 temp->data = value;
 temp->next = *p;
 *p = temp;

 return;
}

```

- 查询函数可以支持根据字符串进行查询, 并通过输出参数来得到结果, 具体实现如下:

```

int search_hash_table(hashtable_t **h, char *key, pid_t result[])
{
 int code = hashCode(key);
 int cnt = 0;
 int index = code % HTABLE_LEN;
 hashtable_t **p = NULL;

 for (p = &h[index]; *p != NULL; p = &((*p)->next)){

 printf("key : %s data : %s\n", key, (*p)->data.topic);

 if (strcmp(key, (*p)->data.topic) == 0){
 result[cnt++] = (*p)->data.pid;
 }
 }
 return cnt;
}

```

- hashmap 输出函数主要用于测试, 具体实现如下:

```

void printf_hash_table(hashtable_t **h)
{
 int i = 0;
 hashtable_t **p = NULL;

 for (i = 0; i < HTABLE_LEN; i++){
 printf("index = %d : ", i);
 for (p = &h[i]; *p != NULL; p = &((*p)->next)){
 printf("%s ", (*p)->data.topic);
 }
 putchar('\n');
 }
 return;
}

```

```
#include <stdio.h>
```

```
#include "hashmap.h"
```

```
int main(void)
{
 hashtable_t **h = NULL;

 datatype_t value[] = {
 {"hello",123},
 {"hello",889},
 {"world",456},
 {"adfdfd",789},
 };

 h = create_hashtable();

 insert_data_hash(h,"hello",value[0]);
 insert_data_hash(h,"hello",value[1]);
 insert_data_hash(h,"world",value[2]);
 insert_data_hash(h,"adfdfd",value[3]);

 int result[5] = {0};
 int ret = search_hash_table(h,"world",result);
 if (ret != 0){
 for (int i = 0;i < ret;i++){
 printf(" %d\n",result[i]);
 }
 }

 printf_hash_table(h);
 return 0;
}
```

---

全文完

---

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

