

3.3 哈希表之链地址法_物联网 / 嵌入式工程师 - 慕课网

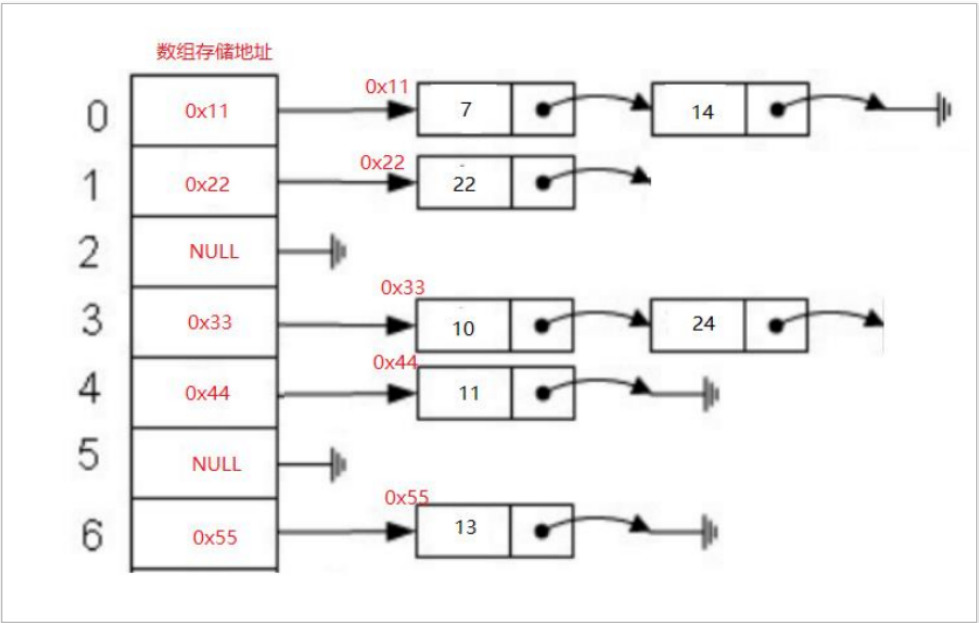
“ 慕课网慕课教程 3.3 哈希表之链地址法涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

3. 哈希表之链地址法

链地址法的本质是把我们的数组和链表一起结合起来使用了。我们数组中存储的不是有效的数据，而我们链表的结点，链表的结点中存储的才是我们的有效的数据。我们把结点的地址存放到数组中。当发生冲突的时候我们按照从小到大的顺序把同一个下标对应的结点，连接在一起。

```
int a[7] = {10,22,13,11,24,7,14};
f(key) = key % 7;
```

```
f(10) = 3;
f(22) = 1;
f(13) = 6;
f(11) = 4;
f(24) = 3 ;
f(7)  = 0 ;
f(14) = 0;
```



hash_linkaddr.c

```
#define MAX 7

typedef int datatype_t;
typedef struct node{
    datatype_t data;
    struct node *next;
}hashtable_t;
```

- 创建一个空的 huash 表数组，在堆区分配空间, 数组中存储的全部都是指针。

```

hashtable_t **create_hashtable()
{
    int i = 0;
    hashtable_t **h = NULL;
    h = (hashtable_t **)malloc(MAX * sizeof(hashtable_t *));

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

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

    return h;
}

```

```

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

    int index = 0,i = 0;
    index = key % MAX;

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

    temp = (hashtable_t *)malloc(sizeof(hashtable_t));
    temp->data = key;

    temp->next = *p;
    *p = temp;

    return ;
}

```

```

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

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

```

```

int search_hash_table(hashtable_t **h, datatype_t key)
{
    int index = key % MAX;
    hashtable_t **p = NULL;

    for(p = &h[index]; *p != NULL;p = &(*p)->next)
    {
        if((*p)->data == key)
            return 1;
    }
    putchar('\n');
    return 0;
}

```

```

int main(int argc, const char *argv[])
{
    hashtable_t **h = NULL;
    datatype_t data[MAX] = {10,22,13,11,24,7,14};
}

```

```
datatype_t value = 0;
datatype_t ret;
int i = 0;

h = create_hashtable();

for(i = 0; i < MAX; i++)
{
    insert_data_hash(h, data[i]);
}
printf_hash_table(h);
printf("please input you want to find value : ");
scanf("%d", &value);

ret = search_hash_table(h, value);
if(ret == 0){
    printf("no such data in the hash-table!\n");
}else{
    printf("value = %d is exist!\n", value);
}

return 0;
}
```

运行结果:

```
index = 0 : 7 14
index = 1 : 22
index = 2 :
index = 3 : 10 24
index = 4 : 11
index = 5 : 7 14
index = 6 : 13
please input you want find value : 13
value = 13 is exist
```

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

