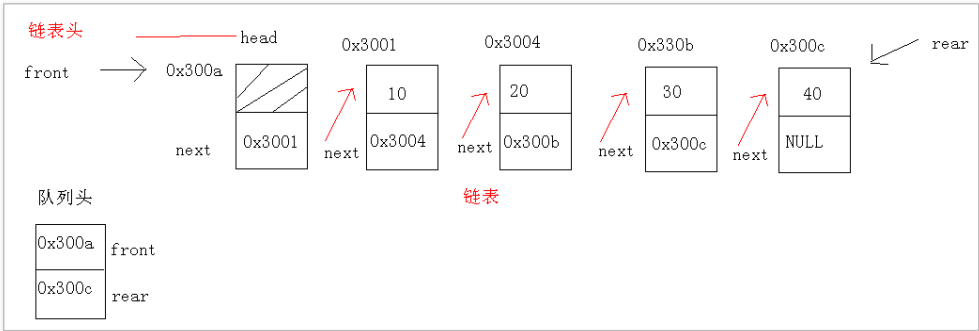


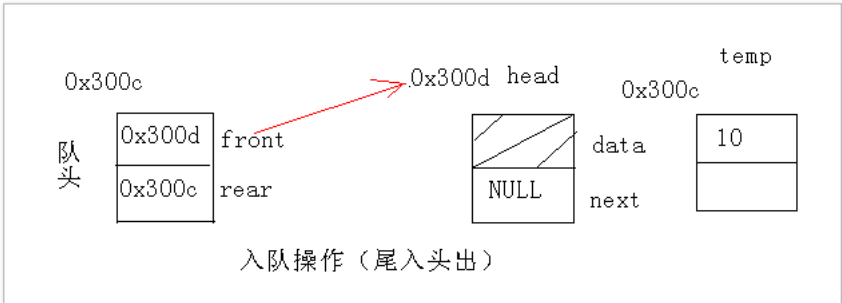
4.3 链式队列操作_物联网 / 嵌入式工程师 - 慕课网

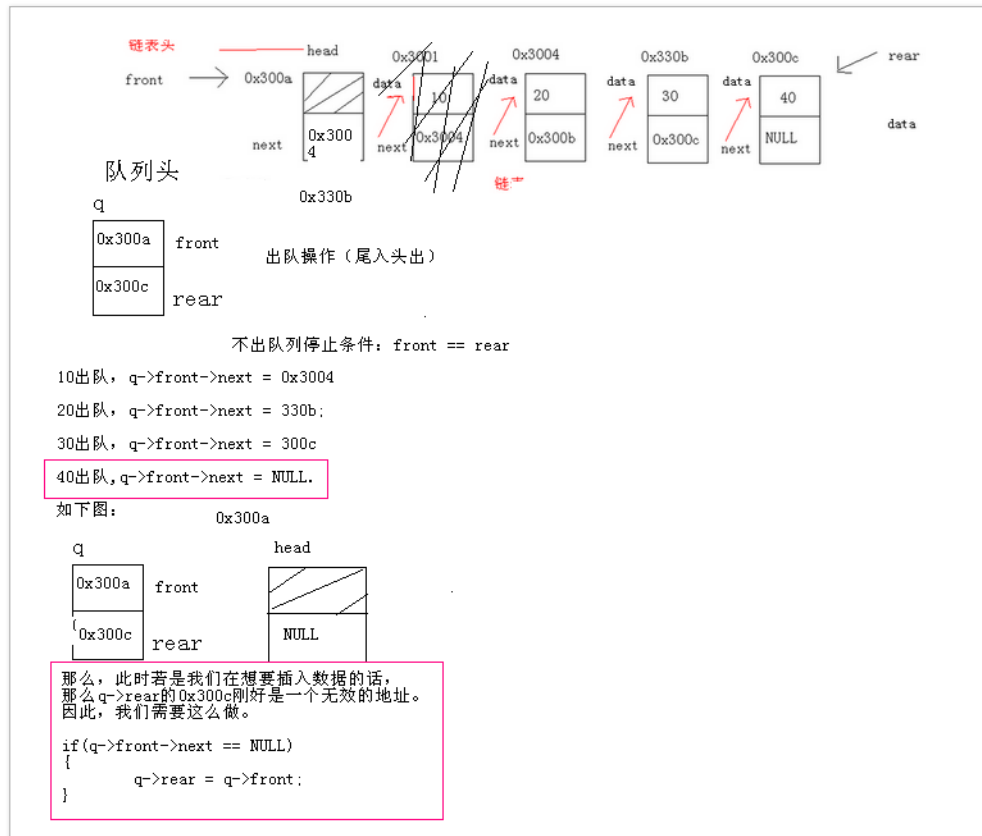
“ 慕课网慕课教程 4.3 链式队列操作涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

链式队列： 插入操作在队尾进行，删除操作在队头进行，由队头指针和队尾指针控制队列的操作。



队列头 + 带头节点的链表。





```
typedef int data_t;
```

```
typedef struct node
{
    data_t data;
    struct node *next;
}linknode_t;
```

```
typedef struct
{
    linknode_t *front;
    linknode_t *rear;
}linkqueue_t;
```

```
linkqueue_t *create_empty_linkqueue()
{
    linkqueue_t *q = NULL;
    linknode_t *head = NULL;

    head = (linknode_t *)malloc(sizeof(linknode_t));
    head->next = NULL;

    q = (linkqueue_t *)malloc(sizeof(linkqueue_t));

    q->front = q->rear = head;

    return q;
}
```

```
int is_empty_linkqueue(linkqueue_t *q)
{
    return q->front == q->rear ? 1 : 0;
}
```

```
void enter_linkqueue(linkqueue_t *q, data_t data)
{
    linknode_t *temp = NULL;
```

```
temp = (linknode_t *)malloc(sizeof(linknode_t));
temp->data = data;

temp->next = q->rear->next;
q->rear->next = temp;

q->rear = temp;

return ;
}

data_t delete_linkqueue(linkqueue_t *q)
{
    linknode_t *temp = NULL;
    data_t data;

    temp = q->front->next;
    data = temp->data;

    q->front->next = temp->next;
    free(temp);
    temp = NULL;

    if(q->front->next == NULL)
    {
        q->rear = q->front;
    }

    return data;
}

int main()
{
    linkqueue_t *q = NULL;
    int i = 0;

    q = create_empty_linkqueue();

    for(i = 0; i < 10; i++)
    {
        enter_linkqueue(q, i);
    }

    while(!is_empty_linkqueue(q))
    {
        printf("%d ", delete_linkqueue(q));
    }
    printf("\n");

    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

