

# Report-13-b5

学号：1850772 姓名：张哲源

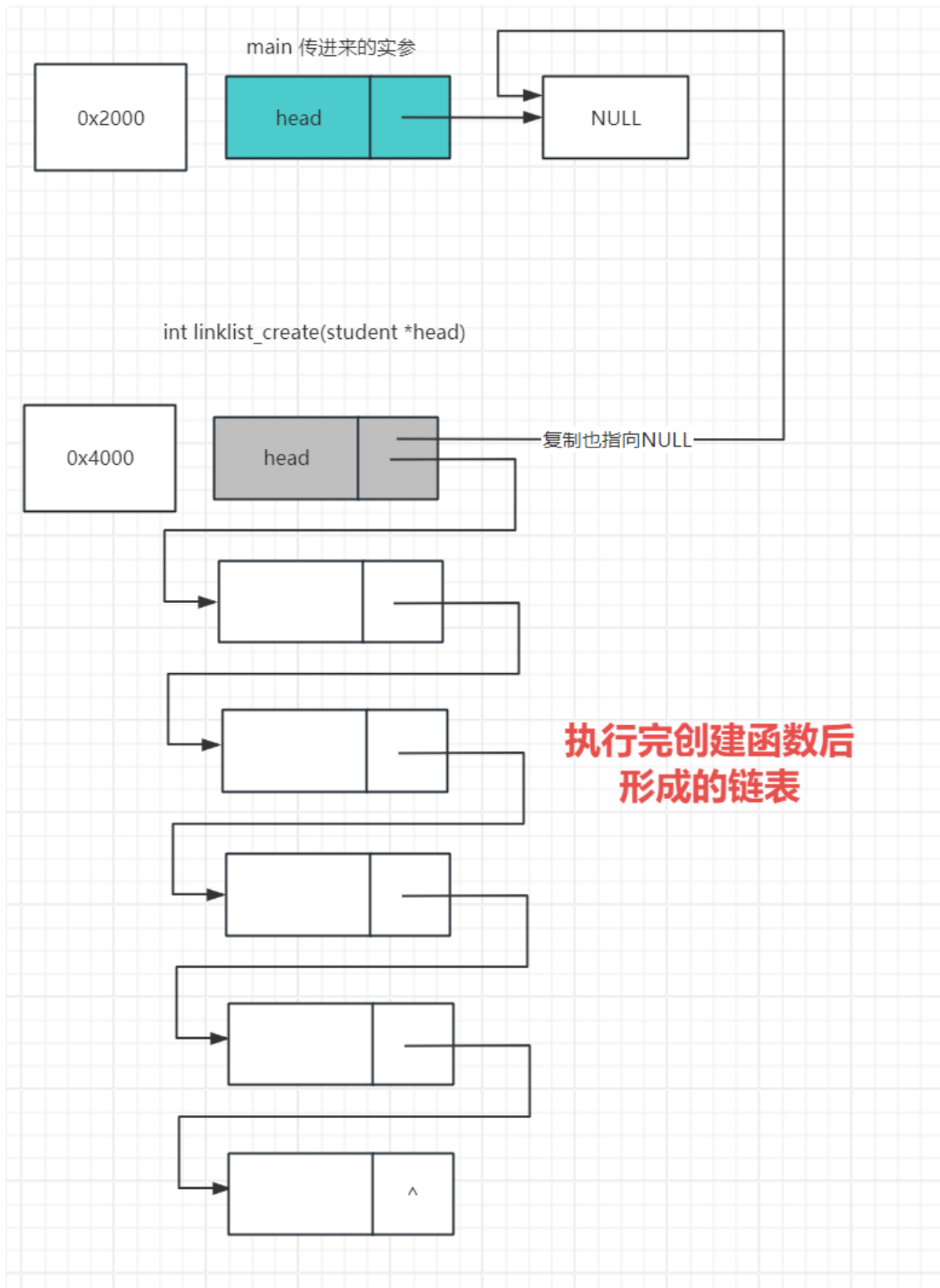
## 1. 链表的建立是否正确

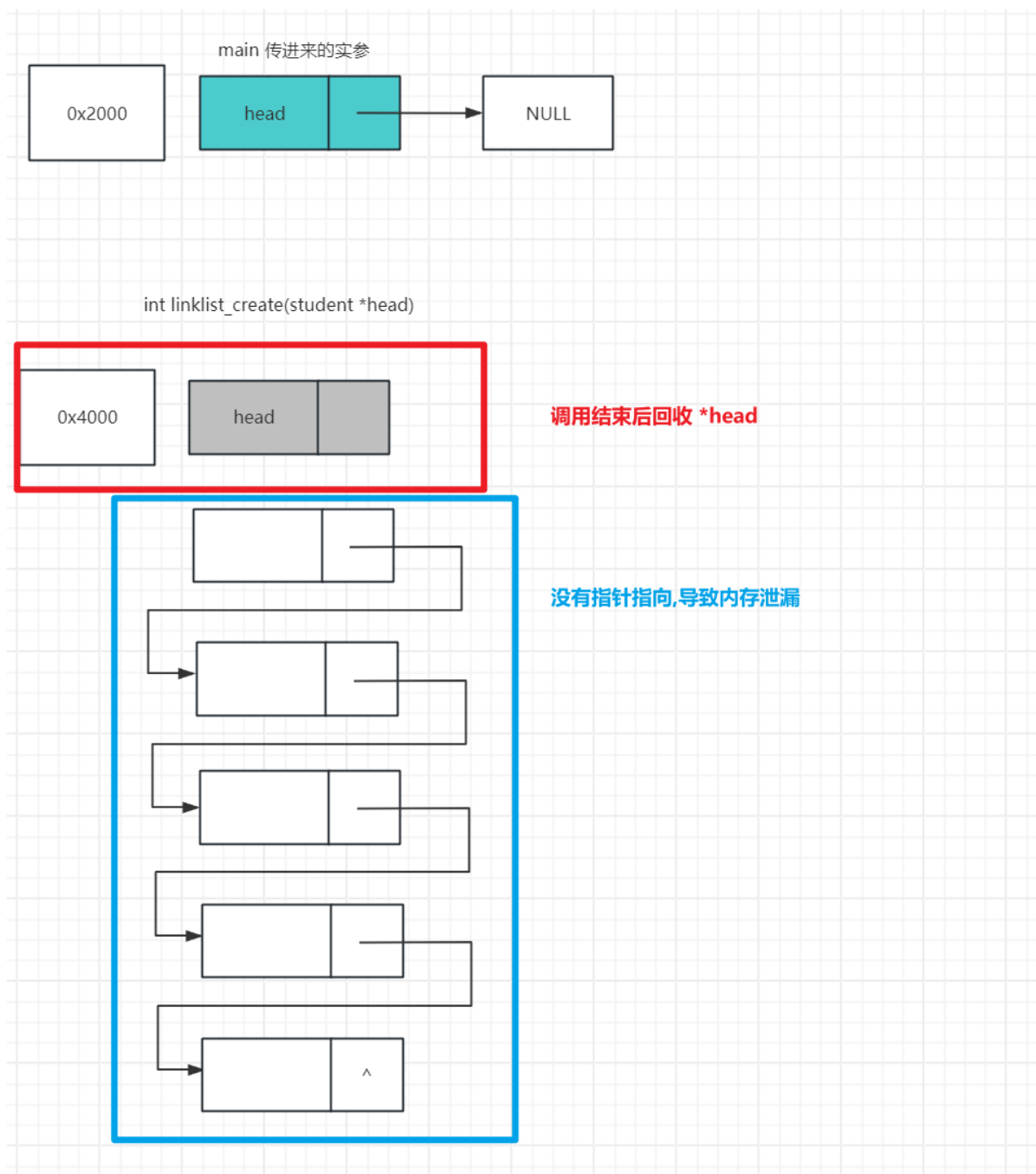
链表的建立有一个问题，即 `linklist_create` 函数的参数 `head` 传递方式。

虽然 `head` 是传递进来的指针，但在函数内部，它被重新赋值为新分配的节点，这意味着在 `main` 函数中 `head` 不会指向有效的链表。

应该将 `head` 传递的指针的地址，或者使用双重指针，以便在函数内部可以修改 `head` 的指向。

内存图如下





## 2. 为什么遍历不成功

由于 `head` 在 `linklist_create` 中的问题, 导致在 `main` 函数中无法正确遍历链表。

即使在 `linklist_create` 中创建了链表, 由于参数传递的问题, `head` 在 `main` 中仍然为 `NULL`。

## 3. 链表的销毁是否成功

不考虑链表建立不正确的情况下

链表的销毁函数正确

## 4. 程序内存丢失的情况

有内存泄漏的情况。

虽然 `linklist_destroy` 函数尝试释放节点的内存，但链表的头指针 `head` 在 `main` 函数中丢失了

这意味着没有办法正确释放整个链表的内存。

此外，还存在在 `linklist_create` 中未释放之前的链表节点的问题。

## 5.

### 5.1 修改正确的程序

方法1, 引用传参(C++ 方式, 批阅时请忽视这一条):

第一处修改

```
1 // 原函数
2 int linklist_create(student *head)
3 {
4     ...
5 }
```

改为 引用传参

```
1 int linklist_create(student& head)
2 {
3     ...
4 }
```

第二处修改

```
1 int linklist_destroy(student *head)
2 {
3     ...
4     return OK;
5 }
```

改为 引用传参 , 同时也确保头结点的释放

```
1 int linklist_destroy(student& head)
2 {
3     ...
4
5     head = nullptr; // Set head to nullptr after destruction
6     return OK;
7 }
```

第三处和第四处 修改函数声明

```

1 | int linklist_create(student *head);
2 | int linklist_traverse(student *head);
3 | int linklist_destroy(student *head);

```

改为

```

1 | int linklist_create(student*& head);
2 | int linklist_traverse(student* head);
3 | int linklist_destroy(student*& head);

```

## 方法2 , 指向指针的指针(C方式)

第一处修改 修改函数声明 ,形参为指向指针的指针

```

1 | int linklist_create(student *head);
2 | int linklist_traverse(student *head);

```

改为

```

1 | int linklist_create(student **head);
2 | int linklist_traverse(student *head);

```

第二处和第三处修改 修改创建链表函数

第13 行的 head 变为 \*head

函数的形式参数 \*\*head 改为 \*head

```

1 | int linklist_create(student *head)
2 | {
3 |     student *p = NULL, *q = NULL;
4 |     int i;
5 |
6 |     for (i = 0; i < 5; i++) {
7 |         if (i > 0)
8 |             q = p;
9 |         p = new(nothrow) student; //思考: 为什么不能用malloc
10 |        if (p == NULL)
11 |            return ERROR;    //注:此处未释放之前的链表节点,就借助操作系统来释放(非
    标用法)
12 |        if (i == 0)
13 |            head = p; //head指向第1个结点
14 |        else
15 |            q->next = p;
16 |        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
17 |        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
18 |        p->next = NULL;
19 |    }
20 |    return OK;
21 | }

```

改为

```

1 | int linklist_create(student** head)
2 | {

```

```

3      student* p = nullptr, * q = nullptr;
4      int i;
5
6      for (i = 0; i < 5; i++) {
7          if (i > 0)
8              q = p;
9          p = new(nothrow) student;
10         if (p == nullptr)
11             return ERROR;
12         if (i == 0)
13             *head = p;
14         else
15             q->next = p;
16         cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
17         cin >> p->name >> p->num >> p->sex;
18         p->next = nullptr;
19     }
20     return OK;
21 }

```

#### 第四处修改

main函数传参的位置, `head` 改为 `&head`

```

1  int main()
2  {
3      ...
4      if (linklist_create(head) == OK) {
5          linklist_traverse(head);
6          ...
7      }

```

改为

```

1  int main()
2  {
3      ...
4      if (linklist_create(&head) == OK) {
5          linklist_traverse(head);
6          ...
7      }

```