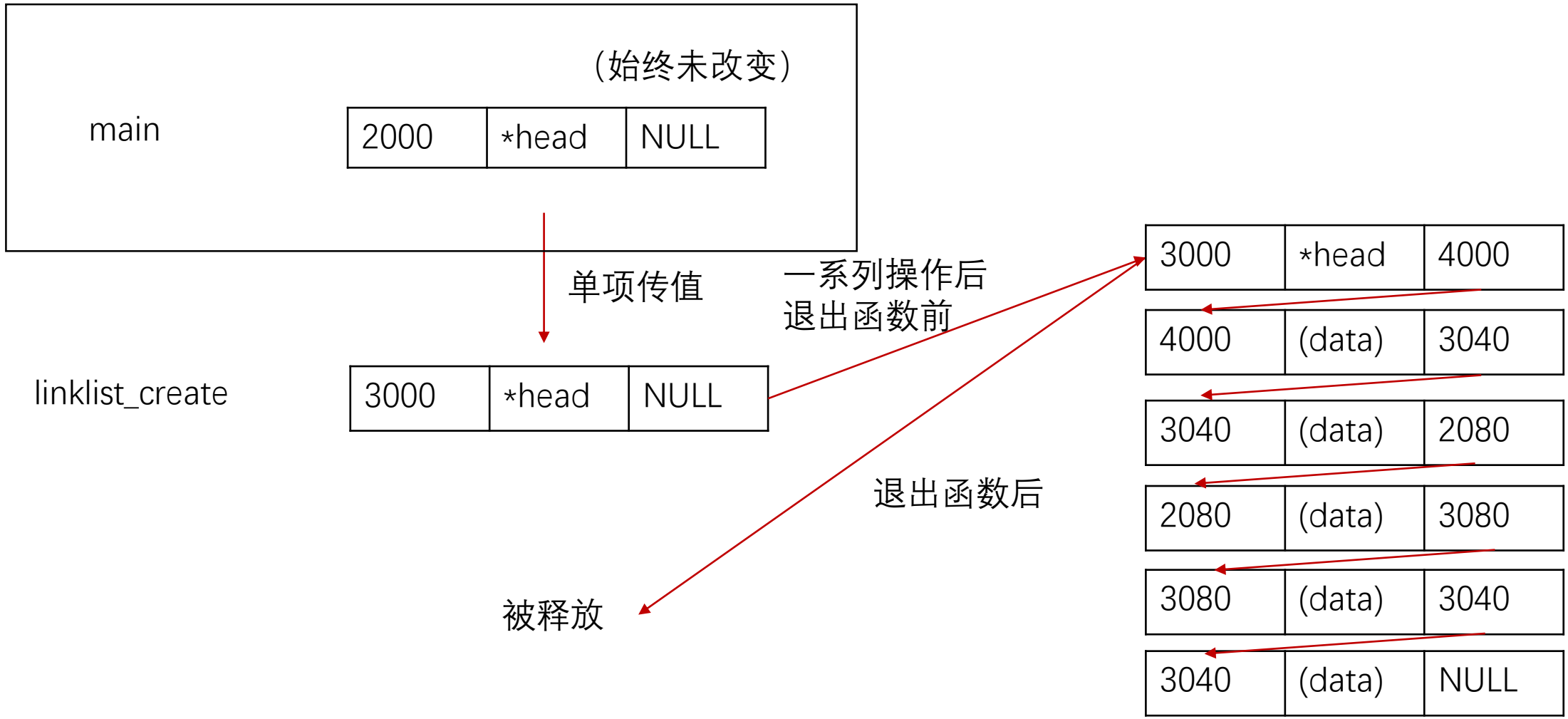
7-b6答案

1850059 计1班 杨志远

1、链表的建立是否正确?

- 不正确。
- 函数linklist_create中的一系列操作是正确的，也就是说在linklist_create中，链表的建立是正确的。但函数修改的head指针是作为实参直接传入函数的，因此在退出函数时，形参被释放，main函数中的head并没有被修改。这就导致了整个链表都被丢失在内存中，无法被程序访问，直到操作系统采取行动才被释放。
- 以下是图解。

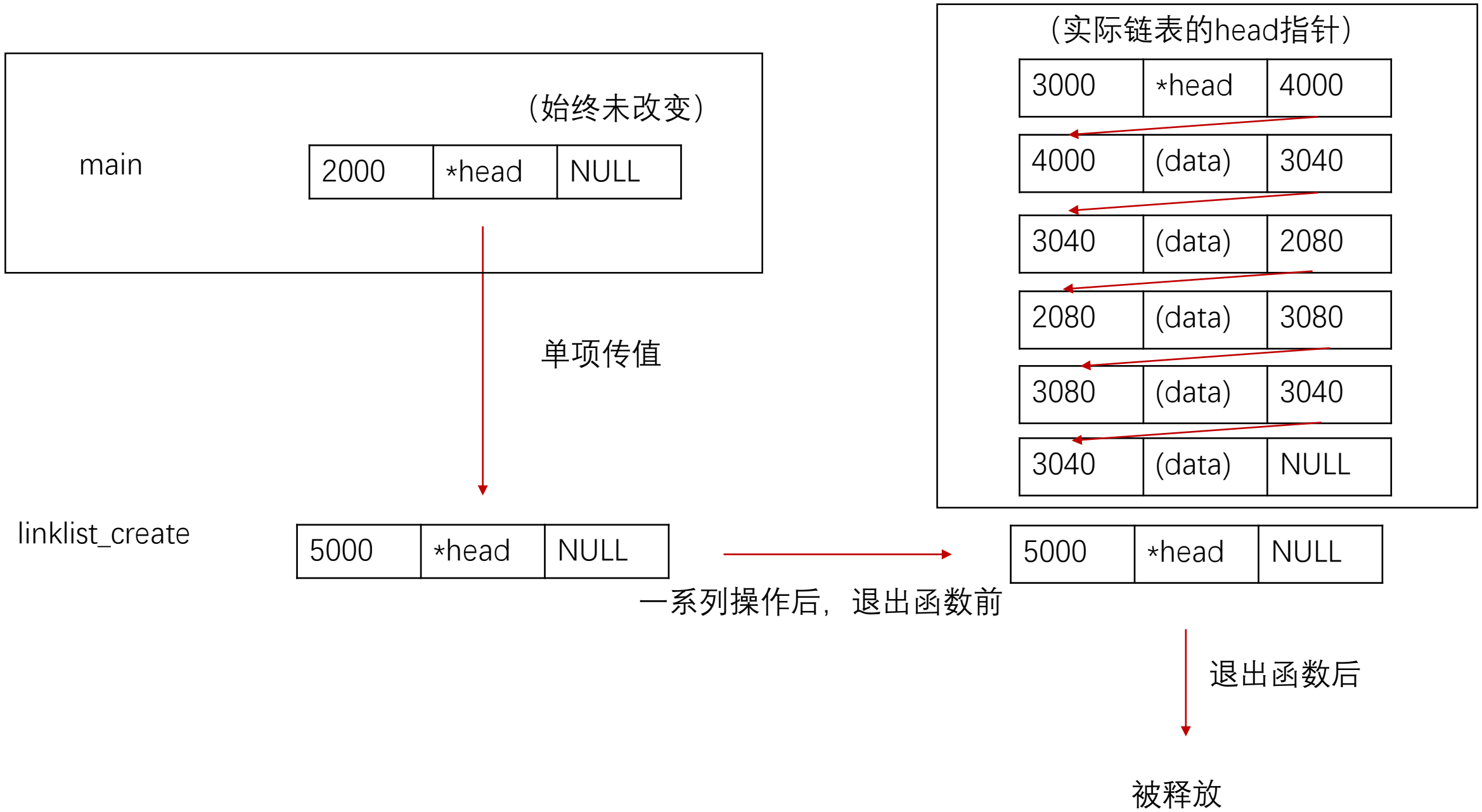

（始终未改变）
main
2000
*head
NULL
单项传值
linklist_create
3000
*head
NULL
一系列操作后
退出函数前
退出函数后
被释放
3000
*head
4000
4000
(data)
3040
3040
(data)
2080
2080
(data)
3080
3080
(data)
3040
3040
(data)
NULL

2、为什么遍历不成功?

- 由于链表的建立发生了错误，main函数中的head指针的值始终是NULL，这就导致了传入linklist_traverse函数中的实参head的值同样是NULL。而linklist_traverse函数中，在判断出head是空指针后，函数没有输出任何信息就返回了，因此遍历失败了。

3、链表的销毁是否成功了？

- 失败。
- 由于main函数的head指针始终是空指针，因此传入linklist_destroy的实参head同样是空指针。在执行到while语句时，因为不符合条件，整个while语句都没有被执行，函数就返回了。而最初linklist_create函数建立的链表并没有被销毁，因此就导致了内存泄漏的问题。
- 以下是图示。


main
(始终未改变)
2000
*head
NULL
单项传值
linklist_create
5000
*head
NULL
一系列操作后，退出函数前
(实际链表的head指针)
3000
*head
4000
4000
(data)
3040
3040
(data)
2080
2080
(data)
3080
3080
(data)
3040
3040
(data)
NULL
5000
*head
NULL
退出函数后
被释放

4、程序是否有内存丢失情况？如果有，发生在哪个函数被调用的阶段?

- 有。
- 正如之前所述，在函数linklist_create调用完成并返回后，建立的链表就由于head指针的指向出现了问题，在整个程序的生存周期中无法被任何方法调用，也就无法被访问。链表占据的这部分内存就被丢失，直到被操作系统回收。
- 图示与第3题相同。

5、如何改动，能使程序正确?

- 一共修改四处，都与函数linklist_create的参数有关。
- Line16: int linklist_create(student *head)
- => int linklist_create(student **head)
- Line20: int linklist_create(student *head)
- => int linklist_create(student **head)
- Line32: head = p
- => *head = p
- Line71: if (linklist_create(head) == OK) {
- => if (linklist_create(&head) == OK) {