**今天的工作**

继续完善alias和unalias，**修改pipe和文件重定向。完成&后台功能（没完成）**

**遇到的问题**

1. 怎么判断chr型指针数组有没有被初始化
2. ‘’和“”的区别
3. 测试时发现之前写的函数有很多不必要的传参
4. 发现新传入的alias关系会覆盖掉之前的alias关系
5. 关于unalias怎么判断要取消的别名是否存在
6. 关于unalias怎么删除节点

**解决的问题**

1. 如果是全局变量，可以判断指针是否为空
2. ‘’里的会是\0结尾，””里的会转化成ASCII码
3. 修改了函数的原型，去掉了不必要的传参，比如说遍历链表的时候并不用传头结点和当前节点的指针，因为是全局变量啊！直接就可以赋值了，真的不知道自己之前在想什么
4. 这是因为currentRecord=head;这句的初始化写在了alias()的第一句，每次执行都会从链表头开始存，自然就覆盖了。所以改成在shell入口处就初始化，这样就没问题了。真的是坑啊，主要还是太大意了。顺便还检查出遍历链表函数那边一开始flag值没有清0，一定要清0不然结果会有问题的。
5. unalias好像很简单，因为也是遍历啥的，但还是跳了一些坑，比如说判断要取消的别名是否存在，一开始想的是遍历，遍历完如果没有找到就打印not found，但是这样会造成一个结果。。。遍历永远不结束，具体为什么我还没想清楚，while循环结束的条件是p==null，但是就是不结束，我猜是因为没有初始化不代表=null。于是只能换一种方法就是用之前写的遍历链表找值返回flag的函数if\_alias，在一开始调用if\_alias，如果flag值为0说明这个别名不存在，打印not found，然后再进行遍历链表，删除节点。
6. 这里也踩了好多坑——？一开始是想tmp=p->next但是可能有些累赘了，如果是删后面的几条别名就会一直遍历，理由还未知，看起来也不方便。用p->next->next也很好？这一部分其实可以再研究一下，还有就是没有free。所以其实还有待研究。

**待进一步解决**

unalias更好的实现

**关键代码**

**1.unalias实现**

/\*\*\*\*\* Info \*\*\*\*\*/

/\* Author: ZHJ \*/

/\* Function: 实现unalias\*/

void unalias\_zhj(){

if\_alias(commandCompose[1]);

if(flag==0){

printf("%s:not found",commandCompose[1]);

return;

}

char deletestr[MAX\_CMD\_LEN/2+1];

strcpy(deletestr,commandCompose[1]);

//char deletestr[50]="ls";

struct Node \*p=head;

//struct Node \*tmp=(struct Node\*)malloc(sizeof(struct Node));

//tmp=p->next;

//p=p->next;

while((p->next)!=NULL){

if(!strcmp(deletestr,((p->next)->data).word)){

p->next=(p->next)->next;

//printf("success");

//free(p->next);

break;

}

p=p->next;

}

}

**2.判断用户输入的是不是别名judge\_alias()**

void judge\_alias(){

flag=0;

//printf("%s\n",commandCompose[0]);

if\_alias(commandCompose[0]);

//printf("%d\n",flag);

if(flag==1){

//printf("substitute:%s\n",substitute\_result);

strcpy(commandCompose[0],substitute\_result);

/\*在这里将替换的指令copy给cmd\_array[]再进行分析\*/

}

//flag=0;

//printf("%d\n",flag);

}

**其他想法**

回家真的快乐，在家里就是舒服，就算被蚊子咬也舒服，今天看了黑天鹅，唉，人还是不能太逼迫自己。。。

**明天的计划**

今天主要是搞定了unalias和shell的兼容使用问题，其他的都还没搞，组长好像有了新的plan，明天看看再决定。暂时是想要先实现一些简单的功能比如说cat echo这种~