

Lab3实验报告

代码部分

lab3.c

```
#include <stdio.h>
#include <string.h>
#define MAX 15

//冒泡排序
extern void strbubble(char*,char*,int); //给循环调用的相邻数组两两比较排序
extern void strsort(char*,int,int);    //循环

int main(void){
    int i,n;
    scanf("%d",&n);
    char a[n][MAX];
    int temp;
    for(i=0;i<n;i++)
    {
        scanf("%s",a[i]);          //不会读进回车，也不会使下一次开头读入上一次的回车
    }                                //输入
    strsort(a[0],n-1,MAX);          //调用排序函数
    printf("\n");
    for(i=0;i<n;i++)
    {
        printf("%s\n",a[i]);        //输出
    }
    return 0;
}
```

strsort.s

```
.global strsort
.extern strbubble          //调用相邻数组两两比较排序
.text
.arm
strsort:
    mov r3,#-1              //进入loop1后即初始为i=0
loop1:
    add r3,r3,#1
    cmp r3,r1               //i<n-1吗?
    bge end                 //不符合循环条件，跳出循环
    mov r4,#0               //j=0
    mov r6,r0               //a[j]地址，初始j=0
    add r7,r0,r2            //a[j+1]地址，初始j=0，若n=1则会直接跳出循环，不会到这一步产生越界错误
    sub r5,r1,r3            //n-1-i
loop2:
    cmp r4,r5               //j<n-1-i吗?
    bge loop1               //不符合循环条件，跳出内循环
    stmfd sp!,{r0-r7,lr}    //保留现场
    mov r0,r6
    mov r1,r7
```

```

b1 strbubble //strbubble(a[j],a[j+1],MAX),r2一直存的都是MAX
ldmfd sp!,{r0-r7,lr} //恢复现场
add r4,r4,#1
cmp r4,r5 //j+1后立即比较，否则a[j+1]在i=0的情况下会越界
bge loop1
add r6,r6,r2
add r7,r7,r2 //j++后，a[j]和a[j+1]指向的地址各自向后移动一个MAX长度
b loop2 //回到内循环开头，此时其实可以不用在开头再compare，因为实际执行很快所以不优化了
end:
mov PC,LR
.end

//本质实现的代码是：
//for(i=0;i<n-1;i++)
//for(j=0;j<n-1-i;j++)
//strbubble(a[j],a[j+1],MAX);

```

strbubble.s

```

//两个字符串按对应位置两两比较，将大的字符串整个交换到小的后面，不改变整体地址值
.global strbubble
.text
.arm
strbubble:
    mov r3,r0 //r3为数组1基址（存在r0）
    mov r4,r1 //r4为数组2基址（存在r1）
    mov r5,#0 //r5为偏移量
compare:
    ldrb r6,[r3] //char类型占一个字节
    ldrb r7,[r4]
    cmp r6,r7
    blt swap //数组1小于数组2则交换
    bgt end //数组1大于数组2直接结束
    add r5,r5,#1
    cmp r5,r2
    beq end //偏移量等于MAX（存在r2）时比较结束，两数组相等
    add r3,r3,#1 //不等于MAX时继续向后比较，char类型占一个字节，所以加1
    add r4,r4,#1
    b compare
swap:
    mov r3,r0 //重新获取基址，偏移量计0，开始交换
    mov r4,r1
    mov r5,#0
loop:
    ldrb r6,[r3]
    ldrb r7,[r4]
    strb r7,[r3] //char类型占一个字节
    strb r6,[r4]
    add r3,r3,#1
    add r4,r4,#1
    add r5,r5,#1
    cmp r5,r2
    beq end //等于MAX时交换完毕
    b loop
end:
    mov PC,LR //返回
.end

```

具体解释用注释标出了。

运行结果

```
[root@FORLINUX6410]# ./lab3
7
12345678901
asdhkgjeutiuh
1354897651354
4567531379845k
159753123456
tukn8h46247
7777777777

tukn8h46247
asdhkgjeutiuh
7777777777
4567531379845k
159753123456
1354897651354
12345678901
```

达到预期要求。