

字符串排序的内容复制与序号交换的比较

(1) 内容复制版

```
#include<stdio.h>
#include<string.h>
#define STR_COUNT 100000
#define STR_LEN 100

int main() {
    time_t t1,t2;
    int i,j;
    char (*str)[STR_LEN];
    char s[100];
    int index[STR_COUNT];
    srand(100);
    str=malloc((STR_COUNT*STR_LEN));
    printf("Line %d\n",__LINE__);
    for(i=0; i<STR_COUNT; i++) {
        for (j=0; j<STR_LEN; j++)
            str[i][j]=rand()%95+32;
        str[i][STR_LEN-1]=0;
    }
    printf("Line %d\n",__LINE__);
    time(&t1);
    for(i=0; i<STR_COUNT; i++) {
        for(j=0; j<STR_COUNT-1-i; j++)
            if(strcmp(str[j],str[j+1])>0) {
                strcpy(s,str[j]);
                strcpy(str[j],str[j+1]);
                strcpy(str[j+1],s);
            }
    }
    printf("Line %d\n",__LINE__);
    time(&t2);
    free(str);
    printf("time=%u\n",t2-t1);
    // for(i=0; i<STR_COUNT; i++)
    //     puts(str[i]);
    return 0;
}
```

(2) 序号交换版

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define STR_COUNT 100000
#define STR_LEN 100

int main() {
    time_t t1,t2;
    int i,j,temp;
    char (*str)[STR_LEN];
    char s[100];
    int index[STR_COUNT];
    srand(100);
    str=malloc((STR_COUNT*STR_LEN));
    printf("Line %d\n",__LINE__);
    for(i=0; i<STR_COUNT; i++) {
        for (j=0; j<STR_LEN; j++)
            str[i][j]=rand()%95+32;
        str[i][STR_LEN-1]=0;
    }
    for(i=0; i<STR_COUNT; i++)
        index[i]=i;
    printf("Line %d\n",__LINE__);
    time(&t1);
    for(i=0; i<STR_COUNT; i++)
        for(j=0; j<STR_COUNT-1-i; j++)
            if(strcmp(str[index[j]],str[index[j+1]])>0) {
                temp=index[j];
                index[j]=index[j+1];
                index[j+1]=temp;
            }
    time(&t2);
    printf("time=%u\n",t2-t1);
    // for(i=0; i<STR_COUNT; i++)
    //     puts(str[index[i]]);
    return 0;
}
```

通过实际运行可知，后者运行时间大大缩短。