

《计算机系统基础》Homework

HW0：排序程序的编辑、编译和调试

实验目的：熟悉开发环境、掌握开发和调试的基本过程以及工具。

实验要求：对实验步骤中给出的源程序进行编辑、编译、链接，调试。

实验报告：

1. 说明你做实验的过程（重要步骤用屏幕截图表示）。
2. 提交出源程序。
3. 提交可执行目标文件。
4. 分析或回答下列问题。
 - （1）分析同一个源程序在不同机器上生成的可执行目标代码是否相同。
提示：从多个方面（如ISA、OS 和编译器）来分析。
 - （2）你能在可执行目标文件中找出函数 printf ()对应的机器代码段吗？能的话，请标示出来。
 - （3）为什么源程序文件的内容和可执行目标文件的内容完全不同？

报告提交截止日期：9月10日

实验步骤：

- 1、根据提供的图片编辑文本，并保存为相应的.c 和.h 文件。

该程序主要写了排序、求和算法，程序源码如下图所示：

bubblesort.h:

```
1 #include <stdio.h>
2 void bubblesort(int s[],int n);
```

bubblesort.c:

```
1 #include "bubblesort.h"
2 void bubblesort(int s[],int n){
3     int i,j,t;
4     for (i=1;i<n;i++)
5     {
6         for (j=1;j<n+1-i;j++)
7         {
8             if (s[j]>s[j+1])
9             {
10                t=s[j];
11                s[j]=s[j+1];
12                s[j+1]=t;
13            }
14        }
15    }
16 }
```

add.h:

```
1 #include <stdio.h>
2 int add(int s[],int n);
```

add.c:

```

1  #include "add.h"
2  int add(int s[],int n){
3      int result=0;
4      int i;
5      for (i=1;i<=n;i++)
6      {
7          result+=s[i];
8      }
9      return result;
10 }

```

printresult.h:

```

1  #include <stdio.h>
2  void printArray(int s[],int n,char* str);

```

printresult.c:

```

1  #include "printresult.h"
2  void printArray(int s[],int n,char* str){
3      printf("%s",str);
4      int i;
5      for (i=1;i<11;i++)
6      {
7          printf("%5d",s[i]);
8      }
9      printf("\n");
10 }

```

main.c:

```

1  #include "add.h"
2  #include "bubblesort.h"
3  #include "printresult.h"
4  #define bool char
5  #define true 1
6  #define false 0
7  void main(){
8      int a[11],i;
9      int b[11];
10     printf("请输入10个数据：\n");
11     for (i=1;i<=10;i++)
12     {
13         scanf("%d",&a[i]);
14         b[i]=a[i];
15     }
16     bool flag=true;
17     while (flag==true)
18     {
19         printf("\n1.冒泡排序\n2.求和\n3.打印排序结果\n4.退出");
20         printf("\n请选择排序序号：");
21         int number=0;
22         scanf("%d",&number);
23         int sum=0;
24         switch (number)
25         {
26             case 1:

```

```

26         case 1:
27             bubblesort(a,10);
28             break;
29         case 2:
30             sum=add(a,10);
31             printf("\n数组求和结果为 : %5d\n",sum);
32             break;
33         case 3:
34             printArray(b,10,"\n原始数组为 : ");
35             printArray(a,10,"\n排序后数组为 : ");
36             break;
37         case 4:
38             flag=false;
39             break;
40         default:
41             printf("\n请选择正确的序号！请继续！");
42             break;
43     }
44 }
45 printf("\n");
46 }

```

1. (1) 使用 gcc 直接编译为执行文件:

gcc -o main main.c bubblesort.c add.c printresult.c

- (2) 现编译再连接:

1. 首先使用 gcc -c main.c 语句为所有.c 文件编译.o 目标文件, 可以用-o 命令命名输出目标文件。

2. 用 ld 指令链接目标文件:

```

ld -o main main.o bubblesort.o add.o printresult.o -e main --sysroot=/ --build-id
--eh-frame-hdr -m elf_i386 --hash-style=gnu --as-needed -dynamic-linker /lib/ld-linux.so.2 -z
relro /usr/lib/gcc/i686-linux-gnu/4.8/../../../../i386-linux-gnu/crt1.o
/usr/lib/gcc/i686-linux-gnu/4.8/../../../../i386-linux-gnu/crti.o
/usr/lib/gcc/i686-linux-gnu/4.8/crtbegin.o -L/usr/lib/gcc/i686-linux-gnu/4.8
-L/usr/lib/gcc/i686-linux-gnu/4.8/../../../../i386-linux-gnu
-L/usr/lib/gcc/i686-linux-gnu/4.8/../../../../lib -L/lib/i386-linux-gnu -L/lib/./lib
-L/usr/lib/i386-linux-gnu -L/usr/lib/./lib -L/usr/lib/gcc/i686-linux-gnu/4.8/../../../../.. -lgcc
--as-needed -lgcc_s --no-as-needed -lc -lgcc --as-needed -lgcc_s --no-as-needed
/usr/lib/gcc/i686-linux-gnu/4.8/crtend.o
/usr/lib/gcc/i686-linux-gnu/4.8/../../../../i386-linux-gnu/crtn.o

```

(ld 链接时要包含很多系统库, 可以用 gcc -v main.c 来查看系统链接需要哪些库, 把 collect2 换成 ld, 生成的/tmp/ccBCU0rh.o 即为 main.c 编译出来的 main.o 文件, 删掉该句替换成-o main main.o bubblesort.o add.o printresult.o -e main), 如图:

```

ericson@ericson-virtual-machine:~/wenjian3$ gcc -c main.c
ericson@ericson-virtual-machine:~/wenjian3$ gcc -c bubblesort.c
ericson@ericson-virtual-machine:~/wenjian3$ gcc -c add.c
ericson@ericson-virtual-machine:~/wenjian3$ gcc -c printresult.c
ericson@ericson-virtual-machine:~/wenjian3$ ld -o main main.o bubblesort.o add.o
printresult.o -e main --sysroot=/ --build-id --eh-frame-hdr -m elf_i386 --hash-
style=gnu --as-needed -dynamic-linker /lib/ld-linux.so.2 -z relro /usr/lib/gcc/i
686-linux-gnu/4.8/../../../../i386-linux-gnu/crt1.o /usr/lib/gcc/i686-linux-gnu/4.8
/../../../../i386-linux-gnu/crti.o /usr/lib/gcc/i686-linux-gnu/4.8/crtbegin.o -L/us
r/lib/gcc/i686-linux-gnu/4.8 -L/usr/lib/gcc/i686-linux-gnu/4.8/../../../../i386-lin
ux-gnu -L/usr/lib/gcc/i686-linux-gnu/4.8/../../../../lib -L/lib/i386-linux-gnu -
L/lib/./lib -L/usr/lib/i386-linux-gnu -L/usr/lib/./lib -L/usr/lib/gcc/i686-lin
ux-gnu/4.8/../../../../lgcc --as-needed -lgcc_s --no-as-needed -lc -lgcc --as-need
ed -lgcc_s --no-as-needed /usr/lib/gcc/i686-linux-gnu/4.8/crtend.o /usr/lib/gcc/
i686-linux-gnu/4.8/../../../../i386-linux-gnu/crtn.o
ericson@ericson-virtual-machine:~/wenjian3$ ./main
请输入10个数据:
10 9 8 7 6 5 4 3 2 1

1.冒泡排序
2.求和
3.打印排序结果

```

2. 反汇编:

使用 `objdump` 指令进行反汇编:

`objdump -S main.o`

`objdump` 命令详见文档

3. 调试:

使用 `gdb` 命令调试, 调试之前首先用 `gcc -g` 命令来生成调试信息, 否则调试失败:

`gcc -g -o main main.c bubblesort.c add.c printresult.c`

`gdb main`

`gdb` 命令详见文档

用 `info registers` 查看寄存器内容。