

## HW2: 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. 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？

实验步骤：

1. 以下程序实现了**选择排序**和求和算法，程序源码如下所示。请补全代码，完成功能。

selectionSort.h

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

selectionSort.c

```
1 #include "selectionSort.h"
2 void selectionSort(int s[], int n){
3     // TODO: write your code here
4 }
```

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     // TODO: write your code here
4 }
```

#### printArray.h

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

#### printArray.c

```
1 #include "printArray.h"
2 void printArray(int s[], int n, char *str){
3     printf("%s", str);
4     int i;
5     for(i = 0; i < n; i++){
6         printf("%5d", s[i]);
7     }
8     printf("\n");
9 }
```

#### main.c

```
1 #include "add.h"
2 #include "selectionSort.h"
3 #include "printArray.h"
4 #include <time.h>
5
6 #define bool char
7 #define true 1
8 #define false 0
9 #define LENGTH 10
10
11 int main(){
12     int a[LENGTH], i;
13     int b[LENGTH];
14     int randValue = 0;
15     srand(time(NULL));
16
17     for(i = 0; i < LENGTH; i++){
18         randValue = 1 + (int)rand() % LENGTH;
19         a[i] = randValue;
20         b[i] = a[i];
21     }
22
23     printArray(a, LENGTH, "\nrandom array: ");
24
25     bool flag = true;
26     while(flag){
27         printf("\n1.selection sort\n2.sum\n3.print result\n4.exit");
28         printf("\nchoose a number:");
29
30         int number = 0;
```

```

31     scanf("%d", &number);
32
33     int sum = 0;
34     switch(number)
35     {
36         case 1:
37             selectionSort(a, LENGTH);
38             break;
39         case 2:
40             sum = add(a, LENGTH);
41             printf("\nresult of sum: %d\n", sum);
42             break;
43         case 3:
44             printArray(b, LENGTH, "\noriginal array:\t");
45             printArray(a, LENGTH, "\nsorted array:\t");
46             break;
47         case 4:
48             flag = false;
49             break;
50         default:
51             printf("\nplease choose a correct number and continue!");
52             break;
53     }
54
55     printf("\nDone!\n\n");
56 }
57
58 return 0;
59 }

```

2. 将源程序文件进行预处理、编译、汇编和链接，以生成可执行文件。以下为生成可执行文件的两种方式：

1) 使用 `gcc` 直接生成可执行文件 `gcc -o main main.c selectionSort.c add.c printArray.c`

2) 首先，使用 `gcc -c` 命令将所有 `.c` 文件编译成 `.o` 文件（可以用 `-o` 选项命名输出的可重定位目标文件），然后再用 `ld` 命令进行链接，以生成可执行目标文件。（用 `ld` 命令链接时要包含很多系统库，可以用 `gcc -v main.c` 来查看需要哪些库，如下图所示：

```

/usr/lib/gcc/x86_64-linux-gnu/4.8/collect2 --sysroot=/ --build-id --eh-frame-hdr -m
elf_x86_64 --hash-style=gnu --as-needed -dynamic-linker /lib64/ld-linux-x86-64.so.2 -
z relro /usr/lib/gcc/x86_64-linux-gnu/4.8/../../../../x86_64-linux-gnu/crt1.o /usr/lib/g
cc/x86_64-linux-gnu/4.8/../../../../x86_64-linux-gnu/crti.o /usr/lib/gcc/x86_64-linux-gn
u/4.8/crtbegin.o -L/usr/lib/gcc/x86_64-linux-gnu/4.8 -L/usr/lib/gcc/x86_64-linux-gnu/
4.8/../../../../x86_64-linux-gnu -L/usr/lib/gcc/x86_64-linux-gnu/4.8/../../../../lib -L/
lib/x86_64-linux-gnu -L/lib/../lib -L/usr/lib/x86_64-linux-gnu -L/usr/lib/../lib -L/u
sr/lib/gcc/x86_64-linux-gnu/4.8/../../../../tmp/cc0w72W6.o -lgcc --as-needed -lgcc_s --
no-as-needed -lc -lgcc --as-needed -lgcc_s --no-as-needed /usr/lib/gcc/x86_64-linux-g
nu/4.8/crtend.o /usr/lib/gcc/x86_64-linux-gnu/4.8/../../../../x86_64-linux-gnu/crtn.o

```

图中便是需要链接的系统库，红框处是 `gcc` 产生的临时文件，手动链接时要排除这些临时文件，命令如下  
图：

```
~/Desktop/homework/2 gdb ➤ ld -o main main.o add.o selectionSort.o printArray.o --sy  
sroot=/ --build-id --eh-frame-hdr -m elf_x86_64 --hash-style=gnu --as-needed -dynamic  
-linker /lib64/ld-linux-x86-64.so.2 -z relro /usr/lib/gcc/x86_64-linux-gnu/4.8/../../../../  
../x86_64-linux-gnu/crt1.o /usr/lib/gcc/x86_64-linux-gnu/4.8/../../../../x86_64-linux-gn  
u/crti.o /usr/lib/gcc/x86_64-linux-gnu/4.8/crtbegin.o -L/usr/lib/gcc/x86_64-linux-gnu  
/4.8 -L/usr/lib/gcc/x86_64-linux-gnu/4.8/../../../../x86_64-linux-gnu -L/usr/lib/gcc/x86  
_64-linux-gnu/4.8/../../../../lib -L/lib/x86_64-linux-gnu -L/lib/../lib -L/usr/lib/x8  
6_64-linux-gnu -L/usr/lib/../lib -L/usr/lib/gcc/x86_64-linux-gnu/4.8/../../../../lgcc -  
-as-needed -lgcc_s --no-as-needed -lc -lgcc --as-needed -lgcc_s --no-as-needed /usr/l  
ib/gcc/x86_64-linux-gnu/4.8/crtend.o /usr/lib/gcc/x86_64-linux-gnu/4.8/../../../../x86_6  
4-linux-gnu/crtn.o
```

3. 使用 `OBJDUMP` 命令进行反汇编（请自行查阅 `OBJDUMP` 命令的使用方法）。例如，可使用 `objdump -S` 命令进行反汇编
4. 使用 `GDB` 命令进行各种调试（`GDB` 命令参见教材附录C，也可自行查阅网上相关文档）。调试之前首先用 `gcc -g` 命令生成调试信息，否则调试失败。

```
1 gcc -g -o main main.c selectionSort.c add.c printArray.c  
2 gdb main
```

要求用各种 `GDB` 命令对程序进行调试（例如用 `info registers` 查看寄存器内容）。