**《计算机系统基础实验》注意事项**

请大家认真阅读本任务书，在群里下载“实验报告模板”，仔细阅读其中的相关说明。实验前请按照答疑群里的“实验环境说明”准备上机环境。

课内学时上机时段要签到。签到考勤将影响课程综合评分，无故缺勤1/3以上课内上机实验学时者，实践课程成绩为不及格。

课内学时上机前要预习，要做好一定的准备，明确实验要求。

课内学时上机时，主要是记录实验中看到的现象（包括截图），分析讨论各种现象的内在原理，找出错误的原因，以促进对课本知识的理解。实验中应尽可能尝试利用调试工具观察程序执行的过程以及查看内存、堆栈、寄存器的状态。

学生在完成某次实验后应及时按照任务书及实验报告模板的要求，撰写实验报告。实验报告各个章节的目录参照“实验报告模板”。

为满足实验过程指导的要求，本实践课每完成一次任务书的所有实验需要提交一次实验报告的电子版给老师检查。

实验报告电子版文件名格式：

CS2001\_U202012345张三丰.DOC

每个实验的源程序和执行程序的电子版都是需要单独打包的（不要将整个工程文件都打进来了），在提交实验报告时，相关的源程序和执行程序的打包文件同时提交。源程序和执行程序的打包文件名格式：

CS2001\_U202012345张三丰\_1.ZIP

其中,“\_1”表示实验一中涉及到的所有源程序和执行程序；“\_2” 表示实验二中涉及到的所有源程序和执行程序；“\_3”对应实验三中涉及到的所有源程序和执行程序；依次还有“\_4” “\_5”“\_6”。

全部实验（8次上机完成5个任务书的任务）完成之后要提交含有5个实验内容的一个完整实验报告的纸质版，作为成绩评定的主要依据。提交纸质版实验报告的同时，还需要将实验报告电子版（1个WORD文件，总页数不要超过40页，文件大小不要超过5MB）、5个实验的源程序和执行程序的打包文件（5个ZIP文件）一起提交。

实验报告排版格式总体要求见“实验报告模板”中的说明。

另外请注意，计算机系统基础实验更强调实验的过程与思考，而不是把某个功能的程序调通了就完事。一个如实记录了实验过程及进行了深入分析思考的实验报告的成绩，就算程序没有调通，都可能比一个程序没有问题但没有什么过程记录与思考的实验报告成绩高很多。

《计算机系统基础实践》任务书

# 实验一 数据的表示

**一、实验目的与要求**

⑴ 熟练掌握程序开发平台(VS2019/GCC+GDB调试工具) 的基本用法，包括程序的编译、链接和调试；

⑵ 熟悉地址的计算方法、地址的内存转换；

⑶ 熟悉数据的表示形式。

**二、实验内容**

**任务1 数据存放的压缩与解压编程**

定义了 结构 student ，以及结构数组变量old\_s[N], new\_s[N]; (N=5)

struct student {

char name[8];

short age;

float score;

char remark[200]; // 备注信息

};

编写程序，输入N个学生的信息到结构数组old\_s中。将 old\_s[N] 中的所有信息依次紧凑(压缩)存放到一个字符数组message中，然后从 message 解压缩到结构数组 new\_s[N]中。打印压缩前(old\_s)、解压后(new\_s)的结果，以及压缩前、压缩后存放数据的长度。

要求：

1. 输入的第0个人姓名(name)为自己的名字，分数为学号的最后两位；
2. 编写指定接口的函数完成数据压缩

压缩函数有两个： int pack\_student\_bytebybyte(student\* s, int sno, char \*buf);

int pack\_student\_whole(student\* s, int sno, char \*buf);

s为待压缩数组的起始地址； sno 为压缩人数； buf 为压缩存储区的首地址；两个函数的返回均是调用函数压缩后的字节数。pack\_student\_bytebybyte要求一个字节一个字节的向buf中写数据；pack\_student\_whole要求对short、float字段都只能用一条语句整体写入，用strcpy实现串的写入。

1. 使用指定方式调用压缩函数

old\_s数组的前N1（N1=2）个记录压缩调用pack\_student\_bytebybyte 完成；后N2（N2==3）个记录压缩调用pack\_student\_whole，两种压缩函数都只调用1次。

（4） 使用指定的函数完成数据的解压

解压函数的格式：int restore\_student(char \*buf, int len, student\* s);

buf 为压缩区域存储区的首地址；len为buf中存放数据的长度；s为存放解压数据的结构数组的起始地址； 返回解压的人数。解压时不允许使用函数接口之外的信息（即不允许定义其他全局变量）

（5）仿照调试时看到的内存数据，以十六进制的形式，输出message的前20个字节的内容，并与调试时在内存窗口观察到的message的前20个字节比较是否一致。

（6）对于第0个学生的score，根据浮点数的编码规则指出其各部分的编码，并与观察到的内存表示比较，验证是否一致。  
 (7) 指出结构数组中个元素的存放规律，指出字符串数组、short类型的数、float型的数的存放规律。

**任务2 编写位运算程序**

按照要求完成给定的功能，并**自动判断程序**的运行结果是否正确。（从逻辑电路与门、或门、非门等等角度，实现CPU的常见功能。所谓自动判断，即用简单的方式实现指定功能，并判断两个函数的输出是否相同。）

1. int absVal(int x); 返回 x 的绝对值

仅使用 !、 ~、 &、 ^、 |、 +、 <<、 >>， 运算次数不超过 10次

判断函数： int absVal\_standard(int x) { return (x < 0) ? -x : x;}

1. int negate(int x); 不使用负号，实现 -x

判断函数： int netgate\_standard(int x) { return -x;}

1. int bitAnd(int x, int y); 仅使用 ~ 和 |，实现 &

判断函数： int bitAnd\_standard(int x, int y) { return x & y;}

1. int bitOr(int x, int y); 仅使用 ~ 和 &，实现 |
2. int bitXor(int x, int y); 仅使用 ~ 和 &，实现 ^
3. int isTmax(int x); 判断x是否为最大的正整数（7FFFFFFF），

只能使用 !、 ~、 &、 ^、 |、 +

1. int bitCount(int x); 统计x的二进制表示中 1 的个数

只能使用，! ~ & ^ | + << >> ，运算次数不超过 40次

1. int bitMask(int highbit, int lowbit); 产生从lowbit 到 highbit 全为1，其他位为0的数。例如bitMask(5,3) = 0x38 ；要求只使用 ! ~ & ^ | + << >> ；运算次数不超过 16次。
2. int addOK(int x, int y); 当x+y 会产生溢出时返回1，否则返回 0

仅使用 !、 ~、 &、 ^、 |、 +、 <<、 >>， 运算次数不超过 20次

1. int byteSwap(int x, int n, int m); 将x的第n个字节与第m个字节交换，返回交换后的结果。 n、m的取值在 0~3之间。  
   例：byteSwap(0x12345678, 1, 3) = 0x56341278

byteSwap(0xDEADBEEF, 0, 2) = 0xDEEFBEAD

仅使用 !、 ~、 &、 ^、 |、 +、 <<、 >>， 运算次数不超过 25次