



《计算机系统基础》分析应用题

第三章作业

学号	5120203245	班级	卓软 2001	姓名	肖尧
----	------------	----	---------	----	----

1. (1) 长度后缀: w 源: 基址 + 比例变址 + 偏移 目的: 寄存器
 (2) b 源: 寄存器 目的: 基址 + 偏移
 (3) b 源: 基址 目的: 寄存器
 (4) L 源: 立即数 目的: 栈

2. (1) 参数 1, 2, 3 所存放的存储单元地址分别为:
 $8(\%ebp)$, $12(\%ebp)$, $16(\%ebp)$

```

(2) void func(int* xptr, int* yptr, int* zptr){
    int y = *yptr;
    int z = *zptr;
    int x = *xptr;
    *yptr = x;
    *zptr = y;
    *xptr = z;
}

```

3. (1) $R[edx] = x$
 (2) $R[edx] = 4 + (x + y)$
 (3) $R[edx] = x + 8y$
 (4) $R[edx] = 12 + y + 2x$
 (5) $R[edx] = 4x$
 (6) $R[edx] = x + y$

4. (1) 立即数应写作 $\$0xFF$
 (2) $\%ax$ 宽度为 16 位, 应使用 `movw`
 (3) 目的操作数不能为立即数
 (4) 源, 目的操作数类型长度不一致, 应使用 `movwl`



5.

(1) 现代计算机在内存中读写数据是按字节块进行操作，理论上任意类型的变量访问可以从任何地址开始，但是计算机系统会要求这些数据的首地址的值为 K （1 位，4 位或者 8 位）的整数倍，这样可以减少访存次数同时简化硬件设计，使得 cpu 在一次读取即可取得该数据的所有位。编程时注意数据的对齐可以减少为了保持占位而浪费的存储空间。

(2) 当把一个字符串复制到缓冲区中时，如果字符串的长度超过缓冲区的大小，将覆盖 buffer 区域以外的部分内存。而若 buffer 数组之上的区域包含一些关键数据，如返回地址和前帧指针 %ebp。返回地址决定了函数返回时程序将会跳转至何处执行。如果缓冲区溢出修改了返回地址，当函数返回时，它将跳转到一个新的地址。

(3) 每次递归调用都会在栈上开辟新的栈帧空间，相比递推函数，每一次递归调用最少会多使用存放返回地址，%ebp 旧址的 8B 空间，因此空间消耗会很大，递归次数过多甚至有可能爆栈。同时，函数调用会消耗大量资源，如每次开辟新的栈帧都需要保护现场，这些工作都需要时间，因此时间消耗也会很大。