

# PA2 平时作业

- 姓名：管昀玫
- 学号：2013750
- 专业：计算机科学与技术

## 问题一

### 问题

If-else语句举例：

```
int get_cont(int*p1, int *p2 )
{
    if (p1>p2)
        return *p2;
    else
        return *p1;
}
```

```
movl 8(%ebp),%eax    //R[eax]<-M[R[ebp]+8]，即R[eax]=p1
movl 12(%ebp),%edx    //R[edx]=M[R[ebp]+12]，即R[edx]=p2
cmpl %edx,%eax        //比较p1和p2，即根据p1-p2结果置标志
jbe .L1               //若p1<=p2，则转L1处执行
movl (%edx),%eax      //R[eax]=M[R[edx]]，即R[eax]=M[p2]
jmp .L2               //无条件跳转到L2 执行
.L1:
movl (%eax),%eax      //R[eax]=M[R[eax]]，即R[eax]=M[p1]
.L2
```

p1和p2对应实参的存储地址分别为R[ebp]+8、R[ebp]+12，EBP指向当前栈帧底部，结果存放在EAX。

为何这里是“jbe”指令？

### 答

在反汇编后的代码中，jbe .L1 这部分使用的是JBE(Jump If Below) 指令。JBE 指令的作用是根据比较结果(比较寄存器的值是否小于 0) 跳转到指定的地址(.L1 地址)。在例子中，cmpl 指令比较了两个指针 p1 和 p2，将比较结果的标志位设置为相应的值。如果 p1 小于等于 p2，那么 jbe 指令就会跳转到标记为 .L1 的位置继续执行，否则就会执行 movl (%edx), %eax 将 p2 所指向的内容作为结果返回。

jbe 指令是无符号数比较的条件跳转指令，当比较结果为小于等于（也就是无符号数上的“不大于”）时跳转。这里使用无符号数比较是因为指针实际上是一个内存地址，是一个无符号数，所以使用无符号数比较更为合适。

使用 JBE 指令可以根据比较结果快速跳转到需要执行的地址，从而提高代码的效率。

## 问题二

## 问题

nemu输出的helloworld和程序中输出的helloworld有什么区别？

## 答

nemu 输出的 `helloworld` 和程序中输出的 `helloworld` 之间的主要区别在于输出的方式和环境。

在程序中输出 `helloworld`，需要将字符串写入标准输出流（如终端），然后由操作系统将输出流中的内容显示到屏幕上。程序会调用系统调用（如 `write()` 或 `printf()`）将字符串写入标准输出流，这些系统调用最终会调用操作系统内核中的相应函数完成输出操作。

而 `nemu` 输出的 `helloworld` 是在模拟器中输出的，模拟器不会调用操作系统内核函数，而是直接将字符串输出到模拟器的控制台或者日志中。这种方式可以帮助开发者在模拟器中调试程序，查看程序运行时的状态信息和调试信息。但是这种输出方式并不代表程序可以在实际的操作系统中运行并输出 `helloworld`。在实际的操作系统中，程序需要使用系统调用将字符串写入标准输出流才能正确地输出到屏幕上。

除此之外，在实际计算机中，"Hello World" 程序的输出可能会被格式化，例如在输出的前缀或后缀添加一些字符或数字，以表示程序运行的时间或内存使用情况。而在 `Nemu` 中，输出可能不会被格式化，因为它们使用的是模拟器提供的虚拟输出。