

HW8

请参考本节中所讲的数组元素寻址翻译方法,完成以下代码的翻译:

- `A[x, y, z] = p`
- 此处, `A` 是一个数组, 定义为 `A[1..10, 1..10, 1..20] of integer`; 每一个 `integer` 占4字节
- 要求画出语法分析树, 标注综合属性及其计算值, 并给出对应的三地包址码序列

先绘制语法分析树, 参考ppt上即可, 应该不是非常难。注意以下情况:

- 注意是 `A[x, y, z] = p` 而不是 `p = A[x, y, z]` 注意区分。
- 部分同学对 `E -> L` 的规约进行了省略, 合并在了一起 (就是没有L节点, 但是有L节点的属性), 可以不算错。
- 缺少计算值, 就是下面的图的蓝色部分。
- `c = A - 884`, 不是844

参考语法树 (来自王晨晗同学)

ps:

1. 应该用赋值号 `:=`, 而不是等号 `=`;
2. 对于 `s` 处的计算值没必要有 `t5`。


```
short int a = 10;
```

file1.c:

```
#include "head.h"
main() {
}
```

file2.c:

```
#include "head.h"
```

在 x86/Linux 系统上, 使用某版本的 GCC 编译器, 编译命令如下:

```
cc file1.c file2.c
```

编译结果报错的主要信息如下:

```
multiple definition of 'a'
```

试分析为什么会报这样的错误。

Ans

在 `head.h` 中, 定义了 `short int a = 10;`。变量 `a` 在函数外定义, 是全局变量。

在 `file1.c` 与 `file2.c` 中, 均有包含头文件 `head.h`, 这相当于在 `file1.c` 和 `file2.c` 中均定义了全局变量 `a`, 即对 `a` 有多重定义, 因此会报错 `multiple definition of 'a'`。

评分标准: 答出全局变量, 均包含头文件导致重复定义即可。

6.5

一个 C 语言程序如下:

```
typedef struct_a {
short i;
short j;
short k;
} a;
typedef struct_b {
long i;
short k;
} b;
main() {
printf("Size of short, long, a and b = %d, %d, %d, %d\n",
sizeof(short), sizeof(long), sizeof(a), sizeof(b));
}
```

该程序在 Ubuntu 12.04.2 LTS (GNU/Linux 3.2.0-42-generic x86_64) 系统上, 经过编译器 GCC: (Ubuntu/Linaro 4.6.301 ubuntu5) 4.6.3 编译后, 运行结果如下:

```
Size of short, long, a and b = 2, 8, 6, 16
```

已知 `short` 类型和 `long` 类型分别对齐到 2 的倍数和 8 的倍数。试问：为什么类型 `b` 的 `size` 会等于 16？

Ans

一个 `b` 类型的变量仅需 10 个字节就足够。但是，当声明结构体 `b` 的数组时，每个数组元素的第一个成员都是 `long`，它需要按 8 字节对齐。所以，`b` 类型数组的元素首地址应是 8 的倍数。

同时，数组 `size` 的计算原则是：每个数组元素的 `size` 乘以数组元素数量。

为保证符合此原则，同时 `b` 类型数组的元素按 8 字节对齐，则要求：

`b` 的 `size` 为 16

评分标准：答出 `b` 结构体第一个成员 8 字节，因此 `b` 结构体需要按照 8 字节对齐即可。

6.6

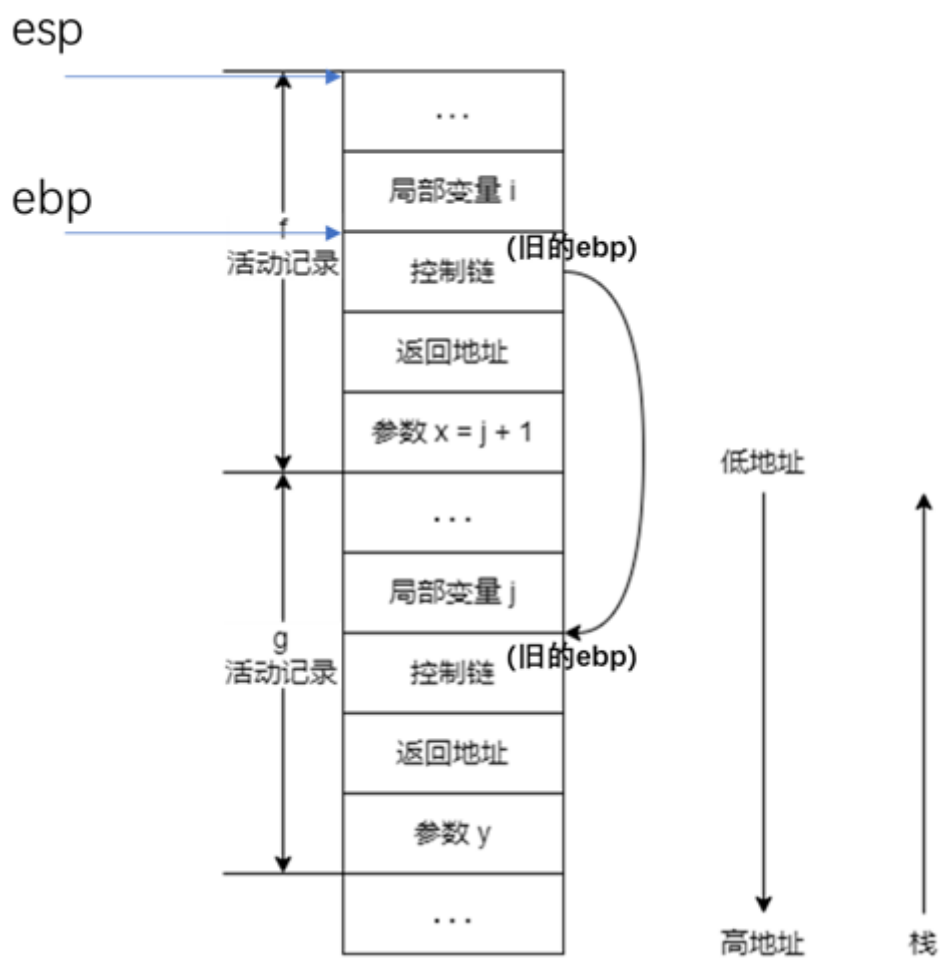
下面是 C 语言两个函数 `f` 和 `g` 的概略（它们不再有其他局部变量）：

```
int f(int x) {int i; ... return i + 1; ...}
int g(int y) {int j; ... f(j + 1); ...}
```

请按照图 6.11 的形式，画出函数 `g` 调用 `f`，`f` 的函数体正在执行时，活动记录栈的内容及相关信息，并按图 6.10 左侧箭头方式画出控制链。假定函数返回值是通过寄存器传递的。

Ans

如图所示



评分标准：题目要求是照着书上画，大体一致即可，但一定需要包含控制链。