

# 课堂作业

## 题1

```
1 (1)
2 T Parent:j=1
3 T Child:i=100,j=1;
4 T Parent:j=2;
5 (2)
6 T Parent:j=1;
7 T Parent:j=2;
8 T Child:i=100,j=1;
9 (3)
10 T Parent:cool;
```

**对于第1种情况，不成立。**

在建立的线程tid中，又定义了变量j，且赋值为0，不能打印出j=1

**对于第二种情况，不成立**

同理，tid线程中j不能为1，T Child无法打印出j=1

**对于第三种情况，成立**

若创建新进程tid后，tid先运行，会改变程序的全局变量buf数据值为cool，T Parent: cool可能打印出现

## 题2

七、设 `tty_insert` 是一个多线程函数体，`tty_buffer` 则是多线程的共享变量，试分析在多个 `tty_insert` 并发的情况下，以下代码可能产生什么安全问题，并给出一种场景。针对此问题，应采用什么方式来避免。（10分）

```
1. struct tty_buffer {
2.     char tty_buffer[MAX];
3.     int used;
4.     int size = MAX;
5. };
6. int tty_insert(struct tty_buffer * tb, char * received)
7. {
8.     int r_size = strlen(received);
9.     if (r_size + tb->used < tb->size) {
10.         memcpy(tb-> tty_buffer + tb->used, received, r_size);
11.         tb->used += r_size;
12.     }
13.     return r_size;
14. }
```

答:

在 `tty_insert()` 中, 首先定义变量 `r_size` 求得函数接受的参数, `char` 数组 `received` 的长度, 比较如果结构体指针 `tty_buffer` 指向的单元中 `char` 数组 `try_buffer` 所用长度和接受的 `received` 长度之和是否超过了定义的 `MAX`, 若不超过则复制 `received` 数组元素到结构体的数组中, 并修改结构体所用数组长度值.

**可能产生的安全问题, 因为 `tty_insert` 是多线程函数体, 可以多个并行执行, 比如同时有4个线程同时接受到输入, 可能单个线程都认为数组没有越界进行复制, 但是可能4个线程接受的输入数组长度之和超过了 `tty_buffer` 的长度限定, 造成数组越界.**

**采取方式: 在多线程同时处理多个输入时, 应该设计线程之间进行通信, 在知道总共的输入和长度后再进行是否可以复制的判定**