

1. 下面的程序采用 Pthreads API。该程序的 LINE C 和 LINE P 的输出分别是什么？

```
#include <pthread.h>
#include <stdio.h>

int value = 0;
void *runner(void *param); /* the thread */

int main(int argc, char *argv[])
{
    pid_t pid;
    pthread_t tid;
    pthread_attr_t attr;

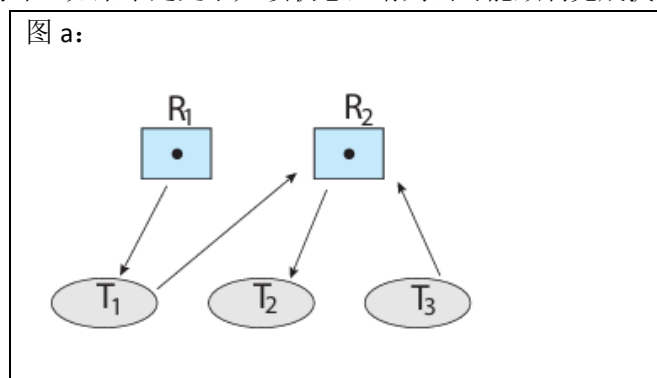
    pid = fork();

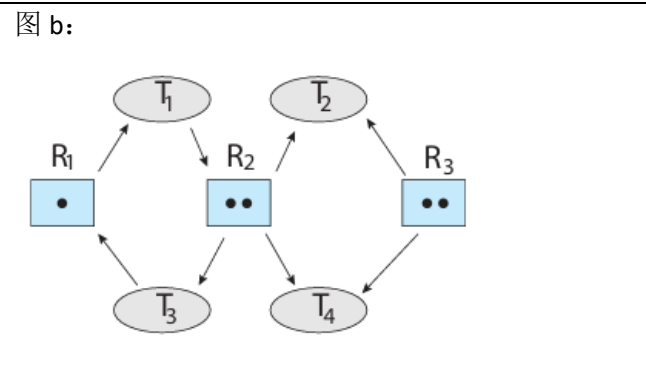
    if (pid == 0) { /* child process */
        pthread_attr_init(&attr);
        pthread_create(&tid,&attr,runner,NULL);
        pthread_join(tid,NULL);
        printf("CHILD: value = %d",value); /* LINE C */
    }
    else if (pid > 0) { /* parent process */
        wait(NULL);
        printf("PARENT: value = %d",value); /* LINE P */
    }
}

void *runner(void *param) {
    value = 5;
    pthread_exit(0);
}
```

2. PPT 中给出的读者-写者问题的解决方案，是否会导致饥饿现象？如果会，请列举一种可能会产生饥饿现象的场景。

3. 下面的两个图中，哪个(些)处于死锁状态？如果处于死锁状态，请列出处于死锁的线程和资源形成的环。如果不是处于死锁状态，请列出可能顺利完成执行的线程顺序。





(图 b 中 $R_2 \rightarrow T_2$ 改成 $T_2 \rightarrow R_2$)

4. PPT 第 36 页，两个问题中选择一个回答。起始状态为完成(1,0,2)的分配之后。