

初始时: $a = 1, b = 1, c = 1$

```
线程1:      线程2:
P(a);      P(c);
P(b);      P(b);
V(b);      V(b);
P(c);      V(c);
V(c);
V(a);
```

- * 12.30 考虑下面这个会死锁的程序。

初始时: $a = 1, b = 1, c = 1$

```
线程1:      线程2:      线程3:
P(a);      P(c);      P(c);
P(b);      P(b);      V(c);
V(b);      V(b);      P(b);
P(c);      V(c);      P(a);
V(c);      P(a);      V(a);
V(a);      V(a);      V(b);
```

- 列出每个线程同时占用的一对互斥锁。
- 如果 $a < b < c$, 那么哪个线程违背了互斥锁加锁顺序规则?
- 对于这些线程, 指出一个新的保证不会发生死锁的加锁顺序。

- ** 12.31 实现标准 I/O 函数 `fgets` 的一个版本, 叫做 `tfgets`, 假如它在 5 秒之内没有从标准输入上接收到一个输入行, 那么就超时, 并返回一个 NULL 指针。你的函数应该实现在一个叫做 `tfgets-proc.c` 的包中, 使用进程、信号和非本地跳转。它不应该使用 Linux 的 `alarm` 函数。使用图 12-47 中的驱动程序测试你的结果。

```
code/conc/tfgets-main.c
1  #include "csapp.h"
2
3  char *tfgets(char *s, int size, FILE *stream);
4
5  int main()
6  {
7      char buf[MAXLINE];
8
9      if (tfgets(buf, MAXLINE, stdin) == NULL)
10         printf("BOOM!\n");
11     else
12         printf("%s", buf);
13
14     exit(0);
15 }
```

code/conc/tfgets-main.c

图 12-47 家庭作业题 12.31~12.33 的驱动程序

- ** 12.32 使用 `select` 函数来实现练习题 12.31 中 `tfgets` 函数的一个版本。你的函数应该在一个叫做 `tfgets-select.c` 的包中实现。用练习题 12.31 中的驱动程序测试你的结果。你可以假定标准输入被赋值为描述符 0。
- ** 12.33 实现练习题 12.31 中 `tfgets` 函数的一个线程化的版本。你的函数应该在一个叫做 `tfgets-thread.c` 的包中实现。用练习题 12.31 中的驱动程序测试你的结果。
- ** 12.34 编写一个 $N \times M$ 矩阵乘法核心函数的并行线程化版本。比较它的性能与顺序的版本的性能。
- ** 12.35 实现一个基于进程的 TINY Web 服务器的并发版本。你的解答应该为每一个新的连接请求创建一个新的子进程。使用一个实际的 Web 浏览器来测试你的解答。
- ** 12.36 实现一个基于 I/O 多路复用的 TINY Web 服务器的并发版本。使用一个实际的 Web 浏览器来测试你的解答。