shlab实现一个tiny shell，有fg、bg、jobs、quit四个内置命令，可以调用其他命令，可以在命令后加&指明在后台执行，可以通过ctrl+c中断前台进程，可以用ctrl+z暂停前台进程。这个lab主要是熟悉进程和信号，看似只有两个概念，但是这两个概念的细节非常多和杂，需要对它们有比较全面和深入的理解。

官方已经给出了整体框架，我们需要完成下面的函数：

eval

builtin\_cmd

do\_bgfg

waitfg

sigchld\_handler

sigint\_handler

sigtstp\_handler

具体的实现可以参看这篇文章，写的比较详细<https://zhuanlan.zhihu.com/p/422490811>。

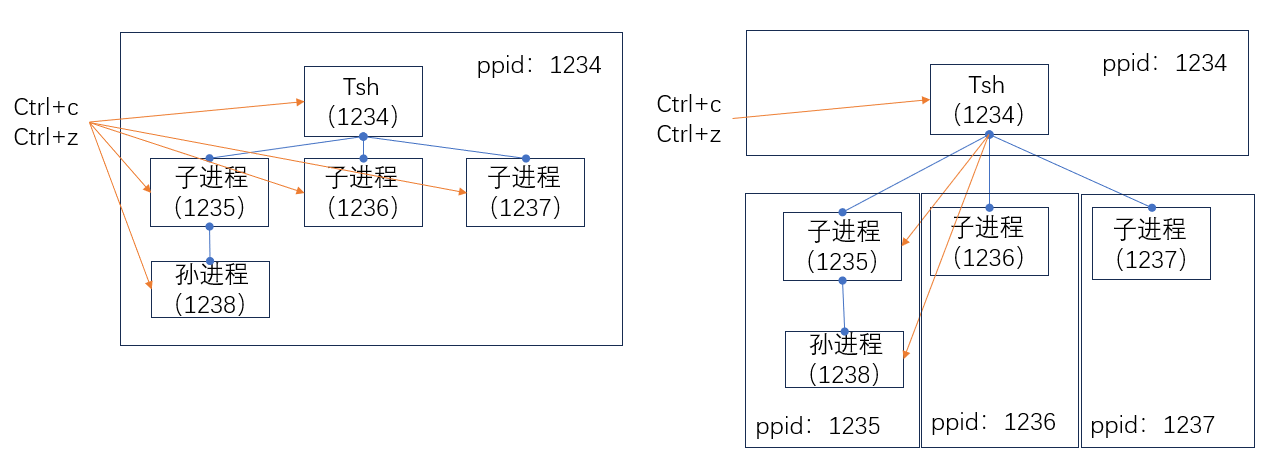
下面说说一些需要注意的点

1. 前台可能是一个进程，也可能是一个进程组，包含多个进程；

比如当我们的tsh运行时，就位于系统shell的前台，当它运行一个程序，fork子进程后，默认它们是同一个组，子进程还可能再fork孙进程，子子孙孙无穷尽，但是默认他们都和tsh位于同一个进程组。这里要区别系统shell的前后台和tsh的前后台，tsh运行后对于系统shell是前台，tsh的前台是它等待结束的命令。

tsh有两个地方跟上述相关，

1. ctrl+c和ctrl+z会同时发送给父进程（tsh）和n个子进程（它们构成系统shell的前台进程组），tsh收到信号会捕获然后处理，所有子进程收到会按照默认方式处理。而我们的要求是中断或者暂停tsh的前台，所以要把tsh的子进程独立成组，让tsh自己成组，键盘信号只发送给tsh，然后它捕获后再发送给它的前台子进程组。
2. kill的第一个参数是-pid，负的pid表示发送信号到进程组为pid的所有进程，如果是pid，则只发给对应的进程，该进程的子进程就会成为僵尸进程。



1. 访问全局共享变量要同步，防止因为并发导致的问题。具体办法就是访问前阻塞信号，访问后解除阻塞。

我搞了个简单的shell脚本，可以一下测试所有的test。

#!/bin/bash

make $101

make $102

make $103

make $104

make $105

make $106

make $107

make $108

make $109

make $110

make $111

make $112

make $113

make $114

make $115

make $116

测试自己的tsh，输出保存到 tshtest.txt中。

./test.sh test > tshtest.txt

测试参考tshref，输出保存到 tshreftest.txt中。

./test.sh rtest > tshreftest.txt

最好不要同时运行上面的命令，因为有些test会运行ps a命令，同时会让输出变得混乱。

然后可以使用文件比较工具输出

github地址：https://github.com/gyxkgz/csapp-lab-solution