**Shell\_Lab**

该实验要求完成几个函数，实现一个linux tiny shell，编程方面没有太多难度，要求是要仔细读以下CSAPP的第八章的内容，同时也需要仔细阅读已有的代码。需要实现的函数及其功能及需要注意的点如下所示：

**• eval: Main routine that parses and interprets the command line. [70 lines]**

书中有eval的代码架构，可以作为参考，需要注意的是要在fork函数之前屏蔽SIGCHLD信号，在addjob函数之后恢复SIGCHLD信号，从而避免addjob和deletejob的竞争导致的问题。

**• builtin\_cmd: Recognizes and interprets the built-in commands: quit, fg, bg, and jobs. [25lines]**

该函数只需要比较字符串来分别指令，并根据指令调用相关函数即可。

**• do\_bgfg: Implements the bg and fg built-in commands. [50 lines]**

该函数实现bg和fg指令，首先确认后面的参数是jid还是pid并找到对应的job，如果是bg指令则给暂停的程序发送继续信号，并修改job状态为BG。如果是FG指令(如果程序暂停)给程序发送继续信号，之后修改job状态为FG，同时调用waifg等待程序执行完毕。

**• waitfg: Waits for a foreground job to complete. [20 lines]**

调用fgpid获取当前前台程序的pid，如果与传入的需要等待的pid相等，则持续循环直至不等。

**• sigchld handler: Catches SIGCHILD signals. [80 lines]**

主要是处理子程序的结束和暂停信号，完成正常和异常结束子进程的删除工作，同时也将暂停子进程的状态修改为暂停。

**• sigint handler: Catches SIGINT (ctrl-c) signals. [15 lines]**

向当前前台执行的进程发送SIGINT信号结束该进程。

**• sigtstp handler: Catches SIGTSTP (ctrl-z) signals. [15 lines]**

如果找到当前前台进程并发送暂停信号。