

Shell Lab 实验报告

前置要求

先大致观察提供函数的作用，免得之后查表麻烦

然后结合ppt和书上内容对要实现的`evalbuiltin_cmd`, `waitfg`函数有大致想法

之后按task来按顺序就行，不确定的或者与标准输出反了的再查阅资料

1.完成`eval()`函数

书上需要补充的已经在实验说明里很清楚了

在`fork()`子进程之前需要屏蔽`SIGCHLD`信号，免得打断父进程`addjob`，然后注意在父子进程结束后各解屏蔽两次就行

比较重要的是将把创建的子进程丢入一个新的进程组

2.完成`bulitin_cmd()`函数

书上补充`listjobs`和`do_bgfg`

3.完成`waitfg()`函数

采用`pause()`来强制等待前台任务完成

4.完成`sigint_handler()`&`sigstp_handler()`函数

向前台任务发送`SIGINT(SIGTSTP)`信号，但是代码中没有向子进程发也过了？
(好像用`-pid`就是解决了，说是发送至整个前台进程组，但是子进程已经被分进别的进程组里了)

5.完成`sigchld_handler()`函数

在完成这个函数之前，部分task测试会出现输出顺序乱掉的情况，完成之后即解决

即需要我们手写回收函数防止出现僵尸进程，采用
`WNOHANG|WUNTRACED`即可

然后利用返回信息来确认是子进程是通过什么原因死亡

如果是`EXITED`，说明是正常返回直接删除；如果是`SIGNALED`，说明是被信号中断；
如果是`STOPPED`，说明停止，修改工作状态

6.完成`do_bgfg()`函数

如果命令行里输入没有，说明命令参数丢失

如果输入里以%打头，说明jid；如果输入是一个数字打头，说明是pid

否则输入参数错误

然后发送信号，修改工作状态，后台就直接输出处理的命令，前台就`waitfg()`

总结

总的来说不难，基本遇到问题网上都能找到答案，大部分工作在课上和ppt里都有展示

注意一些`-pid`细节和`setpgid()`基本就可