# 会议记录一

会议时间：2016/3/21

会议主持人：韩慧敏

与会人员名单：李雨霜，张然殊，张慧昕

会议记录：韩慧敏

会议主题：解决PPT中指出的问题和测试过程中发现的程序bug。

自由发言

1. 发言人：李雨霜，议题：kill()函数

问题：

父进程执行完kill(pid, SIGUSR1)命令后子进程被杀死，要保证子进程在父进程Kill函数执行前运行。

解决方法：

要使父进程等待子进程的运行，在在execOuterCmd函数的父进程的kill函数前加上sleep(1)。使父进程在子进程结束之后才能执行kill函数。

1. 发言人：张慧昕，议题：wait()函数

问题：

运行前台作业，执行ctrl+z命令挂起：正常

 执行fg %<int>命令继续运行前台作业：错误

（命令提示符在作业运行完之前就打印出来，即父进程没有等待子进程执行完）

解决方法：

wait函数要等待子进程收到开始启动进程的信号后运行才能得到正确的结果，需要在fg\_exec函数中的kill函数后加上sleep(1).

1. 发言人：韩慧敏，议题：Ctrl\_Z命令

问题：

运行后台作业过程中，键入ctrl+z会导致后台作业停止运行。

解决方案：

在execOuterCmd函数的子进程调用execv函数之前加上setpgid(0,pgid)，为子进程设定单独的进程组。

解决过程中出现问题：

调用了setpgid函数之后，子进程就不在前台进程组了，失去了对终端的控制权，无法执行需要输入的命令。

解决方案：

在子进程和父进程中都为子进程调用setpgid设立单独的进程组，然后在父进程的waitpid函数之后，调用tcsetpgrp函数将终端控制权转移给子进程，因为此时的终端控制权在父进程，所以该函数写在父进程中。

解决过程中再次出现问题：

父进程把终端控制权交给了子进程，当子进程运行结束时，终端控制权无主，程序出现异常。

解决方案：

在子进程的末尾，也就是子进程运行结束之后，再次调用tcsetpgrp函数将终端控制权转交父进程。

1. 发言人：张然殊，议题：新测试出的bug（rmjob函数）

问题：当后台有程序运行时，输入一个无关指令，例如格式不正确或无法处理的fg，bg指令，会导致程序运行结束后不能从jobs作业链表中移出。

解决方法：观察代码发现，无关指令置ingnore为1，使进程不能从jobs作业链表中移出。则在fg\_exec,bg\_exec函数判断指令是否有效的时候，如果指令无效，则将ingnore置为0。bg\_exec函数中，若操作不是挂起状态的进程，也视为无关指令，将ingnore置为0.

解决过程中遇到问题：测试后发现，用fg命令将./Demo&命令进行后台运行的进程提至前台运行，程序运行结束后也不会从jobs作业链表中删除。

解决方法：问题原因是rmjobs函数需要子进程结束时返回一个信号才能删去进程，所以在fg\_exec函数里加上判断：

for(i=1,now = head;now != NULL;now = now->next,i++){

if(now -> pid == pid){

if(strcmp(now -> state , RUNNING) == 0)

ingnore = 0;

else

ingnore = 1;

}

}

会议总结：

本次会议以助教给出的三个“坑”为突破点，从这里入手，先找到这些坑的解决办法，初步了解程序整体的运行方法，自学一部分关于信号，设立进程组的知识，初步解决了程序中明显的bug。

分配任务：

张慧昕：管道的实现。

李雨霜：yacc语法分析器的修改，lex词法分析器。

韩慧敏：Ctcrl\_C外部命令的实现和内部命令的实现。

张然殊：代码回归性测试及bug修复。

会议结束

会议记录提交人：韩慧敏

会议记录审阅人：张然殊