会议流程

2016/3/22

主讲：李搏

旁听：林子义、黄秋宇

书记官：胡亚龙

校验：林子义

审核：李搏

第一部分：进程的运行。

林：为什么要fork，不直接exec。为什么不把fork和exec做成一个函数？

问题：是否 fork后面都是exec？

execute函数讲解：

execute->struct->{fg(inner Instr),bg(inner Instr),exit,history}->fork->（多线程）

1、重定向->execv->signusr1,goon

2、addjob(pid) 得到pid，和shell不是一个进程组，收到不同的信号。

->sigchld(CHLD 处理僵尸进程 如果子进程退出，父进程没wait，子进程就会变成僵尸进程。)->CHLD。

->rNjob->回到shell。

林：前台运行->sigsuspend

李：wait 有信号就会产生变化，不会区分是否是开始还是结束。

黄：shell 自己是一个进程组，每个命令是一个进程组，管道是一个自己的进程组（自己管理多个进程，其余的由操作系统自己管理多个进程）。

譬如管道为了方便管理挂起和结束，就需要一个进程组所有进程同时收到信号。

ppid:父进程标志。

前台收到ctrl\_z ,ctrl\_v，其余进程屏蔽。收到进程会进程组会采取一样的操作。

rNjob:会发送多次SIGCHLD会重复删除，删除链表，根据PID找不到目标就不会删除。

Sigtton 和sigttin ：不处理，如果后台运行完毕之后需要用到前台，shell采取的方法不响应。

除了管道的进程组之外，其余的进程组由操作系统自己管理。

SIGNCHLD:信号不稳定，对于管道不行，因为有多个进程，检测有没有，不检测进程的个数。多重中断。中断之后回到原状态。

Begin:

Fork():为了保证子进程先运行goon()死循环，要保证执行的顺序是addJob() startJob endJob rmJob。

父进程（等待） 子进程发送sigusr：

子进程等待 父进程addJob(),发送sigusr：

子进程开始执行execv:

执行完毕发送sigchld，rmjob。

END

管道：

begin:

管道不允许后台。

管道入口和普通命令入口不一样。execute2。

将指令分割：通过begin end，产生不同的指令的标号。

创建管道是dup2(pd[0],0),将pd[0]复制给0（标准输入），1（标准输出）。

pipe(pd(int[2])),输出目标是在数组的1，输入方是在数组的0.

管道执行过程：先创建一个进程组，为每个进程创建一个pid，放在一个管道的进程组里，重定向在执行外面的进程函数里面，pipeflg是标志管道里的进程属性，是否需要输入和输出。

重定向在执行之前。

execv:（页表,SIGNAL（保持））、（IO,PID（更改））.判断是否存在，参数是否合法。

保证管道全部运行完毕才开始执行其余的。

killpg.来发送信号。fg %pgid。

管道的清理都是单独的有别于别的进程，单独设计。

End

more:

more 先将内容读到内存，读到-1，然后按照格式响应。暂时不支持后台运行。