会议记录

参与人员：黄智勇（14061190）、王贺（14061184）、兰帅（14061203）

第一次：（2016/03/20）

首次会议讨论了本次试验的具体内容，明确了实验的基本要求以及提高要求，初步针对实验内容进行了分工：

黄智勇为本次实验的主要负责人，负责总体设计实验方案，添加管道及Yacc/Lex语法分析部分；

王贺负责整体测试程序，并负责修复发现的BUG，同时对实验报告内容进行修正；

兰帅负责调试内部命令、实验报告的书写以及会议内容记录。

第二次：（2016/03/23）

细化了部分分工，相互之间讨论了一些不太理解的概念，加深了对实验内容的理解。

黄：I/O重定向怎么理解，为什么说是“重定向”？

兰：对Linux中的进程来说，每个打开的文件都是通过文件描述符(FD)来标识的，在进程创建的时候，内核为进程默认创建了0、1、2三个特殊的FD，所谓的I/O重定向也就是让已创建的FD指向其他文件。

黄：那它具体是怎么实现的呢？

王：可以使用复制文件描述符系统调用dup2()来实现重定向。首先判断用户输入的指令是不是标准输入，如果不是，则说明用户使用了重定向命令，则先将标准输入关闭，再用dup2()复制文件描述符给标准输入或者标准输出（或者标准错误输出），然后将输出重定向到指定文件中。

兰：对于fork()函数和exec()函数，你们是怎么理解的？

王：fork()可以函数启动一个新的进程，这个进程几乎是当前进程的一个拷贝；函数fork()创建一个新的子进程，其子进程会复制父进程的数据与堆栈空间并继承父进程的用户id、组id、环境变量，以及打开的文件、工作目录和资源限制等，需要注意的是这些继承来的信息是复制而来的，并非指向相同的内存空间。

exec()函数族允许当前进程启动另外一个程序，但是一个进程一旦调用exec类函数，它本身就"死亡"了。一个程序如果想启动另一程序的同时自己依旧能够继续运行的话，就需要将fork与exec结合起来使用。

兰：shell命令解释器是怎么执行的呢？有没有什么先后顺序？

shell命令解释器在执行命令时，先查找内部命令，若要执行的命令不是内部命令，再调用外部命令。

王：怎样能够实现让命令在后台执行？

黄：若在命令末尾加入符号&，那么这条命令将后台执行。具体来说，首先将屏幕标准输入关闭防止后台命令获取输入，然后就是改变消息处理方式，使父进程不用等待子进程执行完毕，从而可以在子进程运行的同时执行下一条命令。

兰：相比于内部命令，外部命令有哪些不同的特征呢？

黄：通常外部命令的实体并不包含在shell中，这些可执行的外部命令保存在一个独立的文件当中，但是其命令执行过程是由shell程序控制的，当一个外部命令被执行时，一个新的进程即被创建同时命令被执行。shell程序管理外部命令执行的路径查找、加载存放，并控制命令的执行，外部命令是通过调用execv来实现的，如vi

第三次：（2016/04/04）

兰：shell本身是一个进程组，管道是一个自己的进程组。

王：wait()一有信号就会产生变化，不会区分子进程是否是运行结束或者退出，或者是其它的状态变化。