MyShell 使用手册

一．shell 的概念

Linux 下的 Shell 是用户使用 Linux 的桥梁，是用户和 Linux 内核之间的接口程序。Shell 应用程序为用户提供了一个界面，用户通过这个界面访问操作系统内核的服务。每一个 Shell 程序也拥有自己的一些内建命令和内部变量。

二．Myshell 执行命令

在 myshell 中有三种执行命令

1. 内置命令(Builtin)

Myshell 在执行内置命令的时候不会派生新的进程，而是由 shell 程序直接执行。查看这些命令的帮助信息需要用 help command 查看。

2. 外部命令

myshell 中的外部命令的执行是调用一些可执行的二进制文件。myshell 在执行他们的时候会创建出子进程，通过子进程调用 exec 函数来执行外部命令。这时子进程的环境就会被相应外部命令的环境所取代。

3. 执行 shell 脚本

在执行脚本程序时，myshell 会先判断命令是 myshell 内部命令还是外部命令。如果是内部命令则由 myshell 直接调用执行，若是外部命令则创建新的进程执行。

三．Myshell 基本功能

1. 内建命令

命令	指令格式	说明
bg	bg [job_spec ...]	将进程调到后台执行
cd	cd [dir]	改变当前的工作目录
clr	clr	清屏操作
echo	echo [arg ...]	将变量或者字符打印到标准输出
dir	dir [dir]	列出指定目录下的所有文件
environ	environ	列出所有的环境变量
exit	exit	正常退出 myshell

help	help [inst]	显示内建命令的帮助列表
jobs	jobs	显示出当前正在运行的进程信息
myshell	myshell [filename]	解释 shell 脚本文件
pwd	pwd	显示当前的工作目录
set	set [-$] [command or arg ...]	设置位置参数
shift	shift [n]	移动位置参数
time	time	显示出系统当前的日期和时间
unset	unset [name]	取消变量的设定

其中：
set 命令允许通过指令设置位置参数，例如 set $(ls | grep temp)。
set 和 echo 的输入中也能够解析变量和引号符号。
例如：
set “$HOME” “hello world”
echo “$HOME” “hello world”

2.外部命令

myshell 如果在内建命令列表中无法找到相应指令，这通过调用子进程调用外部命令来执行相应指令。

3.脚本解释

myshell 可以通过读入文件提取命令。

可以使用以下命令进行脚本读入：myshell filename

myshell 命令可以通过读入文件处理一组命令集。读入文件时，myshell 会对文件内容进行逐行读入并解释，当到达文件尾时，myshell 会退出。当使用 myshell 命令时没有使用参数，这时 myshell 会在屏幕上显示提示符请求用户输入相应参数。

4.后台执行程序

在命令后加上空格和&，便可以让命令在后台执行。

myshell 通过创建子进程来实现程序的后台执行。主进程则直接跳转等待下一次命令输入，而不用等待子进程的操作结束。子进程在执行完毕相应的命令后也会自动结束进程。

四．变量操作

1.系统环境变量

Linux 系统中的环境变量在操作系统启动的时候被初始化，与系统所使用的 Shell 版本无关。常用的环境变量有 PATH、HOME、USER、PWD 等。在 myshell 中，设计为环境变量可以从系统中读取。

2.本地变量

Shell 内部设置的变量。各个 shell 版本的本地变量各自独立，不能被其他 shell 访问或者修改。

在 myshell 中用户用等号设置本地变量。等号左边是变量名，必须以字母开头。等号右边是变量的值。变量赋值支持指令赋值，即将指令输出的结果赋值给变量。例如：a=$(ls | grep XXX) ，变量 a 的值就是执行 ls | grep XXX 的执行结果。也可以直接将值赋值给变量，同时支持引号和变量解析。例如：a="This is $SESSION"，则变量 a 的结果即为将$HOME 替换以后的字符串，即 "This is ubuntu"。指令中也可以采用变量，例如 cd $HOME 就是转到用户目录下。

unset 指令可以取消本地变量的赋值，即从变量列表中删除相应变量。

3.位置参数

位置参数是 shell 中一类特殊的本地变量，可以用于命令行向 shell 脚本中传递参数使用。$1 代表第 1 个参数，$2 代表第二个参数…

在 myshell 中通过 set 命令可以给位置参数赋值。在设置位置参数时 set 同样支持指令解析，即将指令输出的结果用于参数的值。例如：set $(ls | grep xxx)即将指令 ls | grep xxx 的结果赋值给位置参数。也可以直接输入赋值，set 同样支持变量和引号操作，例如 set $SESSION "hello world" ，其中$1 等于 ubuntu，$2 等于 hello world。

其中还有一些特殊的位置参数。$@表示位置参数的所有的值，$#表示位置参数的个数。

shift 操作可以实现位置参数的移动。例如 shift n：将位置参数向前移动 n 位，即第 n+1 位移动到第 1 位，n+2 位到第 2 位… 前 n 位被丢掉等…

五．输入与输出重定向

LINUX 中默认的标准输入(stdin=0)和输出(stdout=1)对应的都是终端的屏幕。而输入输出重定向，就是将一个文件、命令、程序、脚本的输出或输入重新定向到另一个文件，命令，程序，脚本。实质上标准输入输出各自也都是一个文件(在系统启动时就已经打开，其文件描述符分别为 0 和 1)。

在 myshell 中通过>来实现输出重定向，即将表准输入输出到目标文件中而不是屏幕端。>右边的文件可以存在也可以不存在。如果文件不存在，就新建文件。然后将输出结果覆盖文件内容。符号”>>”表示将将输出结果添加在文件末尾。

myshell 中通过<来实现输入重定向，右边跟目标文件。如目标文件不存在，则报错。意思是指令的输入来自于目标文件内容而不是屏幕终端的输入。

六．管道操作

在 linux 中可以使用管道符（”|”）将一个命令的标准输出被导入或者重定向为下一个命令的标准输出。例如 command1 | command2 意思是将 commamd1 的输出作为 command2 的输入进行操作。

myshell 中同样支持管道操作。例如：ls | grep temp > test.txt 意思为将 ls 的输出作为 grep 操作的输入，并将结果重定向到 test.txt 文件中。