Myshell 使用手册

计算机科学与技术 1401 班 庞博

一. 什么是 Shell

Shell 本质上是一个命令语言解释器, 是一个用户和系统之间的接口程序. 用户输入的每一个命令都由 shell 先解释然后再传给 Linux 内核执行。 每一个 Shell 也拥有自己的一些内建命令和内部变量。

二. Myshell 的启动和指令执行

每个程序启动时会先进行一些工作，以使得后面的工作能更好地执行。Myshell 启动时首先会自动对环境变量初始化，读取系统的环境变量，并写入 shell 的部分内部变量。其次对系统环境变量 PATH 中的所有目录检索，把所有文件名添加到字典树中，以便命令自动补全和错误提示使用。最后，初始化位置参数，创建工作进程栈。

程序初始化完毕之后，终端上首先出现命令提示符，等到光标开始闪烁，用户就可以开始输入命令，输入的命令在屏幕上立即显示。值得注意的是，在 myshell 的命令中，每个指令名，程序名和符号等都要用且仅用一个空格隔开，否则可能无法正确地解释用户输入的命令。

用户输入一串命令后，Myshell 首先检查命令是否是内部命令，不是的话再检查是否是一个应用程序或可执行文件, 这里的应用程序可以是 Linux 本身的实用程序，比如 ls 和 rm，也可以其他用户安装的程序，比如 codeblocks。然后 Myshell 试着在搜索路径里寻找这些应用程序。搜索路径是一个能找到可执行程序的目录列表(PATH)。如果你键入的命令不是一个内部命令并且在路径里没有找到这个可执行文件，将会显示一条错误信息。而如果命令被成功的找到的话，Myshell 的内部命令或应用程序将被分解为系统调用, 并传给 Linux 内核。

三. Myshell 基本功能

1. 内建命令

cd: 改变当前工作目录
clr: 清屏
dir: 显示一个目录下的所有文件
environ: 显示环境变量
echo: 打印语句或变量值
help: 显示帮助
quit: 退出程序
jobs: 显示当前的作业清单
myshell: 解释脚本文件
set: 显示变量或设置位置参数
export: 设置环境变量
shift: 移动位置参数
unset: 取消变量的设定
history: 显示当前已保存的历史输入

2. 命令历史记录

用户从启动 myshell 之后，每个输入过的命令都会保存在 myshell 的历史记录中。使用历史记录列表最简单的方法是用上方向键。按下上方向键后最后键入的命令将出现在命令行上。再按一下则倒数第二条命令会出现，以此类推。如果上翻多了的话也可以用向下的方向键来下翻。这些历史命令同样可以被正常编辑。此外，输入 history 命令可以一次输出已保存的所有历史记录

3. 命令补全

用户输入命令时若按下 tab 键，myshell 会自动根据用户已输入的字符串，先在命令集中搜索，若有以用户已输入字符串开头的命令，则自动将其补全(第一个搜索到的命令)；若命令集中没有以该字符串开头的命令，则 myshell 会另起一行列出环境变量 PATH 中所列的所有目录中以该字符串开头的文件名和目录名。

4. 错误提示

当用户输入完一个命令给 myshell 进行解释，myshell 检测到该命令无法找到，且不存在以该命令为名称的文件时，myshell 会搜索与该命令相近(仅相差一个字母)的命令并列出该命令及其所在目录。

5. 脚本解释

Myshell 能够从文件中提取命令行输入，例如 myshell 使用以下命令行被调用：
myshell batchfile：这个批处理文件应该包含一组命令集，则 myshell 会逐行读取该文件的内容并进行解释。当到达文件结尾时 shell 退出。

如果 myshell 被调用时没有使用参数，它会在屏幕上显示提示符请求用户输入。

6. 工作进程管理/后台执行

在一个命令后空格加上&，可以让该命令创建一个子进程在后台执行，主线程直接跳转到等待下一次输入，而不必等待上一个命令执行完毕。子线程执行完毕后也自动结束该进程。用 jobs 命令可以显示当前正在后台执行的子进程。

7. 路径解释：路径分为绝对路径和相对路径。当只输入了目标文件的文件名时，默认在当前工作目录下查找该文件，无法找到则判定该文件不存在；当文件名的输入中以/或.或~开头时，myshell 会判定其为绝对路径，其中.会被替换为当前工作目录，~会被替换成主目录，这些处理完成后所得字符串即为完整的绝对路径，直接转到该路径读取文件.

四. 变量相关

1. 系统环境变量：指的是当前用户的 LINUX 系统中的环境变量，在操作系统启动时被初始化，与所使用的 shell 无关，包括 PATH，HOST，USER,CWD 等。但在 myshell 中可以从系统读取，也可以通过 export 命令来设置新的系统环境变量。

2. 本地变量：指 myshell 内设定的变量，各个 shell 的本地变量各自独立，不能被其他 shell 访问或修改，包括 PS1，PS2 以及位置参数等。用户可以直接用形如 变量名 = 变量值(等号两边有空格)来设置新的本地变量或修改已有的变量,其中变量名不能有空格，变量的值可以用多个空格隔开。也可以用 unset 变量名 来取消变量的设定(不

影响系统环境变量)，当关闭 myshell 的进程时这些本地变量也会自动丢弃。

3. 位置参数：位置参数也是一种特殊的本地变量，用于从命令行向 shell 脚本传递参数，$1 表示第 1 个参数，$2 表示第二个参数，以此类推，$0 为脚本的名字。输入 set `命令` 可以将命令的输出（以空格为分界）依次作为位置参数。若位置参数超过 9 个，则第 9 个之后的位置参数无法直接用 $数字 访问，需要用 shift 命令进行前移。

   此外，还有一些特殊的位置参数，如$#表示传递到脚本的参数的数量；$@和$*表示传递到脚本的所有参数；$?表示命令执行的返回值，0 表示没有错误，非 0 表示有错误。

4. 变量取用：可以用 echo 变量名 命令输出变量的值 (若是位置参数则变量名为$+数字或符号)

五. 输入输出重定向

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)

1. 输出重定向 > : 输出重定向符号的左端是一条指令，若没有重定向符号，则其执行结果的输出将直接输出到屏幕上。而加上了 > 重定向符号之后(右边必须是一个文件名)，该指令的执行结果输出将输出到后面的文件中。若后面的文件名在当前目录下不存在，则创建该文件；若已存在，则输出将会覆盖该文件的原内容。若不希望输出覆盖到该文件，可以把 > 改成 >>,这样输出就会在已存在的该文件末端开始写入输出结果（若文件不存在依然是创建一个新文件）。
2. 输入重定向 < : 输入重定向符号的左端是同样是指令，右边同样是文件名. 当不使用<符号时，指令执行过程中的输入流将会从终端的屏幕读取。而加上 < 重定向符号之后，就会先从 < 右边的文件中读取其内容，作为 < 左边的指令执行的标准输入(若右边的文件无法找到则会报错)

myshell 常用环境变量：

变量名	变量值
HOST	主机名
USER	用户名
PATH	执行文件搜索路径
PWD	当前工作目录
HOME	主目录
PS1	一级提示符变量
PS2	二级提示符变量
SHELL	当前 SHELL 所在目录

其他环境变量可以在 myshell 中使用 environ 命令查看