山东大学 软件 学院

Linux应用 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202000300125 | 姓名：贾星宇 | | 班级：2020级5班 |
| 实验题目： shell的使用 | | | |
| 实验学时：8 | | 实验日期：2020年11月27日 | |
| 实验目的：  1. 掌握shell通配符的含义和使用  包括：\* ？[…] [!...]  2. 掌握重定向的使用  2. 掌握管道的使用  3. 掌握输入和输出  4. 掌握tee的使用  5. 掌握正则表达  6. 熟悉搜寻和排序  7.掌握Shell程序的三种运行方式. | | | |
| 硬件环境：  lntel(R) Core(TM) i5-10210U CPU @ 1.60G Hz 2 .1 1 G Hz | | | |
| 软件环境：  VMware Workstation Pro ubuntu Linux | | | |
| 实验过程报告：  (一) shell环境  1. 用date命令查看系统当前时间，在输入da后，按tab键，让shell自动补齐命令的后半部分。    2. 用mkdir命令创建新的目录newdir。首先输入第一个字母m，然后按tab键，由于以m开头的命令太多，shell会提示是否显示全部的可能命令，输入n。    3. 再多输入一个字母k，按tab键，让shell列出以mk开头的所有命令的列表。  4. 在列表中查找mkdir命令，看看还需要多输入几个字母才能确定mkdir这个命令，然后输入需要的字母，再按tab键，让shell补齐剩下的命令。    5. 最后输入要创建的目录名newdir，按回车键执行命令。    6. 用ls命令查看newdir目录下的内容，看newdir是否可以用文件补全功能来输入？不能。    7.输入alias命令，显示目前已经设置好的命令的别名。    8.设置别名ls为ls –l,以长格形式显示文件列表：＃alias ls=‘ls -l’。  9. 显示别名ls代表的命令，确认设置生效：＃alias ls。    10. 使用别名ls显示当前目录中的文件列表。    11. 在使定义的别名不失效的情况下，使用系统的ls命令显示当前目录中的命令列表：＃\ls。    12. 删除别名ls：＃unalias ls。  13. 显示别名ls，确认删除别名已经生效：＃alias ls。    14. 最后再用命令ls 显示当前目录中的文件列表。    l5. 写出定义别名cde为：cd /etc的命令（ ）。    (二)通配符的使用  1. 以tux1帐号登入系统，用ls /bin/\*sh 命令显示系统中的shell种类？系统中共有哪些shell？ls /bin/\*sh命令的作用是什么？    2. 进入/etc目录并且列出该目录下的所有文件。    3. 使用带通配符的ls命令列出文件名：  (1) 文件名以conf结尾的文件    (2) 文件名以一个字母d或D开头的文件    (3) 文件名的第五位包含一个字母o的文件    (4) 文件名包含文字tab且不区分大小写字母的文件    (5) 文件名以一位数字结尾的文件    (6) 文件名不以一位数字结尾的文件    注意：通配符的匹配是由shell来做的。如果匹配的文件名是一个目录名，ls命令默认将列出那个目录的内容,而不是文件名本身。  可以使用-d参数来阻止该情况的发生。  4. 执行命令ls -d ?[!y]\*[e-f]将会发生什么情况？匹配的最短的文件名会是什么？执行该命令来验证你的答案。  匹配不是y开头的且以e或f结尾的文件。    5. 返回到你的主目录。    (三） 重定向  1. (1)使用cat命令重定向创建一个名为junk的文件，该文件包含了一些文本行。键入一些行以后，中止cat命令返回到shell提示符。  查看你刚才创建的文件的内容。    (2)使用重定向在junk文件中再添加一些行。然后查看该文件的内容，检查你所输入的内容是否完整。      2. 对于命令  $cat junk junk2  (1)将其标准输出重定向到文件cat.out,标准错误重定向到cat.err。  (2)如果显示输出信息、但不保留该命令的错误信息呢？    （四） 管道  3. 使用管道统计你的当前目录里的文件数，不要手动去计数。  4. 执行ls -l> tempfile ; wc -l tempfile ; rm tempfile 命令与第7题作用一样吗？为什么一样或不一样？  5. 使用ls命令并把输出结果保存在名为tempfile2的文件中，然后统计并显示文件数。    （五） 命令组  1. 在同一个命令行上，显示当前的系统日期和登入的所有用户，包括一些说明注释，结果存入到一个文件中。    （六） 进程环境  2. 显示在你的当前进程环境中定义的所有变量。显示所有环境变量。    3. 用命令 $localvar=10 给本地变量localvar赋值。显示该变量的值，然后再显示当前所有的变量。    4. 用 $bash 命令启动一个子shell。再显示变量localvar的值。这次和28中显示的为什么不同？    无法显示localvar  5. 用什么命令可以将localvar的赋值带到子shell中去？    （七）使用grep和正则表达式  1. 复制/etc/passwd文件到你的主目录，并换名为lspasswd。    2. (1)搜寻在lspasswd文件中所有首字母是s的行。    （2）用wc,grep和管道组合成一条命令统计lspasswd文件中首字母是s的行数。    3. 重复上面的操作，但这次只显示符合条件的行的计数。  4. 尝试用grep 和正则表达式在lspasswd中查找其它你想要查找的信息。其中：  ^ 用于匹配行首，用在字符串前面  $ 用于匹配行尾，用在字符串后面  X\* 0个或多个字符X  . 匹配任意单字符  [字符表] 字符表中的任意字符  [^字符表] 任意不在字符表中的字符  \ 屏蔽其后的特殊字符的特殊含义  \{n\} 前导的正则表达式重复n次  \{min,max\} 前导的正则表达式重复min~max次    （八）Shell程序及其三种运行方式  1.使用vi编辑器,编辑一个名为setdir的脚本文件:  dir1=/tmp  dir2=/usr  echo $dir1 $dir2  2.分别用学过的三种不同的运行方式运行该脚本,每次运行一种方式后,再次执行echo $dir1 $dir2    3.这三种运行方式有什么不同?执行echo $dir1 $dir2分别得到什么样的运行结果?为什么? | | | |
| 结论分析与体会：  1.  Shell的常用通配符：  在这里插入图片描述  “[-]”匹配中括号中任意一个字符，-代表一个范围。例如：[a-z]代表匹配一个小写字母。  “[^]”逻辑非，代表匹配不是中括号内的一个字符。例如：[^0-9]代表匹配一个不是数字的字符  {‘’}单引号。在单引号中所有的特殊符号，如“$”和“`”(反引号)都没有特殊含义  Eg：aa=123  Echo ‘$aa’ #输出$aa  {“”}双引号。在双引号中特殊符号都没有特殊含义，但是“$”、“`”、“\”是例外，拥有“调用变量的值”“引用命令”“转移符”的特殊含义  Eg:echo “$aa” #输出结果为123  {``}反引号。反引号括起来的内容是系统命令，在bash中会先执行他。和$()作用一样，不过推荐使用$()，因为反引号非常容易看错。  Eg：aa=’ ls’  Echo “$aa”  {$()}和反引号的作用一样，用来引用系统命令  {#} 在shell脚本中，#开头的行是代表注释  {$}用于调用变量的值，如需要调用变量name的值时，需要用$name的方式得到变量的值  {\}转移符，跟在\之后的特殊符号将失去特殊含义，变为普通字符。如\$将输出“$”符号，而不当做事变量引用。  2.重定向的使用  重定向是一种高效的方法，无需大量的鼠标与键盘操作就可以完成数据的转移。重定向可以分为输入重定向以及输出重定向这两种类型。由于所有程序都有输入或者输出，因此输入和输出的重定向是任何编程语言或脚本语言都自带的功能。输入信息会从 stdin 中读取（标准输入，通常是键盘或鼠标）。  输出信息会被输出到 stdout （标准输出，一个文本文件或者数据流）。  错误信息会被输出到 stderr。  3. 管道的使用  管道是一种两个进程间进行单向通信的机制。因为管道传递数据的单向性，管道又称为半双工管道。管道的这一特点决定了器使用的局限性。管道是Linux支持的最初Unix IPC形式之一，具有以下特点：  \*\*\* 数据只能由一个进程流向另一个进程（其中一个读管道，一个写管道）；如果要进行双工通信，需要建 立两个管道。  \*\*\* 管道只能用于父子进程或者兄弟进程间通信。，也就是说管道只能用于具有亲缘关系的进程间通信。  注意：从管道读数据是一次性操作，数据一旦被读，它就从管道中被抛弃，释放空间以便写更多的数据。  管道命令：  command1 | command2 | command3  操作符是：”|”,它只能处理经由前面一个指令传出的正确输出信息，对错误信息信息没有直接处理能力。然后，传递给下一个命令，作为标准的输入.  4.输入和输出 - 标准输入 默认是来自键盘的输入 stdin 0  - 标准输出 默认输出到终端窗口 stdout 1  - 标准错误输出 默认输出到终端窗口 stderr 2  5.Linux tee命令用于读取标准输入的数据，并将其内容输出成文件。  tee指令会从标准输入设备读取数据，将其内容输出到标准输出设备，同时保存成文件。  语法：  tee [-ai][--help][--version][文件...]  参数：  -a或--append 　附加到既有文件的后面，而非覆盖它．  -i或--ignore-interrupts 　忽略中断信号。  --help 　在线帮助。  --version 　显示版本信息。  6. 正则表达式通常被用来检索、替换那些符合某个模式(规则)的文本。目前多种程序开发语言均支持常规的正则表达式。作为一名系统、运维工程师，正则表达在日常工作中也比较常用。因此，掌握常用的正则表达式显得十分重要。  在目前的Linux系统中，通常搭配三剑客（grep、sed、awk）来使用。目前，正则表达式主要分为三类：基本正则表达式（Basic Regular Expression 又叫Basic RegEx 简称BREs）、扩展正则表达式（Extended Regular Expression 又叫Extended RegEx 简称EREs）、Perl的正则表达式（Perl Regular Expression 又叫Perl RegEx 简称PREs）。基本语法  一个正则表达式通常被称为一个模式（pattern），为用来描述或者匹配一系列符合某个句法规则的字符串。  选择  |表示分割，比如： boy | gril可以匹配“boy”或者“girl”  数量限定  数量限定出了上面讲过的\*，还有+和?,如果在一个模式中不加数量限定符则表示出现一次且仅出现一次：  +表示前面的字符必须出现至少一次（一次或多次），例如，”goo+gle”可以匹配‘goooogle’,’gooooogle’等；  ?表示前面的字符最多出现一次（零次或一次），例如，“colou?r”可以匹配‘colour’和‘color’；  \*表示前面的字符可以不出现，也出现一次或者多次，例如，‘0\*42’可以匹配42，042，0042，00042等等。  范围和优先级  ()圆括号可以用来定义模式字符串的范围和优先级，可以简单的理解位是否将括号内的模式串作为一个整体。例如，”gr(a|r)y”等价于gray|grey；”(grand)?father”匹配father和grandfather。  7. find命令的搜寻条件直接关系到输出结果，默认应该是按从左到右的的顺序判断，如果有逻辑运算（-not、-and、-or）还得再判断条件组合。  find查询的结果是找到一个匹配的项就立即输出结果，一边查找一边输出，查找到的内容不是一次性输出的，所以可能没有统一排序。  但是具体排序规则需要分析find命令的底层实现了  8. Shell程序的三种运行方式：  第一种方式，直接执行权限不足。  第二种方式在执行之前，我们先收回这个脚本的执行权限并做验证。  第三种方式：这个脚本可以直接执行，而且可以显示脚本里面的变量值。就好像这个变量被带出来了一样。 | | | |

1. 当次实验结束后一周内按班组织上交实验报告。

2. 实验报告文件命名为：“学号+姓名”，格式为WORD文档。