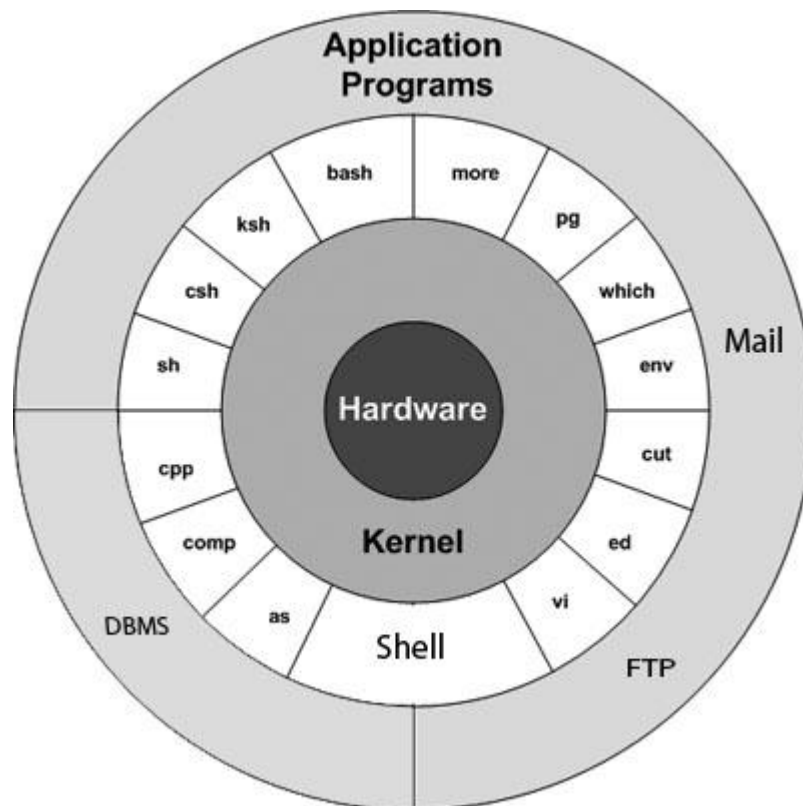


Shell脚本编程基础



内容

①

- Linux命令帮手
 - 命令行编辑
 - screen工具
- Shell特殊字符（1）
 - 通配符
 - 正则表达式

②

- Shell特殊字符（2）
 - 引号
 - 转义符与路径符
 - 输入输出重定向
 - 注释和后台命令
 - 命令组合符
 - 成组命令

③

- Shell简介
 - Shell的概念
 - Shell的特点
 - Shell版本
 - Shell程序示例
 - Shell程序执行方法
- Shell变量
 - 用户自定义变量
 - 位置变量
 - Shell预定义变量
 - 环境变量

④

- 算术运算
- 控制结构
- 函数
- 作业控制
- Shell内置命令
- Shell脚本调试
- Shell脚本实例

内容

①

- **Linux命令帮手**
 - 命令行编辑
 - screen工具
- **Shell特殊字符（1）**
 - 通配符
 - 正则表达式

②

- **Shell特殊字符（2）**
 - 引号
 - 转义符与路径符
 - 输入输出重定向
 - 注释和后台命令
 - 命令组合符
 - 成组命令

③

- **Shell简介**
 - Shell的概念
 - Shell的特点
 - Shell版本
 - Shell程序示例
 - Shell程序执行方法
- **Shell变量**
 - 用户自定义变量
 - 位置变量
 - Shell预定义变量
 - 环境变量

④

- 算术运算
- 控制结构
- 函数
- 作业控制
- Shell内置命令
- Shell脚本调试
- Shell脚本实例

第6章 Linux命令帮手

第7章 Shell特殊字符

第一节 通配符

第二节 正则表达式

第三节 引号

第四节 转义符与路径符

第五节 输入输出重定向

第六节 注释和后台命令

第七节 命令连接符

第八节 成组命令



第6章 Linux命令帮手



Linux命令帮手

- 命令行编辑
- Screen工具

命令行编辑

- 命令行自动补齐
- 命令历史
- 光标跳转
- 命令编辑

自动补齐

➤ 命令自动补齐

\$ history #输入到his时按Tab键，看看会发生什么

➤ 文件名自动补齐

\$ cat hello.c #输入到cat h时按Tab键，看看会发生什么。如果没有什么发生，就再按一次Tab键看看！

命令历史

- 显示历史命令:

- history [num]
- ↑/ctrl+p, ↓/ctrl+n
- ctrl+r

- 执行历史命令:

- !!或!-1 重复上一条命令
- !n 重新执行第n条历史命令
- !-n 重新执行倒数第n条历史命令
- !string 重新执行以字符串string开头的最近的历史命令
- !?string? 重新执行包含string的最近的历史命令

光标跳转

- Ctrl + a 跳转至命令行首 (Ahead of line)
- Ctrl + e 跳转至命令行尾 (End of line)
- Ctrl + f 向前移动一个字符 (Forward)
- Ctrl + b 向后移动一个字符 (Backward)
- Alt + f 向前跳转到下一个字的第一个字符
- Alt + b 向后跳转到下一个字的第一个字符

命令编辑

- Ctrl + w 删除一个词
- Ctrl + u 删除光标前面的所有字符
- Ctrl + k 删除光标所在位置及其后的所有字符
- Ctrl + d 删除光标所在位置的字符
- Ctrl + c 取消本次命令输入
- Alt + d 删除从光标当前位置，到当前词的结尾字符
- Ctrl + y 粘贴最后一次删除的内容

screen工具

- 我们常需要SSH 或者telnet 远程登录到Linux 服务器，运行一些需要很长时间才能完成的任务，在此期间不能关掉窗口或者断开连接，否则这个任务就会被杀掉，半途而废。
- 我们可以用screen命令解决这个问题。screen命令可以实现当前窗口与任务分离，即使我们离线了，服务器仍在后台运行任务。当我们重新登录服务器，可以读取窗口线程，重新连接任务窗口。

screen的会话与会话窗口



```
$ screen -ls
```

```
There are screens on:
```

```
166. tty1.DESKTOP-QONARUO
```

```
(11/14/22 09:07:58)
```

```
(Attached)
```

```
157. tty1.DESKTOP-QONARUO
```

```
(11/14/22 09:07:45)
```

```
(Detached)
```

```
2 Sockets in /run/screen/S-xiezy.
```

screen的使用方法

会话操作:

- 创建screen会话: `screen`
- 列出screen会话: `screen -ls`
- 恢复screen会话: `screen -r [n]`
- 退出screen会话: `screen -d [n]` #在会话外断开已连接（attached）的会话
- 脱离screen会话: `Ctrl+a, d` #在会话中断开（detach）该会话

会话中的窗口操作:

- 创建会话窗口: `Ctrl+a, c`
- 会话窗口切换:
 - `Ctrl+a, w` 显示所有窗口列表
 - `Ctrl+a, n` 切换到下一个窗口
 - `Ctrl+a, p` 切换到前一个窗口
 - `Ctrl+a, 0..9` 切换到第0..9个窗口
 - `Ctrl+a, [Space]` 由窗口0循序切换到窗口9
 - `Ctrl+a, Ctrl+a` 在两个最近使用的窗口间切换
- 退出会话窗口: `exit`

第7章 Shell 特殊字符

通配符

正则表达式

引号

转义符与路径符

输入输出重定向

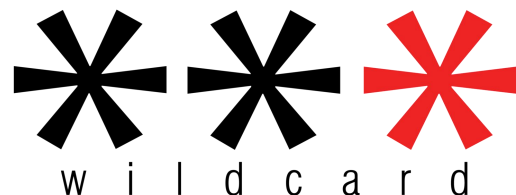
注释和后台命令

命令连接符

成组命令

通配符

（星号）	如file、*.fa	#任意个任意字符
?（问号）	如file?.txt	#1个任意字符
[字符集合]	如[13579] [agh]	#集合中的任意一个字符
[字符范围]	如[0-9] [a-z]	#范围中的任意一个字符
!或^	如file[!3-9]	#除了某范围或集合中的任意一个字符



通配符实例

\$ ls

ant antilope bat bear camel cat cow crab dog duck fish fox frog hog pig wolf

\$ ls *g

dog frog hog pig

ant antilope bat bear camel cat cow crab dog duck fish fox frog hog pig wolf

\$ ls ?og

dog hog

ant antilope bat bear camel cat cow crab dog duck fish fox frog hog pig wolf

\$ ls [a-c]*

ant antilope bat bear camel cat cow crab

ant antilope bat bear camel cat cow crab dog duck fish fox frog hog pig wolf

\$ ls [!a-c]*

dog duck fish fox frog hog pig wolf

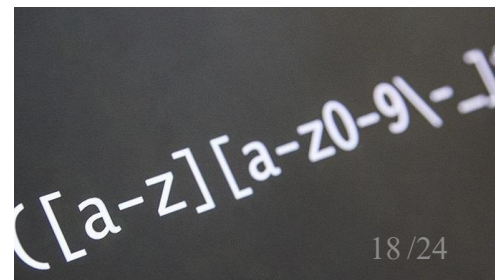
\$ ls [dfhp]*g

dog frog hog pig

ant antilope bat bear camel cat cow crab dog duck fish fox frog hog pig wolf

正则表达式

- **正则表达式**（Regular Expression）是一种文本模式，包括**普通字符**和特殊字符（称为“元字符”）。正则表达式使用单个字符串来描述、匹配一系列符合某个规则的字符串。
- grep、sed、vi、less等命令都支持正则表达式，另外Perl、Python、PHP、JavaScript、R、Ruby、AWK等脚本语言及Java、C#、Delphi等编译语言也都支持正则表达式。
- 正则表达式包括**基本正则表达式**和**扩展的正则表达式**。



基本正则表达式

元字符	含义
*	匹配之前的项0次或多次
.	匹配除了换行符外的任意一个字符
^	匹配行首，如^M 即 匹配以M开头的行
\$	匹配行尾，如G\$ 即 匹配以G结尾的行
[]	匹配中括号中指定的任意一个字符，如[aeiou]匹配任意一个元音字母，[a-z][0-9]匹配一个小写字母和一位数字构成的两个字符
[^]	匹配除中括号内的字符以外的任意一个字符
\	转义符，用于取消特殊符号的含义

基本正则表达式示例

```
$ cat nucleic_acid
```

DNA

RNA

mRNA

tRNA

rRNA

snRNA

snoRNA

miRNA

lncRNA

```
$ grep '[mt]RNA' nucleic_acid
```

mRNA

tRNA

```
$ grep '[^mtr]RNA' nucleic_acid
```

snRNA

snoRNA

miRNA

lncRNA

扩展的正则表达式

元字符	含义
?	匹配之前的项0次或1次
+	匹配之前的项1次或多次
()	匹配表达式，创建一个用于匹配的字串
{n}	匹配其前面的字符恰好出现n次，如[0-9]\{2\} 匹2位数字
{n,}	匹配其前面的字符出现不小于n次
{n,m}	匹配其前面的字符至少出现n次，最多出现m次，如[a-z]\{6,8\} 匹配6到8个小写字母
	匹配 两边的任意一项

扩展的正则表达式示例

```
$ cat nucleic_acid
```

DNA

RNA

mRNA

tRNA

rRNA

snRNA

snoRNA

miRNA

lncRNA

```
$ grep -E '^.?RNA' nucleic_acid
```

RNA

mRNA

tRNA

rRNA

```
$ grep -E "[a-z]{3}RNA" nucleic_acid
```

snoRNA

lncRNA

通配符与正则表达式的区别

- **通配符**是用来**匹配文件名的**，是**完全匹配**；
- **正则表达式**是用来**匹配文件中的文本的**，是**包含匹配**，如**grep**查找一个正则表达式时，任意包含该表达式的行都会显示出来。

课后作业7

在本目录下编辑shell脚本文件ex_07.sh并运行，运行该脚本能实现如下功能：

- (1) 当前目录（运行ex_07.sh时的当前目录）下新建目录ex_dir；
- (2) 进入目录ex_dir；
- (3) 在目录ex_dir中新建文件ex_07.txt；
- (4) 往文件ex_07.txt中写入内容"This is exercise 7"（不包括引号）；
- (5) 复制文件/home/pub/seq/at_LEC1_CDS.fa到目录ex_dir中。

