Technology Tarena 达内科技

Unix基础

C/C++教学体系

“

个人简介

闵卫
minwei@tarena.com.cn

”

Technology Tarena 达内科技

“

Unix操作系统及其常用命令

”

全程目标

- 计算机系统组成
- 永远的Unix
- Unix的发展方向
- Unix的基本概念
- Unix的常用命令

计算机系统组成

- 硬件
  - 中央处理单元(微处理器)
    执行指令，处理数据，如CPU等
  - 内部存储器
    临时存储少量指令和数据，如内存条、显存等
  - 外部存储器
    永久存储海量指令和数据，如硬盘、光盘、U盘等
  - 输入输出部件
    与人或其它计算机系统交互，如键盘、鼠标、显示器、打印机等

计算机系统组成

- 软件
  - 操作系统
    如Unix/Linux、 Windows、 Android、iOS等
    - ✓ 内核(kernel)
      管理存储器、文件、外设等各种软硬件资源
    - ✓ 外壳(shell)
      为用户提供操作界面，接受命令，执行程序
  - 应用软件
    如Word、QQ、Firefox等

流行的Linux发行版本

- Red Hat
  - 支持平台最多
- Debian
  - 最遵循GNU规范
- Ubuntu
  - 简单，易用，安全
- Slackware
  - 历史悠久，复杂，易用性差
- SUSE
  - 深受德国人民喜爱
- Fedora
  - 源自开源社区与Red Hat工程师的合作，技术支持好
- 红旗
  - 中科院软件所，完善的中文支持

永远的Unix

- 史前时代：1961-1969
  - CTSS, Compatible Time-Sharing System
    兼容分时系统，小而简单的实验室原型
  - Multics, Multiplexed Information and Computing Service
    多路信息与计算服务，庞大复杂，不堪重负
  - Unics, Uniplexed Information and Computing Service
    单路信息与计算服务，返朴归真，走上正道

永远的Unix

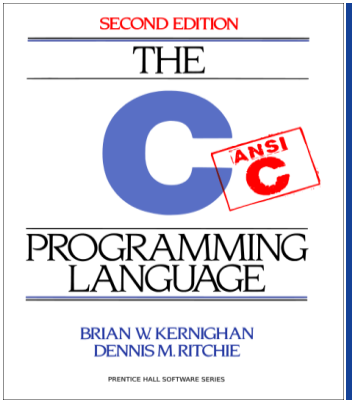
- 创世纪：1969-1971
  - 1969年，贝尔实验室，Ken Thompson和Dennis Ritchie为PDP-7编写Unix操作系统的核心

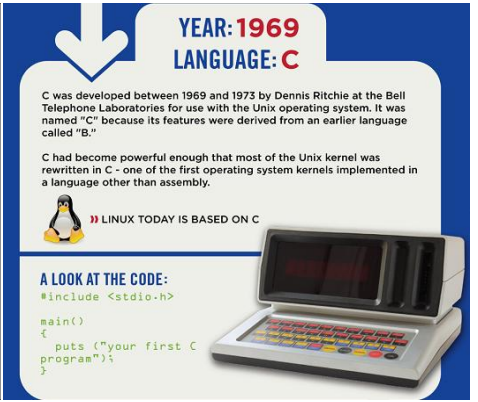
永远的Unix

- 创世纪：1969-1971
  - 1970年，Ken Thompson在BCPL语言的基础上发明了B语言
  - 1971年，第一个Unix应用程序nroff，用B语言和汇编语言混合编写，并在PDP-11上运行

永远的Unix

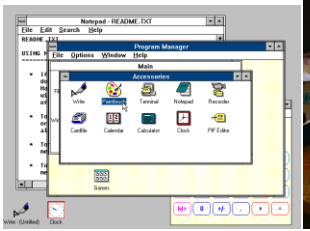
- 出埃及记：1971-1980
  - C语言从1969年起自B语言进化而来。Thompson和Ritchie于1971年成功地用C重写了整个Unix系统
  - 1979年，贝尔实验室发布Unix V7版本，成为Unix程序员公认的第一个真正意义上的Unix操作系统





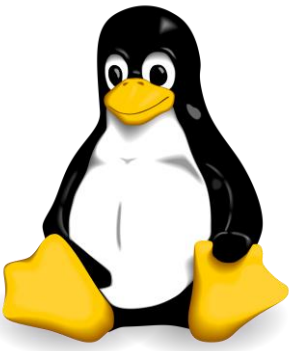
永远的Unix

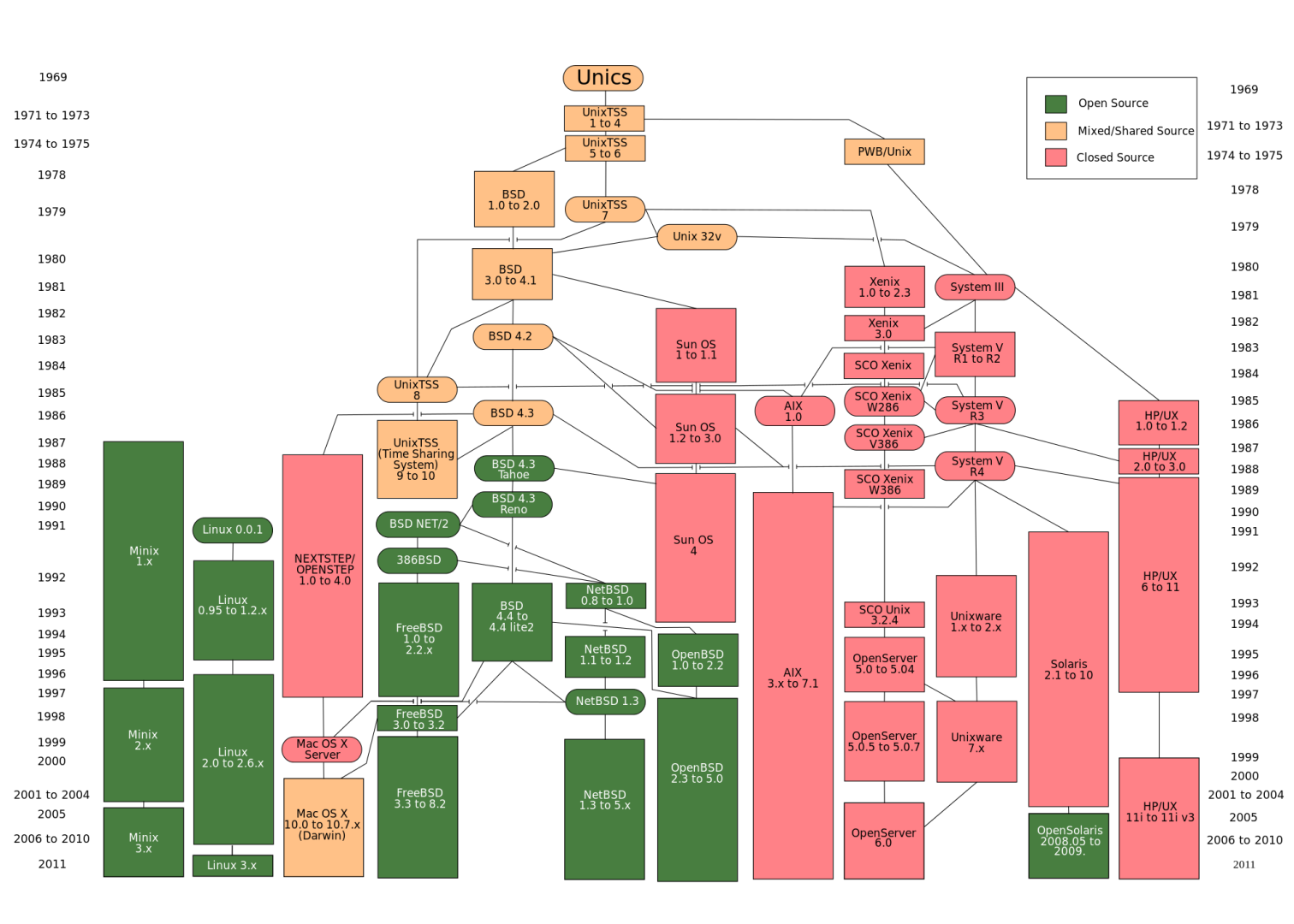
- 帝国崛起： 1980-1990
  - 1983年，第一次Unix战争在AT&T的Unix System V和伯克利的BSD Unix之间爆发
  - 1988年，以IBM、DEC、HP为首的开放软件基金会(OSF)与AT&T/Sun轴心对抗，史称第二次Unix战争
  - 1990年，Windows 3.0发布，来自微软的第一个成功的图形操作系统，为其在九十年代荡平并最终垄断桌面应用市场创造了条件



永远的Unix

- 绝地反击： 1991-1995
  - 1991年，芬兰大学生Linus Torvalds宣布了Linux项目。在互联网爆炸的背景下，Linux的开发相较于此前各种版本的Unix更具优势
  - 1995年，得益于整套GNU工具包的支撑，Linux上软件之多、质量之高，已经达到一个产品级操作系统的水准




1969
1971 to 1973
1974 to 1975
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 to 2004
2005
2006 to 2010
2011
Open Source
Mixed/Shared Source
Closed Source
Unics
UnixTSS 1 to 4
UnixTSS 5 to 6
UnixTSS 7
UnixTSS 8
UnixTSS (Time Sharing System) 9 to 10
PWB/Unix
Unix 32v
BSD 1.0 to 2.0
BSD 3.0 to 4.1
BSD 4.2
BSD 4.3
BSD 4.3 Tahoe
BSD 4.3 Reno
BSD NET/2
386BSD
FreeBSD 1.0 to 2.2.x
FreeBSD 3.0 to 3.2
FreeBSD 3.3 to 8.2
BSD 4.4 to 4.4 lite2
NetBSD 0.8 to 1.0
NetBSD 1.1 to 1.2
NetBSD 1.3
NetBSD 1.3 to 5.x
OpenBSD 1.0 to 2.2
OpenBSD 2.3 to 5.0
Sun OS 1 to 1.1
Sun OS 1.2 to 3.0
Sun OS 4
AIX 1.0
AIX 3.x to 7.1
Xenix 1.0 to 2.3
Xenix 3.0
SCO Xenix
SCO Xenix W286
SCO Xenix V386
SCO Xenix W386
SCO Unix 3.2.4
OpenServer 5.0 to 5.04
OpenServer 5.0.5 to 5.0.7
OpenServer 6.0
System III
System V R1 to R2
System V R3
System V R4
Unixware 1.x to 2.x
Unixware 7.x
Solaris 2.1 to 10
OpenSolaris 2008.05 to 2009.
HP/UX 1.0 to 1.2
HP/UX 2.0 to 3.0
HP/UX 6 to 11
HP/UX 11i to 11i v3
Minix 1.x
Minix 2.x
Minix 3.x
Linux 0.0.1
Linux 0.95 to 1.2.x
Linux 2.0 to 2.6.x
Linux 3.x
NEXTSTEP/ OPENSTEP 1.0 to 4.0
Mac OS X Server
Mac OS X 10.0 to 10.7.x (Darwin)

Unix的发展方向

- 服务器
  - Linux最初就是从服务器领域开始迅速发展起来的，各大Linux厂商的主要收入均来自服务器应用市场
- 嵌入式
  - Linux广泛支持各种设备，且免费、开源，近年来嵌入式Linux系统取得了飞速发展
- 桌面系统
  - 随着开源文化的发展，越来越多的人投身于开源事业，Linux在桌面系统方面取得了令人瞩目的成就

Unix的基本概念

- Shell
  - Shell俗称外壳，是提供使用界面的软件，即命令解释器。它负责接受用户输入的命令，然后调用相应的应用程序，并向用户输出程序运行的结果
  - Shell又是一种脚本语言。它可以自动解释并执行预先编写的一组命令，并可以定义变量，设置参数，提供许多类似高级语言的流程控制结构
  - 常见Shell
    - ✓ b-sh/bash
    - ✓ csh/tcsh

Unix的基本概念

- 文件系统
  - 文件系统是操作系统在磁盘或分区上组织文件的方法和数据结构
  - 文件系统负责对磁盘空间进行组织和分配，存储文件数据，并对其提供保护和检索服务
  - 文件系统由文件管理软件、被管理的文件和文件数据结构三部分组成
  - 目录、子目录、父目录、当前目录、主目录、根目录、文件、路径、绝对路径、相对路径

Unix的常用命令

所有命令大小写敏感，格式如下：

命令　[-选项]　[参数]

其中，选项和参数是可选的，中间用空格隔开。

Unix的常用命令

- 清屏
  - clear
- 打印当前目录
  - pwd
- 改变当前目录
  - cd
- 显示目录内容
  - ls
    - ✓ `ls -a`，显示全部内容，包括隐藏文件
    - ✓ `ls -l`，显示详细信息，包括文件属性

Unix的常用命令

```
- rw- r-- r-- 1 root root 6 Jul 25 11:48 myfile
- --- --- --- - ---- ---- - ------------ ------
A  B   C   D  E  F    G    H       I         J
```

A：文件类型，d(目录)/-(普通)/l(软链接)
B：属主权限，r(读)/w(写)/x(执行)/-(无)
C：同组权限，r(读)/w(写)/x(执行)/-(无)
D：其它权限，r(读)/w(写)/x(执行)/-(无)
E：硬链接数
F：属主
G：属组
H：字节数
I：最后修改时间
J：文件名

针对目录的写权限：在该目录下增删子目录或文件。
针对目录的执行权限：访问该目录下的内容。

Unix的常用命令

- 修改权限
  - chmod
    - ✓ 符号模式
      chmod u/g/o/a +/-/= r/w/x 文件
      u/g/o/a：属主/同组/其它/所有
      +/-/=：增加/减去/设置
      r/w/x：读取/写入/执行
    - ✓ 数字模式
      chmod 644 myfile
      将权限用三位数字表示，由左至右依次对应属主、同组和其它，每一位数字都是从4/2/1三个数中选出若干取和，它们分别表示：读取/写入/执行

Unix的常用命令

- 创建文件
  - touch
- 创建目录
  - mkdir
    如果需要一次创建多级目录，需要使用-p选项，如：
    mkdir -p 目录1/目录2/目录3

Unix的常用命令

- 拷贝
  - cp

    拷贝文件　　：cp 源文件 目标文件
    原名拷贝文件：cp 源文件 存在目录
    拷贝目录　　：cp -r 源目录 目标目录
    原名拷贝目录：cp -r 源目录 存在目录

Unix的常用命令

- 更名/移动
  - mv

    更名文件：mv 源文件 目标文件
    移动文件：mv 源文件 存在目录
    更名目录：mv 源目录 目标目录
    移动目录：mv 源目录 存在目录

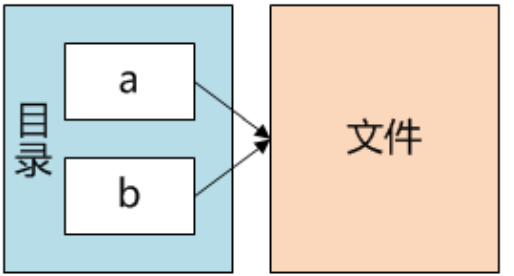
Unix的常用命令

- 删除
  - rm/rmdir

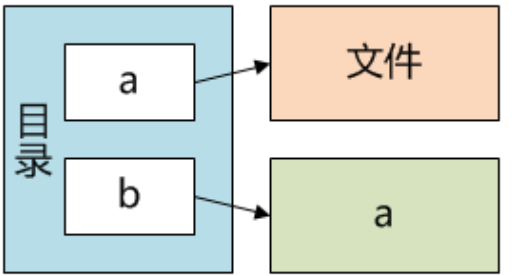
    删除文件　：rm 文件
    删除目录　：rm -r 目录
    删除空目录：rmdir 目录

Unix的常用命令

- 创建链接
  - ln
    - ✓ 硬链接
      ln 目标文件 链接文件
      硬链接的本质就是同一份文件数据与多个不同的文件路径相关联
    - ✓ 软连接
      ln -s 目标文件 链接文件
      软连接的本质就是在一个文件(链接文件)中保存另一个文件(目标文件)的路径信息



硬链接



软链接

Unix的常用命令

- 显示文件
  - cat 文件
    显示文件内容
  - more 文件
    分屏显示文件内容
  - head -n 文件
    显示文件前n行(缺省10行)内容
  - tail -n 文件
    显示文件后n行(缺省10行)内容
  - tail -f 文件
    随文件增长，显示其追加内容

Unix的常用命令

- 查找文件或目录
  - find 目录 条件
    查找特定目录下，满足特定条件的文件或目录。
    其中条件可由以下表达式组合而成：
    - ✓ -name 文件或目录名
    - ✓ -perm 权限
    - ✓ -user 属主
    - ✓ -group 属组
    - ✓ -ctime/atime/mtime -n/+n
    - ✓ -type d/f/l/p/b/c
    - ✓ -size nc

Unix的常用命令

- 查找内容
  - grep 选项 正则表达式 文件/目录
    在指定文件或目录中，查找并输出与正则表达式匹配的文本行。其中选项可为以下组合：
    - ✓ `-n`：同时显示匹配行上下的n行
    - ✓ `-c`：只显示匹配的行数，不显示匹配行的内容
    - ✓ `-h`：当在多个文件中查找时，不显示文件名前缀
    - ✓ `-i`：忽略大小写
    - ✓ `-v`：显示不匹配的行
    - ✓ `-r`：在目录及其子目录中查找
    - ✓ `-w`：将表达式作为一个单词查找

Unix的常用命令

- 查看进程
  - ps：简单显示当前用户有控制终端的进程信息
    - ✓ BSD风格常用选项
      - **a**：所有用户有控制终端的进程
      - **x**：包括无控制终端的进程
      - **u**：以详尽方式显示
    - ✓ SVR4风格常用选项
      - **-e**：所有用户的进程
      - **-a**：当前终端的进程
      - **-f**：按完整格式显示
      - **-u**：特定用户的进程
      - **-g**：特定组的进程

进程信息列表

表项	说明	表项	说明
PID	进程ID	TTY	终端次设备号
PPID	父进程ID	START	进程开始时间
UID	进程属主ID	TIME	进程运行时间
%CPU	CPU使用率	CMD	进程指令
%MEM	内存使用率	NI	进程nice值
RSS	占用物理内存(KB)	PRI	进程优先级
VSZ	占用虚拟内存(KB)	WCHAN	等待内核事件
SZ	占用虚拟内存(页)	STAT	进程状态

进程状态(STAT)

标志	说明	标志	说明
R	运行	<	高优先级
S	可唤醒的睡眠	N	低优先级
D	不可唤醒睡眠	L	有被锁分页
T	停止	s	会话首进程
X	死亡	l	多线程化的进程
Z	僵尸	+	在前台进程组中

练习时间

熟练使用Unix常用命令

“

vi与shell脚本

”

全程目标

- Unix的常用命令
- 文件通配符
- I/O重定向和管道
- 命令分隔符与续行符
- 文本编辑器vi的使用
- 环境变量
- shell脚本

Unix的常用命令

- 杀死进程
  - kill 进程ID
    杀死特定进程
  - kill -信号名/信号值 进程ID
    向特定进程发送信号
  - kill -l
    打印信号列表

传统Unix信号

信号	名称	说明	默认动作
1	SIGHUP	终端接口检测到连接断开	终止
2	SIGINT	用户按中断键(Ctrl+C)	终止
3	SIGQUIT	用户按退出键(Ctrl+\)	终止+core
4	SIGILL	执行非法硬件指令	终止+core
5	SIGTRAP	陷阱调试	终止+core
6	SIGABRT	调用abort函数	终止+core
7	SIGBUS	内存故障	终止+core
8	SIGFPE	算术运算异常	终止+core

传统Unix信号

信号	名称	说明	默认动作
9	SIGKILL	杀死进程，不能被捕获或忽略	终止
10	SIGUSR1	用户自定义	终止
11	SIGSEGV	非法内存访问	终止+core
12	SIGUSR2	用户自定义	终止
13	SIGPIPE	管道或流式套接字异常	终止
14	SIGALRM	alarm时钟或真实计时器到期	终止
15	SIGTERM	终止进程，可以被捕获或忽略	终止
16	SIGSTKFLT	数学协处理器发生栈故障	终止

传统Unix信号

信号	名称	说明	默认动作
17	SIGCHLD	子进程状态改变	忽略
18	SIGCONT	使停止的进程继续	继续/忽略
19	SIGSTOP	停止一个进程	停止
20	SIGTSTP	用户按停止键(Ctrl+Z)	停止
21	SIGTTIN	后台进程读控制终端	停止
22	SIGTTOU	后台进程写控制终端	停止
23	SIGURG	紧急情况发生或收到带外数据	忽略
24	SIGXCPU	超过软CPU时间限制	终止+core

传统Unix信号

信号	名称	说明	默认动作
25	SIGXFSZ	超过软文件长度限制	终止+core
26	SIGVTALRM	虚拟计时器到期	终止
27	SIGPROF	实用计时器到期	终止
28	SIGWINCH	以ioctl函数更改终端窗口大小	忽略
29	SIGIO	异步I/O事件	终止
30	SIGPWR	电源失效	终止
31	SIGSYS	无效系统调用	终止+core

Unix的常用命令

- 切换用户
  - su 用户名
    切换到特定用户(缺省root)
  - su - 用户名
    以新用户(缺省root)身份重新登录
    大部份环境变量和工作目录都会改变

Unix的常用命令

- 修改密码
  - passwd
    修改当前用户的密码
  - passwd 用户名
    修改特定用户的密码
    仅root用户可使用此命令

Technology Tarena 达内科技

练习时间

熟练使用Unix常用命令

文件通配符

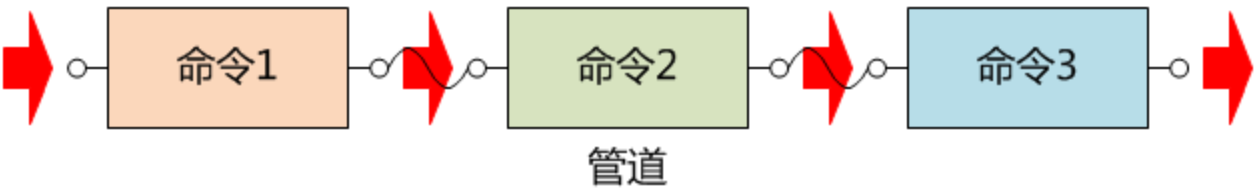
- *
  - 通配若干任意字符
    ls *.txt
- ?
  - 通配一个任意字符
    ls file_?.txt
- []
  - 通配一个在特定字符集中的字符
    ls file_[a-c].txt

I/O重定向和管道

- >
  - 输出重定向
    echo hello > a.txt
- <
  - 输入重定向
    cat < a.txt > b.txt
- >>
  - 追加
    echo world >> b.txt

I/O重定向和管道

- |
  - 管道符
    将前一个命令的输出作为后一个命令的输入
    ls -l /etc | more



命令分隔符与续行符

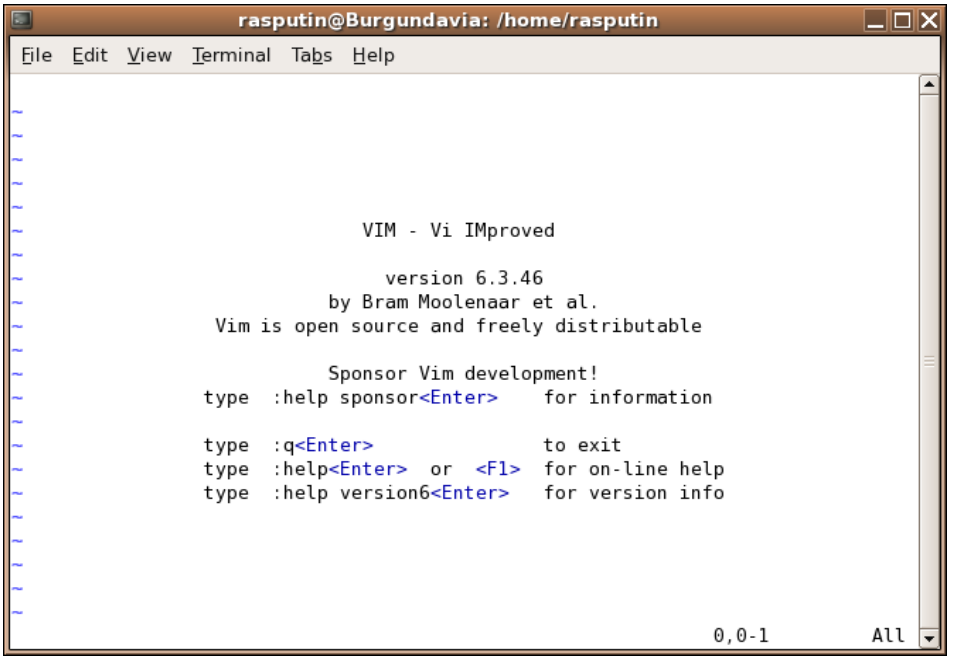
- ;
  - 命令分隔符，在一个命令行中分隔多个命令
    cal; pwd; date
- \
  - 续行符，继续在下一行输入命令
    cat \
    /etc/passwd \
    | \
    sort \
    > \
    user.txt

练习时间

尝试I/O重定向和管道的用法

文本编辑器vi的使用

- 命令选项
- 工作模式
- 基本命令
- 配置文件



vi的命令选项

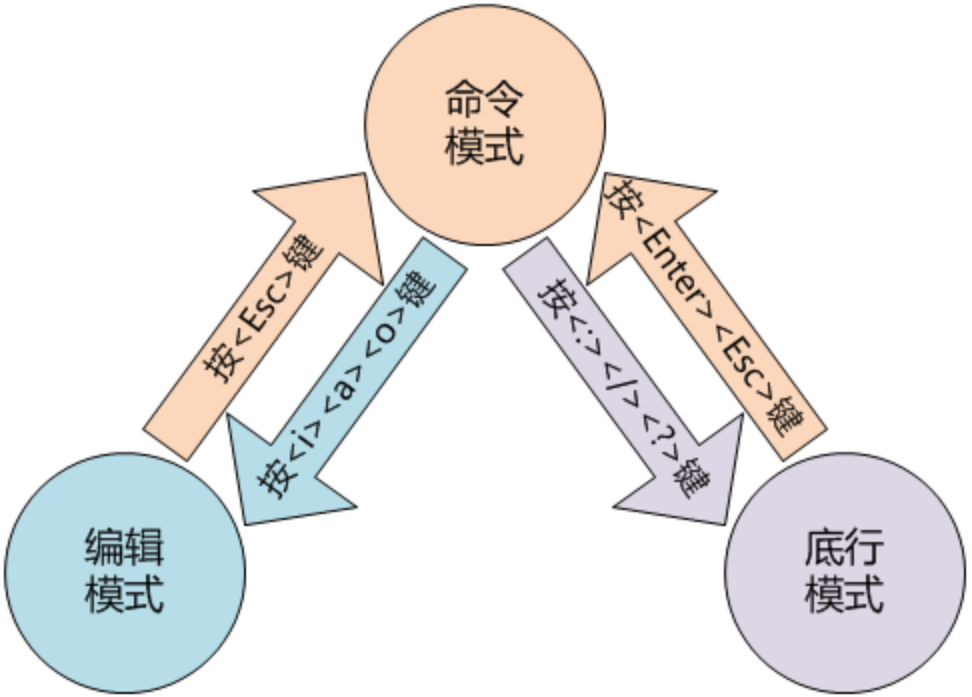
命令选项	说明
vi	编辑无名空文件
vi 不存在的文件	编辑具有特定文件名的新文件
vi 存在的文件	打开并编辑一个已存在的文件
vi +n 文件	打开并编辑文件，将光标置于第n行首
vi + 文件	打开并编辑文件，将光标置于末行首
vi +/pattern 文件	打开并编辑文件，将光标置于第一匹配行首
vi -r 文件	从上次崩溃中恢复，并继续编辑该文件
vi -R 文件	以只读方式打开文件

vi的命令选项

命令选项	说明
vi 文件1 文件2 ...	同时打开并编辑多个文件

- 注意
  - vi命令并不锁定所编辑的文件，因此多个用户可以同时编辑同一个文件。在这种情况下，只有最后一次保存的内容才会被保留。

vi的工作模式



vi的三种工作模式

vi的工作模式

- 在命令模式下按如下键可进入编辑模式
  - **<i>**：在当前位置之前插入
  - **<a>**：在当前位置之后插入
  - **<o>**：在当前行下插入
- 在编辑模式下按**<Esc>**键，返回命令模式

vi的工作模式

- 在命令模式下按如下键可进入底行模式
  - **<:>**：输入底行命令
  - **</>**：从当前位置向文件尾方向搜索
  - **<?>**：从当前位置向文件首方向搜索
- 在底行模式下按**<Enter>**键，执行底行命令并返回命令模式，按**<Esc>**键，取消底行命令并返回命令模式
- 在任何模式下按**<Esc>**键，都能保证将当前模式切换到命令模式

vi的基本命令

- 移动光标
- 翻滚屏幕
- 插入文本
- 删除文本
- 移动文本
- 查找替换
- 复制粘贴
- 取消重做
- 底行命令
- 选项设置

```
1  /* 逆波兰法求解四则运算表达式 */
2
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <string.h>
6  #include "rpn.h"
7  #include "ql.h"
8  #include "sl.h"
9
10 /* 判断运算符的优先级 */
11 static int prior (char op1, char op2) {
12     return strchr ("*/", op1) && strchr ("+-", op2);
13 }
14
15 /* 将中缀表达式转换为后缀表达式 */
16 char* rpn_convert (const char* infix) {
17     QUEUE queue;
18     queue_init (&queue);
19     STACK stack;
20     stack_init (&stack);
21
22     size_t i, len = strlen (infix);
23     /* 遍历中缀表达式中的每一个字符 */
/rpn                                                    6,11          顶端
```

vi的移动光标命令

命令	说明
h	光标左移一个字符
l	光标右移一个字符
k/Ctrl+p	光标上移一行
j/Ctrl+n	光标下移一行
Backspace	光标向文件首方向移一个字符
空格	光标向文件尾方向移一个字符
Enter	光标向文件尾方向移一行
b/B	光标向文件首方向移一个字至字首

vi的移动光标命令

命令	说明
w/W	光标向文件尾方向移一个字至字首
e/E	光标向文件尾方向移一个字至字尾
(	光标移至句首
)	光标移至句尾
{	光标移至段首
}	光标移至段尾
H	光标移至屏幕顶行首
M	光标移至屏幕中行首

vi的移动光标命令

命令	说明
L	光标移至屏幕末行首
0	光标移至当前行首
$	光标移至当前行尾
G	光标移至最后行首
n$	光标下移n行(包括当前行)至行尾
nG	光标移至第n行首
n+	光标下移n行至行首
n-	光标上移n行至行首

vi的翻滚屏幕命令

命令	说明
Ctrl+u	向文件首方向翻半屏
Ctrl+d	向文件尾方向翻半屏
Ctrl+b	向文件首方向翻一屏
Ctrl+f	向文件尾方向翻一屏
nz+	将第n行滚动至屏幕顶
nz-	将第n行滚动至屏幕底
z-	将当前行滚动至屏幕底

vi的插入文本命令

命令	说明
i	在当前位置之前插入
I	在当前行首插入
a	在当前位置之后插入
A	在当前行尾插入
o	在当前行下插入
O	在当前行上插入
nr	替换从当前位置开始的n(缺省1)个字符
R	从当前位置开始替换，直至按<Esc>键

vi的插入文本命令

命令	说明
ns	从当前位置删除n(缺省1)个字符并插入
nS	从当前行删除n(缺省1)行并插入
C	从当前位置开始删至行尾并插入
ncw	从当前位置开始删除n(缺省1)个字后插入
~	切换当前位置字符大小写
J	连接当前行与下一行

vi的删除文本命令

命令	说明
d0	删至行首
d$	删至行尾
ndw	删除从当前位置开始的n(缺省1)个字
ndd	删除从当前行开始的n(缺省1)行
nx	删除从当前位置开始的n(缺省1)个字符
nX	删除当前位置之前的n(缺省1)个字符

vi的移动文本命令

命令	说明
>h/l	当前行右移一个制表位
>nk	当前行及其上n(缺省1)行右移一个制表位
>nj	当前行及其下n(缺省1)行右移一个制表位
n>>	从当前行开始的n(缺省1)行右移一个制表位
<h/l	当前行左移一个制表位
<nk	当前行及其上n(缺省1)行左移一个制表位
<nj	当前行及其下n(缺省1)行左移一个制表位
n<<	从当前行开始的n(缺省1)行左移一个制表位

vi的查找替换命令

命令	说明
/pattern	从当前位置向文件尾方向查找匹配文本
?pattern	从当前位置向文件首方向查找匹配文本
n	沿相同方向继续查找
N	沿相反方向继续查找
:s/p1/p2/g	将当前行中所有的p1替换为p2
:n1,n2 s/p1/p2/g	将第n1到n2行中所有的p1替换为p2
:g/p1/s//p2/g	将文件中所有的p1替换为p2

vi的复制粘贴命令

命令	说明
yw	复制一个字
nyy	复制从当前行开始的n(缺省1)行
p	粘贴到当前位置或行之后

vi的取消重做命令

命令	说明
u	取消先前的命令
:undo	取消先前的命令
.	重做先前的命令

vi的底行命令

命令	说明
:n	光标移至第n行首
:n1 co n2	将第n1行拷贝到第n2行之后
:n1,n2 co n3	将第n1到n2行拷贝到第n3行之后
:n1 m n2	将第n1行移动到第n2行之后
:n1,n2 m n3	将第n1到n2行移动到第n3行之后
:n d	将第n行删除
:n1,n2 d	将第n1到n2行删除
:w	保存

vi的底行命令

命令	说明
:w!	保存只读文件
:q	退出
:q!	不保存直接退出
:x	保存并退出
:x!	保存只读文件并退出
:e 文件	在当前编辑环境中打开文件
:e!	重新装入当前文件，丢弃之前的所有修改
:e#	开始编辑另一个文件

vi的底行命令

命令	说明
:r 文件	将文件内容插入到当前行之后
:n	编辑打开文件列表中的下一个文件
:! 命令	执行命令
:n w ! 命令	将第n行(缺省所有行)作为输入执行命令
:n1,n2 w ! 命令	将第n1到n2行作为输入执行命令
:r ! 命令	将命令的输出插入到当前行之后

vi的选项设置命令

命令	说明
:set all	显示所有选项的当前值
:set nu/nonu	显示行号/不显示行号
:set hlsearch/nohlsearch	加亮查找/不加亮查找
:set autoindent/noautoindent	自动缩进/不自动缩进
:set backup/nobackup	自动备份/不自动备份
:set ignorecase/noignorecase	忽略大小写/不忽略大小写
:syntax on/off	语法加亮/不语法加亮

vi的配置文件

- 在.exrc文件中定义特殊的vi命令
  - 在vi中这些都是底行命令

```
set nu
set autoindent
syntax on
```

练习时间

尝试使用vi编辑文本文件

环境变量

- 环境变量是一个具有特定名字的对象，它包含了一个或多个应用程序所需要的信息
  echo $PATH
  /usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
- 通过命令行可以临时性地修改/设置环境变量，只影响当前会话
  export PATH=$PATH:.
- 通过初始化文件可以永久性地修改/设置环境变量，影响此后的所有登录
  .bash_profile

shell脚本

- 脚本是使用一种特定的描述性语言，依据一定的格式编写的可执行文件，亦称宏程序或批处理文件
- shell脚本可以看做是由一系列shell命令、变量和控制结构组成的文本文件，可被shell脚本解释器解释并执行，其效率略低于二进制可执行程序

```
#!/bin/bash
cal
pwd
date
exit 0
```

Technology Tarena 达内科技

练习时间

尝试编写一个简单的shell脚本，显示当前用户的环境变量、当前目录列表和进程信息

Technology Tarena 达内科技

“

再见

”