

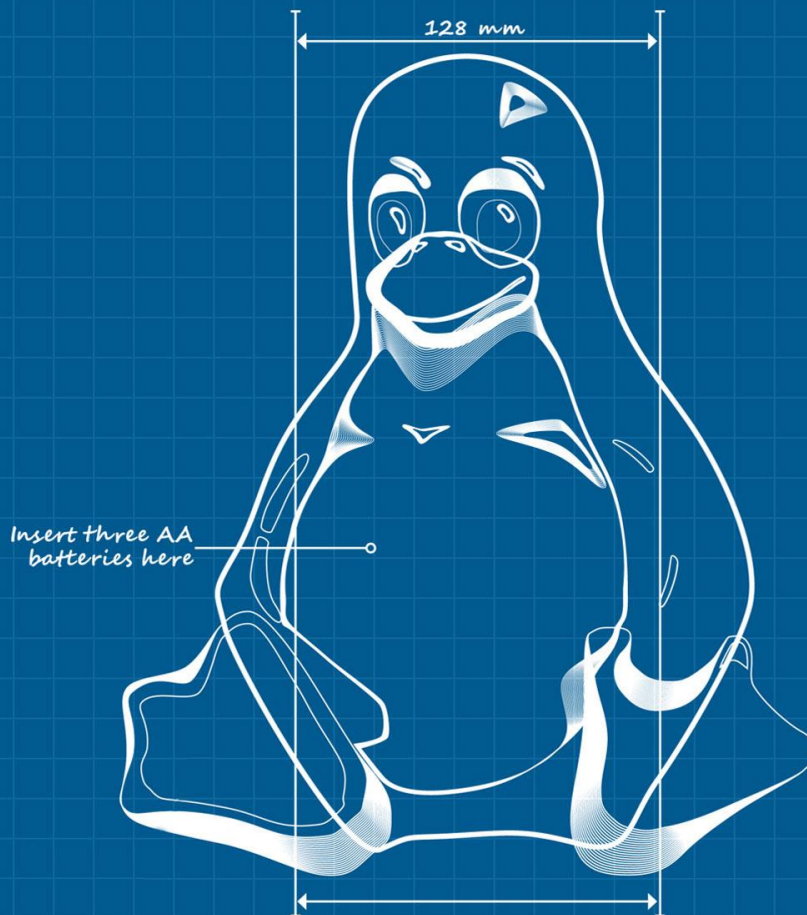
Linux

应用基础

3、基本命令1

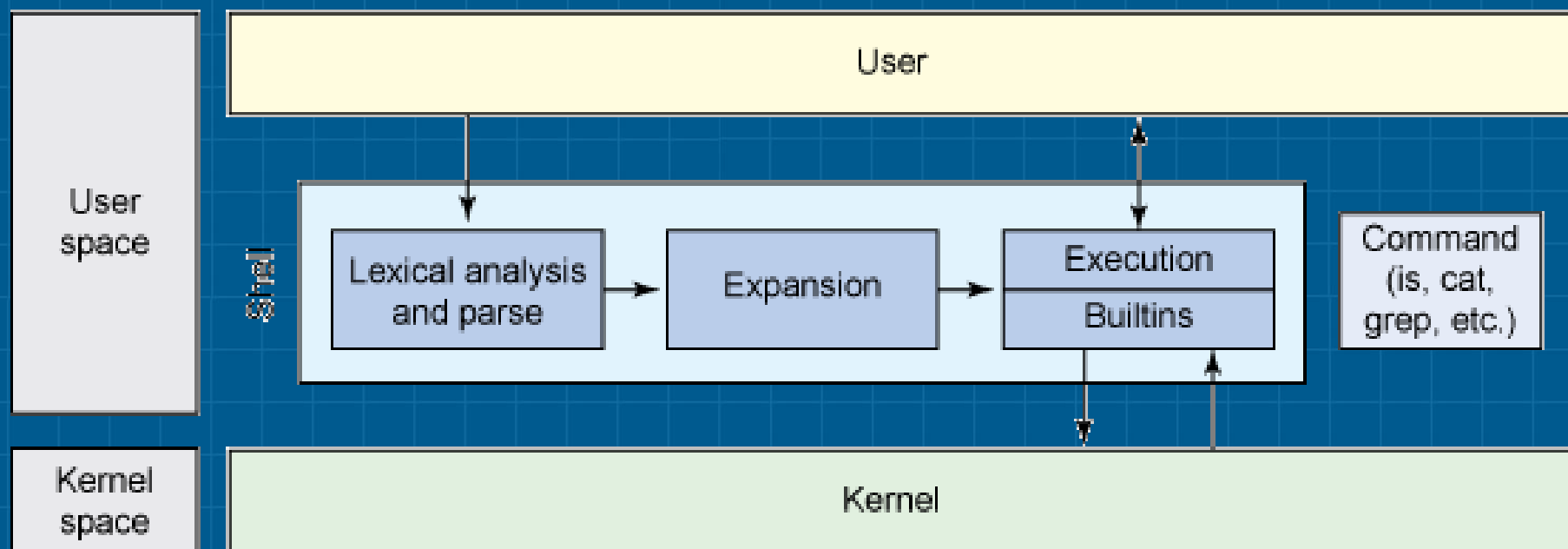
余恒

北京师范大学



Linux 系统组成

- 内核(Kernel)：驱动硬件、管理内存CPU的最底层代码。
- 外壳(Shell)：命令解释器，处理所有命令输入与内核的交互。



CLI & GUI

- CLI (command-line interface):
命令行界面，有稳定的格式，简单的输入输出设定，特别适合进行协作、批处理或者后台定时任务。
- GUI (Graphical User Interface)
图形用户界面，界面设计无疑是门艺术，有鲜明的时代特征和技术烙印，也不可避免会在不久后过时。

为什么要使用终端

- 用户界面的学习成本可以很高。
 - 从按钮分布到菜单安排都需要重新适应
 - 追逐视觉特效就意味着更大的资源开销，更短的使用周期，更多的鼠标操作.....
- 图形化程序之间很难协同工作
- 图形化操作在远程通信时需要大量网络资源

连接服务器

- ip地址：202.112.94.64 端口 7712
- 用户名：学号，密码：学号abc
- 命令：
- `ssh -p 7712 学号@202.112.94.64`
- 工具：
 - Linux & Mac：终端
 - windows: putty
`ftp://202.112.85.96/putty.zip`

认识终端

- 典型终端 (terminal) :
tianwen@tianwen-Lenovo ~ \$
- 提示符 (prompt) : 提示用户输入命令
Bash中为 \$, Csh中为 % , 超级用户 #
- 用户名 (uname) : tianwen
- 机器名 (hostname) : tianwen-Lenovo
- 工作路径 : ~ (即用户个人根目录)

帮助系统

- 命令 MAN
- 帮助的使用方法可参考 `man man`
- 帮助文件是用less命令开启的，操作如下：
 - 空格键：向下翻动一页。
 - `[pagedown]`：向下翻动一页。
 - `[pageup]`：向上翻动一页。
 - `/` 字符串：向下搜索"字符串"的功能。
 - `?` 字符串：向上搜索"字符串"的功能。
 - `q`：离开less程序。

帮助结构

- **man 输出格式**

- 1.NAME 命令名/函数名，以及关于命令/函数的简单说明
- 2.SYNOPSIS 摘要，一般是语法和参数
- 3.DESCRPTION 详细说明
- 4.OPTIONS 相关选项的详细说明
- 5.SEE ALSO 相关命令/函数
- 6.BUGS 已知BUG
- 7.HISTORY 命令/函数历史
- 8.AUTHOR 作者

主机信息

- "uname"命令就是Unix Name的简写。显示机器名，操作系统和内核的详细信息。
- 通过uname -a命令可以显示详细信息。
 - "Linux": 系统内核名
 - "tecmint": 设备的节点名
 - "3.8.0-19-generic": 内核发布版本
 - "#30-Ubuntu SMP": 内核版本
 - "i686": 处理器架构
 - "GNU/Linux": 操作系统名

简单的命令

- `date` 显示当前的时间、日期
 - `-u` 显示UTC时间
 - `+"%F"` 以指定格式显示时间
- `cal` 显示日历
 - `cal 2016`
 - `cal 12 2015`
 - `cal -j 12 2015`
 - `cal 9 1752`
- 终端中的命令包括：命令 [选项] 参数

查找命令

- `which` 查找命令
 - 如果命令存在，返回命令路径
 - 如果命令不存在，报错或无输出
- `type` 备用查找命令
- 终端命令的搜索顺序为：
 - `shell` 内置
 - 系统目录 `/bin`, `/usr/bin` ,
 - 当前目录下的可执行文件

ls 查看目录内容

- -a 全部 (all)。列举目录中的全部文件，包括隐藏文件 (.filename)。位于这个列表的起首处的 .. 和 . 依次是指父目录和当前目录。
- -l 长 (long)。列举目录内容的细节，包括权限 (模式)、所有者、组群、大小、创建日期、文件是否是到系统其它地方的链接，以及链接的指向。
- -t 按时间排列察看内容

ls实例

- `ls -l -a` 长模式显示所有文件
- `ls -lt` 长模式显示文件，按时间序
- `ls -ltr` 长模式显示文件，按时间逆序
- `ls --help` 显示命令帮助
- `ls --version` 显示命令版本
- `ls --reverse` 逆序输出
- `ll`

文件权限

- 作为一个多用户系统，Linux有严格的权限设置，所有文件和目录的访问都受权限的限制。

```
-rw-r--r-- 1 root root 483997 Jul 15 17:31 sobsrc. tgz
```

- 第一个项目指定文件类型，它可以显示以下几种：
 - d — 目录
 - (短线) — 常规文件（而不是目录或链接）
 - l — 到系统上其它位置的另一个程序或文件的符号链接
- 在第一个项目之后的三组中，你可以看到下面几种类型：
 - r — 文件可以被读取
 - w — 文件可以被写入
 - x — 文件可以被执行（如果它是程序的话）

权限的数字表示

- 每种权限设置都可以用一个数值来代表：
 $r = 4$ $w = 2$ $x = 1$ $- = 0$

例：

-rw-r--r-- (644) — 只有所有者才有读取和写入的权限；组群和其他人只有读取的权限。

-rwxr-xr-x (755) — 所有者有读取、写入、和执行的权限；组群和其他人只有读取和执行的权限。

权限变更方法

- 范围包括

u：表示文件所有者

g：表示同组用户

o：表示其它用户

a：表示所有用户

- 操作可以为：

+：添加某个权限

-：取消某个权限

=：赋予给定的权限，并取消原有的权限

- 例子：

```
chmod a+x hello.sh
```

```
chmod 755 hello.sh
```


随堂练习：脚本

- 在用户目录下建立文件
- /home/XXX/test
- 在test中输入如下内容

```
# first script
```

```
cal
```

```
echo $HOME
```

- 更改权限test 为所有人可执行
- 运行 test , 查看结果

cd 改变路径

- `cd` 返回登录目录
- `cd ~` 返回登录目录
- `cd /` 进入系统根目录
- `cd ..` 向上移动一级目录
- `cd ~ XX` 进入XX的登录目录（如果有访问权限）
- `cd /dir/subdir` 进入绝对路径
- `cd ../dir/` 进入相对路径

目录相关

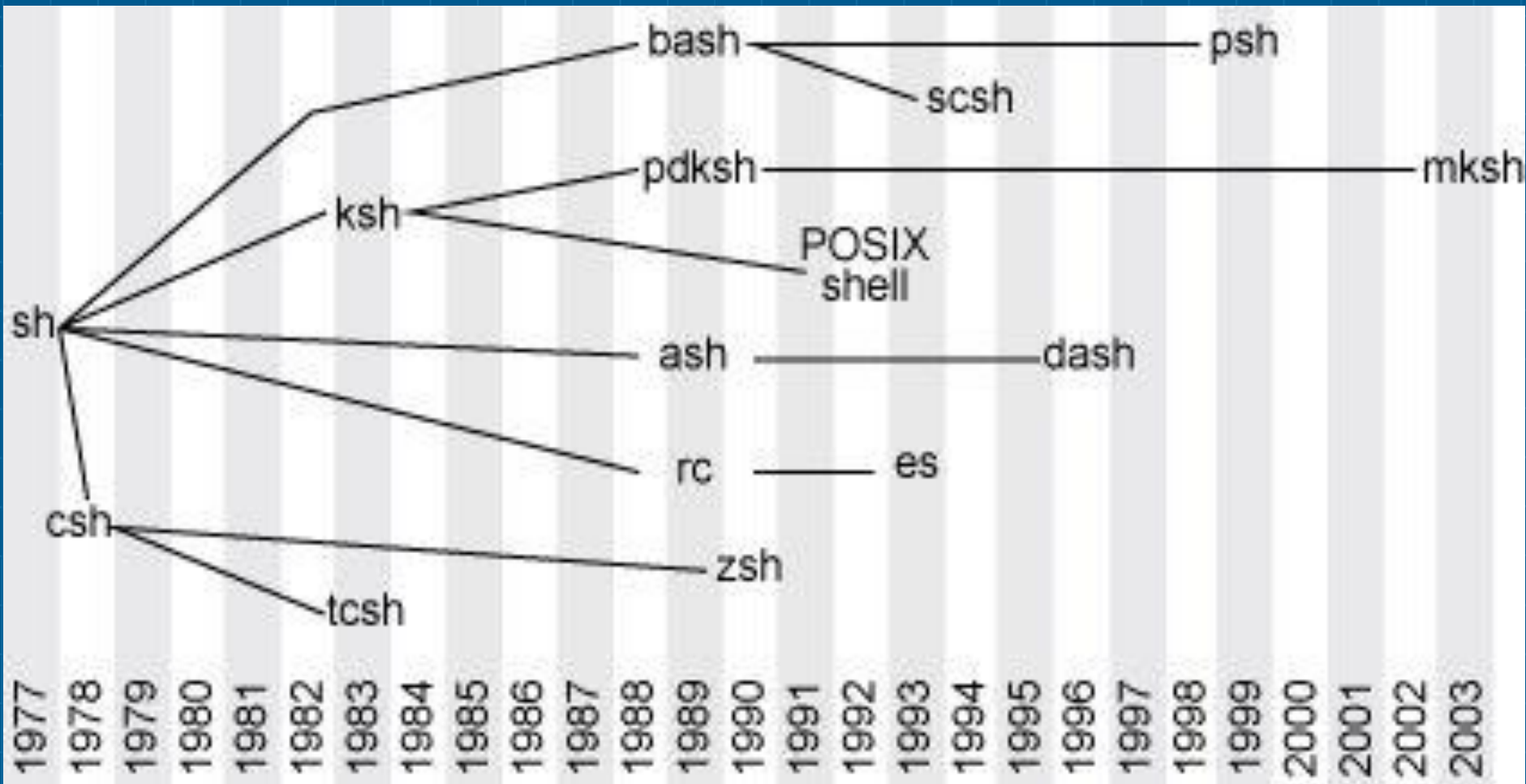
- ls 列出当前目录文件
- pwd 查看当前目录
- cd 进入指定目录
- touch 建立新文件
- cp 复制文件或目录
- rm 删除文件或目录
- mv 移动文件或目录
- mkdir 建立目录
- rmdir 删除目录

cd练习

- 在用户目录下建立文件
/home/XXX/dir/dir1/dir2/test
- 将目录 dir2 移动到 dir目录下
- 删除目录 dir1
- 删除目录 dir2



Shell 类型



Bash特性

- 命令补齐 (Command-Line Completion)
- 通配符 (wildcard character) *?[]
- 命令历史 (command history)
- 输入重定向 (input redirection) <
- 输出重定向 (output redirection) >
- 管道符 (pipe symbol) |
- 用户配置 (user profile)

- **命令补齐** 按下 Tab 键，bash 会自动补齐命令(文件目录名)并显示在屏幕上，如果有多种可能，则全部列出。

- **通配符**

使命令输入变得更简单的方法是在命令中使用通配符。bash 支持三种通配符：

* 匹配任何字符和任何数目的字符

? 匹配任何单字符

[...] 匹配任何包含在括号里的单字符

- **命令历史记录**

用箭头可以回溯之前输入的命令。事实上，bash 将先前输入的命令文本保存在一个历史列表中 `.bash_history`。可以用 `history` 命令来显示命令历史列表。

`#history [n]`

`#fc [-e editor_name] [-n] [-l] [-r] [first] [last]`

`#!c` 重复执行上个以字母c打头的命令

变量

- `A="a b c"` # 为变量赋值
- `A=`date`` # 将命令结果赋给变量
- `A=$(date)` # 将命令结果赋给变量
- `echo $A` # 输出变量
- `read name` # 交互输入变量
- `readonly name` # 设置为只读变量
- `export name` # 定义环境变量
- `Stat$nu=2222` # 变量引用变量赋值
- `unset name` # 变量清除

环境变量

- `env` 显示当前用户所有环境变量
- `$HOME` 用户主文件夹
- `$PATH` 命令搜索路径
- `$SHELL` shell 类型
- `$LANG` 语言环境
- `$DESKTOP_SESSION` 桌面环境

终端打印 echo

- 在终端中打印文本
 - **echo** hello world
- 看看下面这些输出有什么不同
 - **echo** *
 - **echo** *
 - **echo** '*' '*'
 - **echo** \$USER \ \$USER
 - **echo** '\$USER' "\ \$USER" "\$USER"

引用

- 单引号 (') 强引用，引用所有内容。
- 双引号 (") 弱引用，引用除美元 \$，反引号 `，反斜线 \ 之外的所有内容
- 反引号 (`) 命令替换
 - 将一条命令嵌入到另一条命令当中
 - 比如：

```
echo 'The time is `date` .'
```

数值运算

- `$ n=20`
- `$ echo $n`
- `$ n=n+1`
- `$ echo $n`
- `$ n=$n+1`
- `$ echo $n`
- 默均认为字符串
- 对整数变量的数值运算可使用`let`或者`expr`实现
- `$ n=20`
- `$ echo $n`
- `$ let n=n+1`
- `$ echo $n`
- `$ n=`expr $n + 1``
- `$ echo $n`

let命令

- let 表达式可包含变量或整数常数
- 在表达式中支持的操作符及其含义为:
 - +, -, *, /, % 加,减,乘,除,取模
 - >>, <<, &, ^, 左移,右移,位与,位异或,位或
 - !, >=, <=, >, <, ==, !=, &&, ||
 - =, +=, -=, *=, /=, %=, <<=, >>=, &=, ^=, |=
 - 表达式中也可以使用括号, 括号或运算优先级的定义与一般计算机语言中的相同.
- `let m=2*10; let r=m**2`
- `let r=m/3; let r=m%3; let r++`

expr 运算

- 加法 : `r=`expr 4 + 5``
- 乘法 : `r=`expr 4 * 5``
- 括号 : `r=`expr \(5 - 3\) * 3 + 1``
- 除法 : `s=`expr $r / 2``
- 求余 : `s=`expr $r % 5``
- 乘方 : `s=`expr $r ** 2``
- `man expr`
- 注意: 运算符前后需要空格, 特殊字符需要转义符

expr 字符处理

- **expr** length abcda
- **expr** index abcda a
- **expr** index abcdea db
- **expr** substr abcda 3 2
- 如何查找字符串中a第二次出现的位置？
 - max=`expr length abcda`
 - ind1=`expr index abcda a + 1`
 - nstr=`expr substr abcda \$ind1 \$max`
 - **expr** index \$nstr a

方括号 []

- 布尔运算环境, test, man [
 - 字符串比较: == 和 !=
 - 整数比较 -eq, -gt, -ge, -lt, -le
- 例子
 - a=10; b=2
 - ["a" == "b"] ; ["a" == "a"]
 - echo \$?
 - [\$a -eq \$b] ; [2 -gt 1]
 - echo \$?

双括号 (())

- `a=1; b=1`
- `echo $((a++))`
- `echo $((++a))`
- `echo $((a**2))`
- `echo $((a==5))`
- `echo $((a<4))`
- `c=$((--a,b++,7))`
- `echo $a,$b,$c`

日期 date

- date 按制定格式输出日期
 - `date "+%Y-%m-%d %H:%M:%S"`
 - `date --date="next Friday" "+%Y-%m-%d"`
 - `date -d "20160920" "+%s"`
- 计算两日期间隔天数

```
D1=`date -d '20070409' +"%s"`  
D2=`date -d '20070304' +"%s"`  
D3=$((D1 - D2))  
echo $((D3/60/60/24))
```

其它终端技巧

- 鼠标中键复制
 - 用鼠标选中部分内容，单击鼠标中键（滚轮），选中内容会被直接复制到光标当前位置处
 - 如果包含换行符会被直接执行。
- 方向键重复命令
 - 使用小键盘区的上、下方向键，可以回调之前输入的命令。