



Python开发之运维基础

讲师：王晓春

本章内容



马哥教育

IT 人的高薪职业学院

- ◆ 操作系统
- ◆ Linux相关介绍
- ◆ Linux哲学思想
- ◆ 获取Linux
- ◆ 虚拟机
- ◆ Bash
- ◆ 简单命令
- ◆ 帮助用法

◆ OS: Operating System , 通用目的的软件程序

硬件驱动

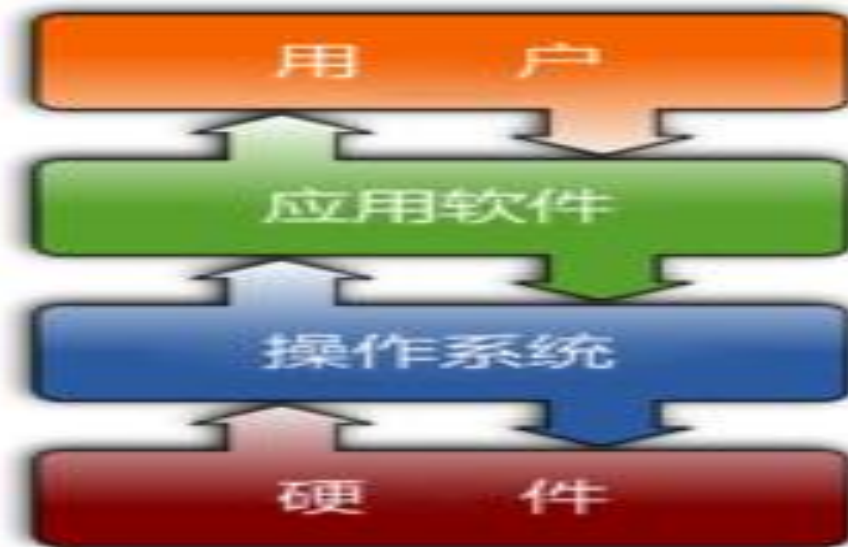
进程管理

内存管理

网络管理

安全管理

文件管理



◆ OS分类：

服务器OS：RHEL, CentOS, Windows Server, AIX

桌面OS：Windows 10, Windows 7, Mac OS, Fedora

移动设备OS：Android, iOS, YunOS

◆ ABI: Application Binary Interface

ABI描述了应用程序与OS之间的底层接口,允许编译好的目标代码在使用兼容ABI的系统中无需改动就能运行

◆ API : Application Programming Interface

API定义了源代码和库之间的接口,因此同样的源代码可以在支持这个API的任何系统中编译

◆ POSIX: Portable Operating System Interface

IEEE在操作系统上定义的一系列API标准

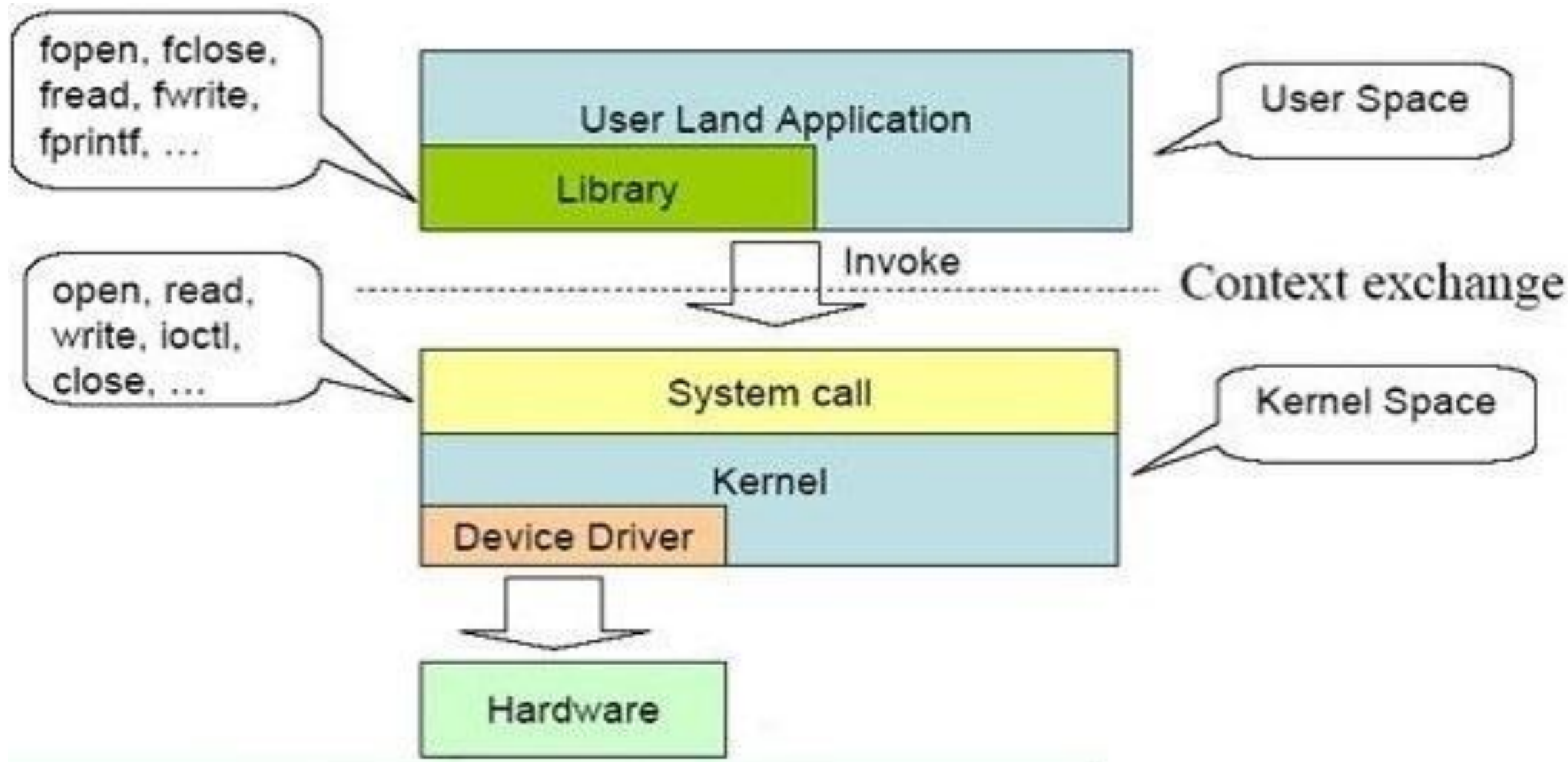
POSIX兼容的程序可在其它POSIX操作系统编译执行

◆ 运行程序格式 :

Windows: EXE, .dll(dynamic link library) , .lib

Linux: ELF, .so(shared object), .a

Library function和system call



◆ 用户空间：User space

用户程序的运行空间。为了安全，它们是隔离的，即使用户的程序崩溃，内核也不受影响

只能执行简单的运算，不能直接调用系统资源，必须通过系统接口（system call），才能向内核发出指令

◆ 内核空间：Kernel space

是 Linux 内核的运行空间

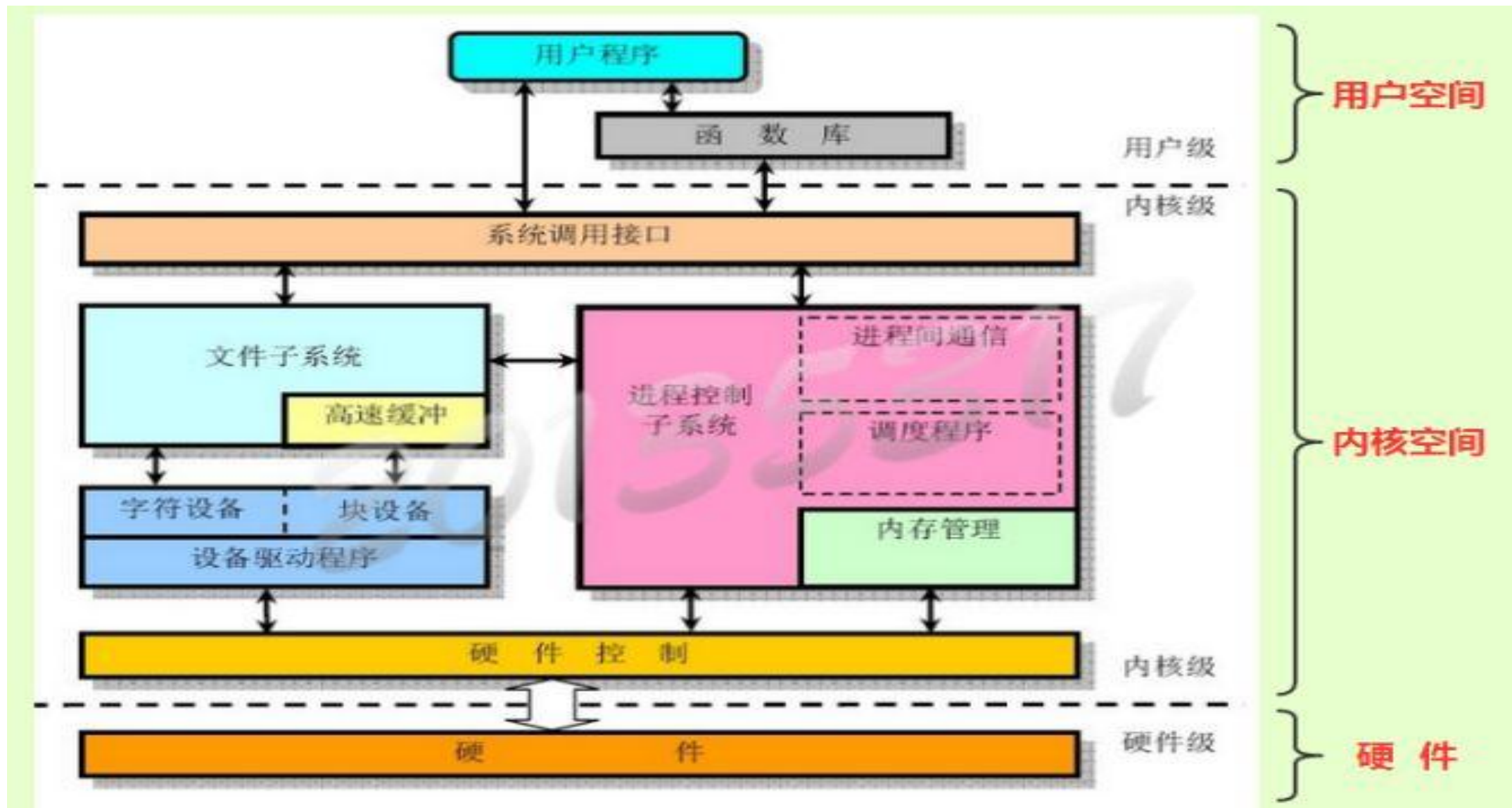
可以执行任意命令，调用系统的一切资源

◆ 示例：

```
str = "www.magedu.com"    // 用户空间
x = x + 100                // 用户空间
file.write(str)            // 切换到内核空间
y = x + 200                // 切换回用户空间
```

◆ 第一行和第二行都是简单的赋值运算，在 User space 执行。第三行需要写入文件，就要切换到 Kernel space，因为用户不能直接写文件，必须通过内核安排。第四行又是赋值运算，就切换回 User space

用户和内核空间



◆ 低级语言

机器语言：0和1

汇编语言：和机器语言一一对应，与硬件相关的特有代码、驱动程序开发

◆ 中级语言：C

系统级应用、驱动程序

◆ 高级语言:python , go , java , Objective-C , C# , php

应用级程序开发

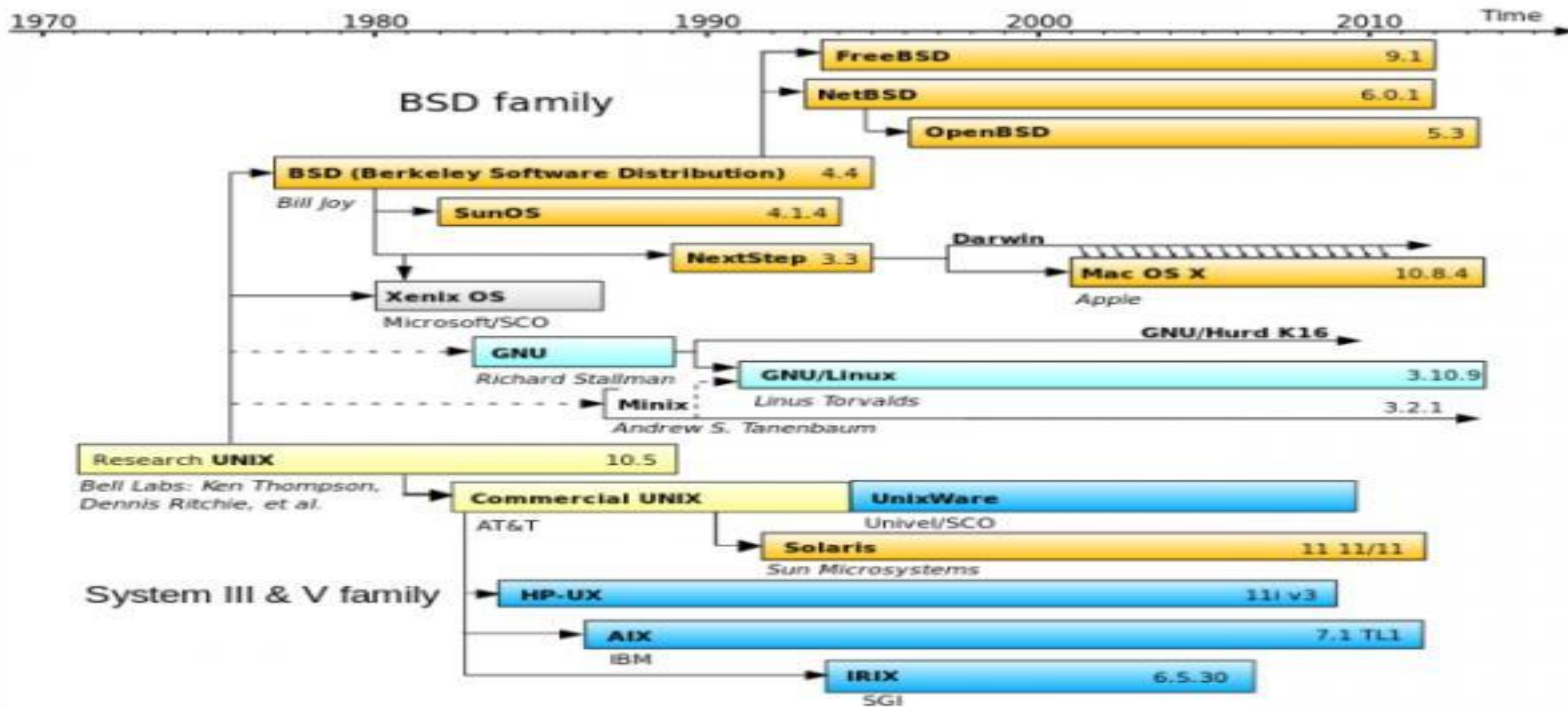
服务器三大操作系统

- ◆ Windows :
- ◆ Linux : GNU/Linux
- ◆ Unix : 1969年Ken Thompson
System: Bell Lab
 - AIX (IBM)
 - Solaris (SUN)
 - HP-UX (HP)
- BSD: (BSRG) Berkeley System Distribution
 - NetBSD
 - OpenBSD
 - FreeBSD

Linux已成趋势



UNIX历史



- ◆ 为了进一步强化大型主机的功能，让主机的资源可以提供更多的使用者来利用，在1964年，由贝尔实验室(Bell)、麻省理工学院(MIT)及奇异公司(GE美国通用电气公司)共同发起了Multics（多路信息计算系统）的计划，Multics计划的目的是让大型主机可以同时支持300个以上的终端机连线使用。不过，到了1969年前后，由于计划进度缓慢，资金也短缺，所以该计划虽然继续在研究，但最终贝尔实验室还是选择了退出

Ken Thompson:一切为了游戏



马哥教育

IT 人的高薪职业学院

- ◆ 1966年从加州大学伯克利分校毕业的Ken Thompson加入了贝尔实验室。参与了Multics系统的研发。他基于Multics开发了“star travel”游戏。不幸的是，1969年由于贝尔实验室退出Multics项目，这同时意味着Ken将没有机器可以再玩这个游戏了。面对此情此景，Ken作为一个创造者的本性立即体现了出来，于是他决定自己写一个操作系统来满足他玩游戏的需要，Ken找到了一台废弃已久的老式PDP-7，并在这台机器上重写了他的游戏。在这个过程中，Ken有了一个主意，要开发一个全新的操作系统。利用PDP-7上的汇编语言，Ken只花了一个月就编写完了操作系统的内核，在这个一个月中，他一周一内核，一个文件系统，一个编辑器和一个编译程序的完成

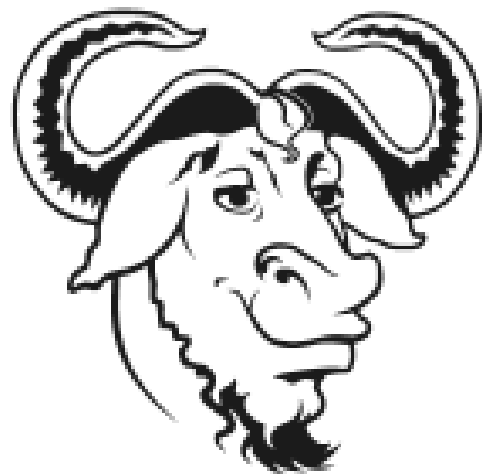
Unix历史

- ◆ 1965前后 Multics
贝尔实验室,麻省理工,通用电气发起
- ◆ 1969 Unix雏形UNICS
Ken Thompson 汇编语言 B语言
- ◆ 1973 正式命名为Unix
Dennis Ritchie C语言
- ◆ 1977 BSD
Berkeley Software Distribution 伯克利大学
- ◆ 1979 System V架构 版权声明
HP-UNIX IBM AIX “不对学生提供源码” 收回版权
- ◆ 1984年 Minix操作系统
Andrew S. Tanenbaum荷兰阿姆斯特丹自由大学计算机科学系



GNU

- ◆ GNU (GNU is Not Unix)
- ◆ 1984年由Richard Stallman发起并创建
- ◆ 目标是编写大量兼容于Unix系统的自由软件
- ◆ 官方网站：<http://www.gnu.org>
- ◆ GPL (GNU General Public License)
- ◆ 自由软件基金会：Free Software Foundation
- ◆ 允许用户任意复制、传递、修改及再发布
- ◆ 基于自由软件修改再次发布的软件，仍需遵守GPL
- ◆ LGPL (Lesser General Public License)
- ◆ LGPL相对于GPL较为宽松，允许不公开全部源代码
- ◆ GNU操作系统：Hurd Hird of Unix-Replacing Daemons



Linux起源



马哥教育

IT 人的高薪职业学院

- ◆ 1991年的10月5日，Torvalds在comp.os.minix新闻组上发布消息，正式向外宣布他自行编写的完全自由免费的内核诞生
(Freeminix-like kernel sources for 386-AT) ——FREAX
英文含义是怪诞的、怪物、异想天开
- ◆ 类Unix的内核，在GPL下发布
- ◆ 官网：www.kernel.org
- ◆ Linux操作系统：
完整的类UNIX操作系统
Linux内核+ GNU工具
如：CentOS,Ubuntu,Android



Linux团队



- ◆ slackware:SUSE Linux Enterprise Server (SLES)
OpenSuse桌面
- ◆ debian: ubuntu , mint
- ◆ redhat : RHEL: RedHat Enterprise Linux
每18个月发行一个新版本
CentOS : 兼容RHEL的格式
中标麒麟 : 中标软件
Fedora : 每6个月发行一个新版本
- ◆ ArchLinux : 轻量简洁
- ◆ Gentoo : 极致性能 , 不提供传统意义的安装程序
- ◆ LFS: Linux From scratch 自制Linux
- ◆ Android: kernel+busybox (工具集) +java虚拟机
- ◆ Linux分支参考网站 : <http://futurist.se/gldt/>

REDHAT红帽公司



马哥教育

IT 人的高薪职业学院

- ◆ Marc Ewing早在卡内基梅隆大学读书的时候就戴着他祖父的红色曲球帽. Marc在Marc Ewing大学常把自己的软件命名为Red Hat — Red hat 1, Red Hat 2 用来区别他人的软件. 所以当他开始自己的Linux项目, 自然就会命名为Red Hat Linux. Ewing非常乐于帮助同学解决计算机问题 “你的朋友可能会说, 去跟我的朋友Marc交流. 谁是Marc? 他是红帽男孩.” 所以, 至少在他们最亲密的朋友当中, 红帽子成为了技术专家的同义词
- ◆ 1993年, Bob Young 建立了ACC公司, 营销Linux和UNIX的支持软件和书籍杂志。
- ◆ 1995年, Bob Young 收购了Marc Ewing的业务, 合并后的ACC公司成为新的Red Hat软件公司, 发布了Red Hat Linux 2.0



Linux版本

Linux的内核版本由3部分组成

- ◆主版本号
- ◆次版本号
- ◆末版本号
- ◆编译版本号
- ◆厂商版本



The Linux Kernel Archives

About

Contact us

FAQ

Releases

Signatures

Site news



Protocol
HTTP
GIT
RSYNC

Location
<https://www.kernel.org/pub/>
<https://git.kernel.org/>
<rsync://rsync.kernel.org/pub/>

Latest Stable Kernel:



4.13.11

Linux发行版本

Linux发行版的名称和版本号是由发行版的维护者决定的

◆RedHat Enterprise Linux 7.3

由Red Hat公司发布



◆SUSE Linux Enterprise Server 12 SP2

由Novell公司发布



◆Debian Linux 9

由Debian维护社区发布



◆Android 8.0

由Google公司发布



Red Hat Linux版本



马哥教育

IT 人的高薪职业学院

Red Hat Linux 1.0 (Mother's Day)		1994年11月03日
Red Hat Linux 2.0		1995年09月20日
Red Hat Linux 3.0 (Picasso)		1996年05月01日
Red Hat Linux 4.0 (Colgate)		1996年10月08日
Red Hat Linux 5.0 (Hurricane)		1997年12月01日
Red Hat Linux 6.0 (Hedwig)		1999年04月26日
Red Hat Linux 6.2 (Zoot)	Red Hat Linux 6.2E	2000年04月03日
Red Hat Linux 7.0 (Guinness)		2000年03月27日
Red Hat Linux 7.2 (Enigma)	Red Hat Enterprise Linux 2	2001年10月22日
Red Hat Linux 8.0 (Psyche)		2002年09月30日
Red Hat Linux 9.0 (Shrike)		2003年03月31日
Red Hat Enterprise Linux 3.0(Taroon)	Fedora 1 (Yarrow)	2003年10月22日
Red Hat Enterprise Linux 4.0		2005年02月15日
Red Hat Enterprise Linux 5.0(Tikanga)		2007年03月14日
Red Hat Enterprise Linux 6.0(Santiago)		2010年11月10日
Red Hat Enterprise Linux 7.0(Maipo)		2014年06月10日

开源 Open Source



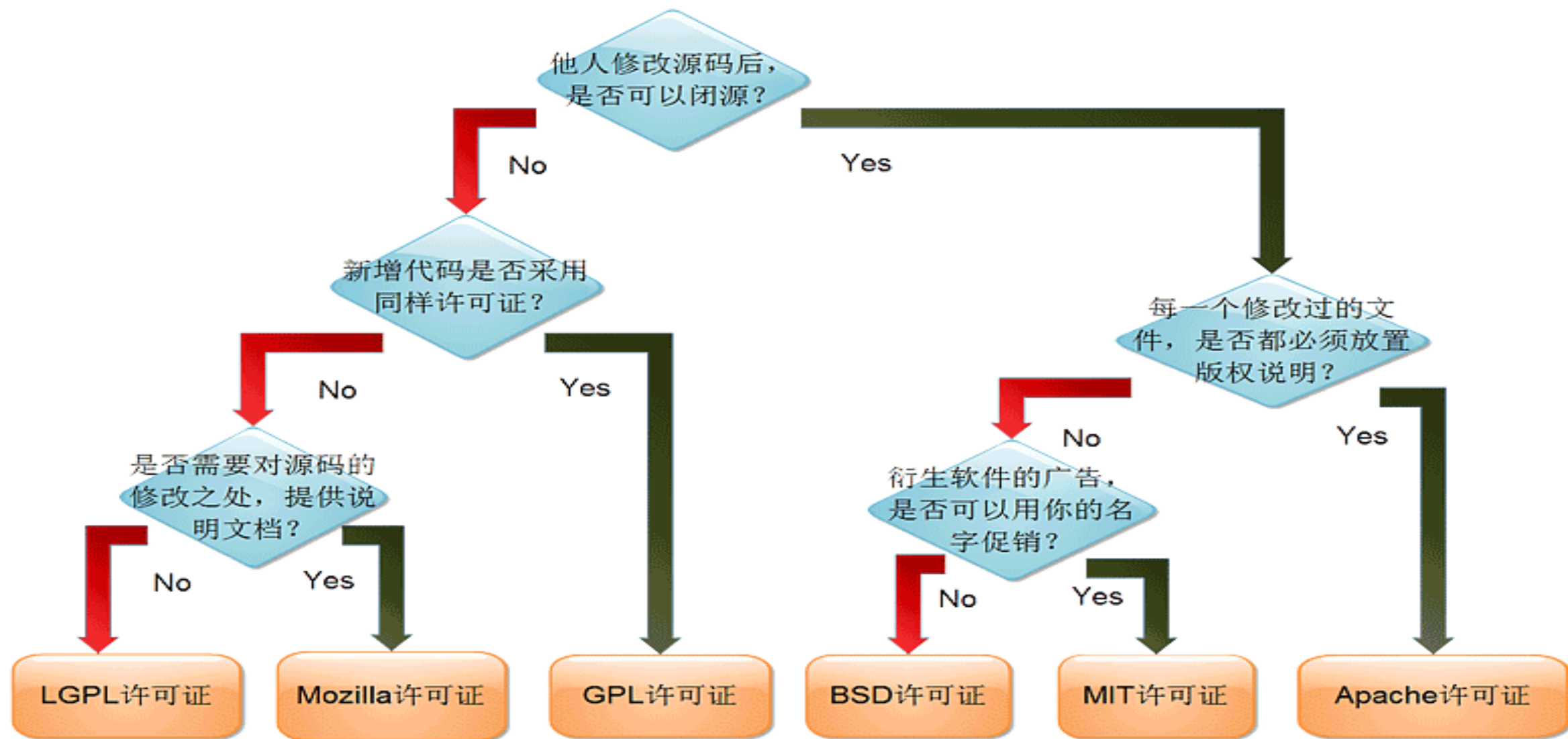
马哥教育

IT 人的高薪职业学院

- ◆ 开源：Open Source，软件和源代码提供给所有人
 - 自由分发软件和源代码
 - 能够修改和创建衍生作品
- ◆ 软件分类：
 - 商业
 - 共享
 - 自由 free

◆ 世界上的开源许可证，大概有上百种

- GPLv2, GPLv3, LGPL(lesser) : 通用公共许可 copyleft
- Apache: apache
- BSD: bsd
- Mozilla
- Apache
- MIT



Linux哲学思想



马哥教育

IT 人的高薪职业学院

- ◆ 一切都是一个文件（包括硬件）
- ◆ 小型，单一用途的程序
- ◆ 链接程序，共同完成复杂的任务
- ◆ 避免令人困惑的用户界面
- ◆ 配置数据存储在文本中

获取发行版



马哥教育
IT 人的高薪职业学院

◆ CentOS

<https://wiki.centos.org/Download>

<http://mirrors.aliyun.com>

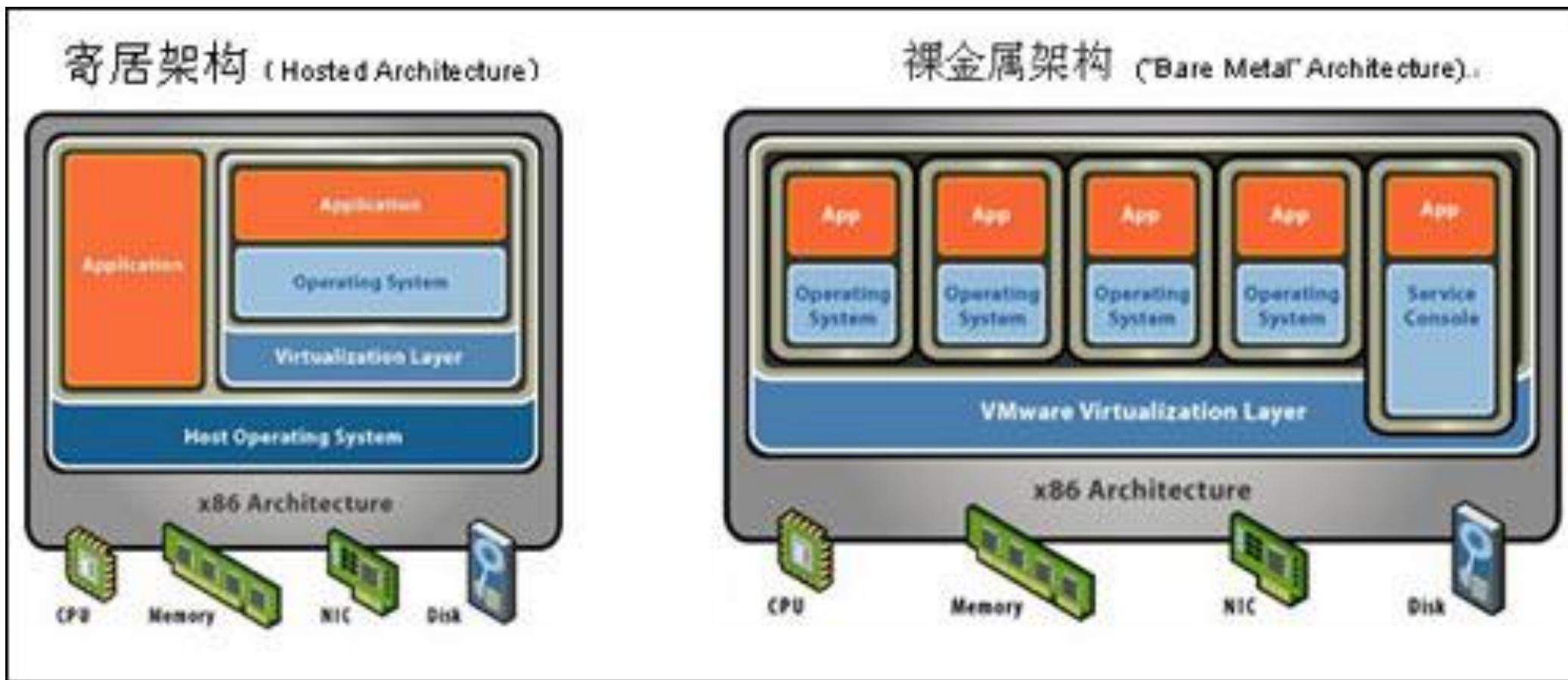
<http://mirrors.sohu.com>

<http://mirrors.163.com>

虚拟机



- ◆ 虚拟机：用软件（如：vmware,virtualbox等）模拟硬件



使用vmware workstation



- ◆ CPU：默认
- ◆ 内存：1G以上，推荐2G
- ◆ 硬盘：一块硬盘，200G
- ◆ 网卡：桥接或仅主机模式
- ◆ 光盘：挂载对应版本的ISO文件

◆ root 用户

一个特殊的管理帐户
也被称为超级用户
root 已接近完整的系统控制
对系统损害几乎有无限的能力
除非必要, 不要登录为 root

◆ 普通 (非特权) 用户

权限有限
造成损害的能力比较有限

终端terminal



- ◆ 设备终端

 - 键盘鼠标显示器

- ◆ 物理终端 (`/dev/console`)

 - 控制台console

- ◆ 虚拟终端(tty : teletypewriters , `/dev/tty#` #为[1-6])

 - tty 可有n个 , `Ctrl+Alt+F[1-6]`

- ◆ 图形终端 (`/dev/tty7`) startx, xwindows

 - CentOS 6: `Ctrl + Alt + F7`

 - CentOS 7: 在哪个终端启动 , 即位于哪个虚拟终端

- ◆ 串行终端 (`/dev/ttyS#`)

 - ttyS

- ◆ 伪终端 (`pty : pseudo-tty` , `/dev/pts/#`)

 - pty, SSH远程连接

- ◆ 查看当前的终端设备 : `#tty`

- ◆ 交互式接口：启动终端后，在终端设备附加一个交互式应用程序

- ◆ GUI : Graphic User Interface

 - X protocol, window manager, desktop

 - Desktop:

 - GNOME (C, 图形库gtk) ,

 - KDE (C++, 图形库qt)

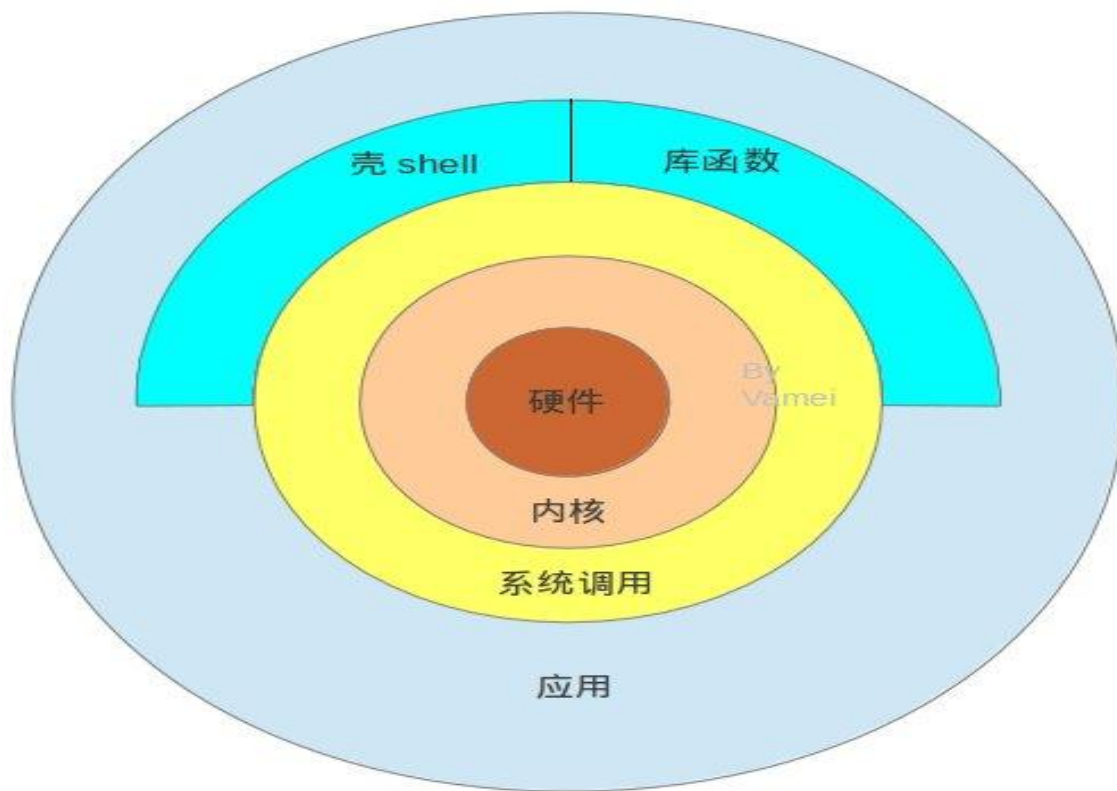
 - XFCE (轻量级桌面)

- ◆ CLI : Command Line Interface

 - shell程序：sh (bourn 史蒂夫·伯恩) csh tcsk ksh (korn) bash (bourn again shell)GPL zsh

什么是shell

- ◆ Shell 是Linux系统的用户界面，提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行
- ◆ shell也被称为Linux的命令解释器 (command interpreter)
- ◆ shell是一种高级程序设计语言



- ◆ GNU Bourne-Again Shell(bash)是GNU计划中重要的工具软件之一，目前也是 Linux标准的shell，与sh兼容
- ◆ CentOS默认使用
- ◆ 显示当前使用的shell
`echo ${SHELL}`
- ◆ 显示当前系统使用的所有shell
`cat /etc/shells`

命令提示符



◆ 命令提示符：prompt

```
[root@localhost ~]#
```

管理员 \$ 普通用户

◆ 显示提示符格式

```
[root@localhost ~]#echo $PS1
```

◆ 修改提示符格式

```
PS1="\[\e[1;5;41;33m\][\u@\h \W]\\$\[\e[0m\]"
```

\e \033 \u 当前用户

\h 主机名简称 \H 主机名

\w 当前工作目录 \W 当前工作目录基名

\t 24小时时间格式 \T 12小时时间格式

\! 命令历史数 \# 开机后命令历史数

◆ 输入命令后回车

提请shell程序找到键入命令所对应的可执行程序或代码，并由其分析后提交给内核分配资源将其运行起来

◆ 在shell中可执行的命令有两类

内部命令：由shell自带的，而且通过某命令形式提供

help 内部命令列表

enable cmd 启用内部命令

enable -n cmd 禁用内部命令

enable -n 查看所有禁用的内部命令

外部命令：在文件系统路径下有对应的可执行程序文件

查看路径：which -a |--skip-alias ; whereis

◆ 区别指定的命令是内部或外部命令

type COMMAND

◆ Hash缓存表

系统初始hash表为空，当外部命令执行时，默认会从PATH路径下寻找该命令，找到后会在这条命令的路径记录到hash表中，当再次使用该命令时，shell解释器首先会查看hash表，存在将执行之，如果不存在，将会去PATH路径下寻找。利用hash缓存表可大大提高命令的调用速率

◆ hash常见用法

hash 显示hash缓存

hash -l 显示hash缓存，可作为输入使用

hash -p path name 将命令全路径path起别名为name

hash -t name 打印缓存中name的路径

hash -d name 清除name缓存

hash -r 清除缓存

- ◆ 显示当前shell进程所有可用的命令别名

`alias`

- ◆ 定义别名NAME，其相当于执行命令VALUE

`alias NAME='VALUE'`

- ◆ 在命令行中定义的别名，仅对当前shell进程有效

- ◆ 如果想永久有效，要定义在配置文件中

仅对当前用户：`~/.bashrc`

对所有用户有效：`/etc/bashrc`

命令别名



- ◆ 编辑配置给出的新配置不会立即生效

- ◆ bash进程重新读取配置文件

```
source /path/to/config_file
```

```
./path/to/config_file
```

- ◆ 撤消别名：unalias

```
unalias [-a] name [name ...]
```

-a 取消所有别名

- ◆ 如果别名同原命令同名，如果要执行原命令，可使用

```
\ALIASNAME
```

```
"ALIASNAME"
```

```
' ALIASNAME'
```

```
command ALIASNAME
```

```
/path/commmand
```


◆ COMMAND [OPTIONS...] [ARGUMENTS...]

选项：用于启用或关闭命令的某个或某些功能

短选项：-c 例如：-l, -h

长选项：--word 例如：--all, --human-readable

参数：命令的作用对象，比如文件名，用户名等

◆ 注意：

- 多个选项以及多参数和命令之间使用空白字符分隔
- 取消和结束命令执行：Ctrl+c , Ctrl+d
- 多个命令可以用;符号分开
- 一个命令可以用\分成多行

◆ Linux的两种时钟

系统时钟：由Linux内核通过CPU的工作频率进行的

硬件时钟：主板

◆ 相关命令

date 显示和设置系统时间

date +%s

date -d @1509536033

hwclock , clock: 显示硬件时钟

-s, --hctosys 以硬件时钟为准，校正系统时钟

-w, --systohc 以系统时钟为准，校正硬件时钟

◆ 时区：/etc/localtime

◆ 显示日历：cal -y

简单命令



◆ 关机 : halt, poweroff

◆ 重启 : reboot

-f: 强制, 不调用shutdown

-p: 切断电源

◆ 关机或重启 : shutdown

shutdown [OPTION]... TIME [MESSAGE]

-r: reboot

-h: halt

-c : cancel

TIME:无指定, 默认相当于 +1

now: 立刻,相当于+0

+m: 相对时间表示法, 几分钟之后; 例如 +3

hh:mm: 绝对时间表示, 指明具体时间

◆ 用户登录信息查看命令：

whoami: 显示当前登录有效用户

who: 系统当前所有的登录会话

w: 系统当前所有的登录会话及所做的操作

◆ nano 文本编辑

◆ screen命令：

创建新screen会话

```
screen -S [SESSION]
```

加入screen会话

```
screen -x [SESSION]
```

退出并关闭screen会话

```
exit
```

剥离当前screen会话

```
Ctrl+a,d
```

显示所有已经打开的screen会话

```
screen -ls
```

恢复某screen会话

```
screen -r [SESSION]
```

- ◆ echo命令
- ◆ 功能：显示字符
- ◆ 语法：echo [-neE][字符串]
- ◆ 说明：echo会将输入的字符串送往标准输出。输出的字符串间以空白字符隔开, 并在最后加上换行号
- ◆ 选项：
 - -E （默认）不支持 \ 解释功能
 - -n 不自动换行
 - -e 启用 \ 字符的解释功能
- ◆ 显示变量
 - echo "\$VAR_NAME " 变量会替换，弱引用
 - echo '\$VAR_NAME ' 变量不会替换，强引用

- ◆ 启用命令选项-e，若字符串中出现以下字符，则特别加以处理，而不会将它当成一般文字输出

\a 发出警告声

\b 退格键

\c 最后不加上换行符号

\n 换行且光标移至行首

\r 回车，即光标移至行首，但不换行

\t 插入tab

\\ 插入\字符

\0nnn 插入nnn（八进制）所代表的ASCII字符

```
echo -e '\033[43;31;5mmagedu\033[0m'
```

\xHH 插入HH（十六进制）所代表的ASCII字

ASCII 字符代码表 一

高四位 低四位		ASCII非打印控制字符										ASCII 打印字符													
		0000					0001					0010		0011		0100		0101		0110		0111			
		0					1					2		3		4		5		6		7			
		十进制	字符	ctrl	代码	字符解释	十进制	字符	ctrl	代码	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	ctrl	
0000	0	0	BLANK NULL	^@	NUL	空	16	▶	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	`	112	p		
0001	1	1	☺	^A	SOH	头标开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q		
0010	2	2	☺	^B	STX	正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r		
0011	3	3	♥	^C	ETX	正文结束	19	!!	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s		
0100	4	4	◆	^D	EOT	传输结束	20	¶	^T	DC4	设备控制 4	36	\$	52	4	68	D	84	T	100	d	116	t		
0101	5	5	♣	^E	ENQ	查询	21	♫	^U	NAK	反确认	37	%	53	5	69	E	85	U	101	e	117	u		
0110	6	6	♠	^F	ACK	确认	22	■	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v		
0111	7	7	●	^G	BEL	震铃	23	↑↓	^W	ETB	传输块结束	39	'	55	7	71	G	87	w	103	g	119	w		
1000	8	8	◼	^H	BS	退格	24	↑	^X	CAN	取消	40	(56	8	72	H	88	X	104	h	120	x		
1001	9	9	○	^I	TAB	水平制表符	25	↓	^Y	EM	媒体结束	41)	57	9	73	I	89	Y	105	i	121	y		
1010	A	10	◉	^J	LF	换行/新行	26	→	^Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z		
1011	B	11	♂	^K	VT	竖直制表符	27	←	^[ESC	转意	43	+	59	;	75	K	91	[107	k	123	{		
1100	C	12	♀	^L	FF	换页/新页	28	└	^\	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124			
1101	D	13	🎵	^M	CR	回车	29	↔	^]	GS	组分分隔符	45	-	61	=	77	M	93]	109	m	125	}		
1110	E	14	🎵	^N	SO	移出	30	▲	^_	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~		
1111		15	◼	^O	SI	移入	31	▼	^-	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	Δ	Back space	

注：表中的ASCII字符可以用:ALT + “小键盘上的数字键” 输入

命令行扩展、被括起来的集合

◆ 命令行扩展：\$() 或 ``

把一个命令的输出打印给另一个命令的参数

```
echo "This system's name is $(hostname) "
```

```
This system's name is server1.example.com
```

```
echo "i am `whoami` "
```

```
i am root
```

◆ 括号扩展：{ }

打印重复字符串的简化形式

```
echo file{1,3,5}
```

```
file1 file3 file5
```

```
rm -f file{1,3,5}
```

```
echo {1..10}
```

```
echo {a..z}
```

```
echo {000..20..2}
```

◆ 命令补全

内部命令：

外部命令：bash根据PATH环境变量定义的路径，自左而右在每个路径搜寻以给定命令命名的文件，第一次找到的命令即为要执行的命令

用户给定的字符串只有一条唯一对应的命令，直接补全

否则，再次Tab会给出列表

◆ 路径补全

把用户给出的字符串当做路径开头，并在其指定上级目录下搜索以指定的字符串开头的文件名

如果惟一：则直接补全

否则：再次Tab给出列表

- ◆ 保存你输入的命令历史。可以用它来重复执行命令
- ◆ 登录shell时，会读取命令历史文件中记录下的命令 `~/.bash_history`
- ◆ 登录进shell后新执行的命令只会记录在缓存中；这些命令会用户退出时“追加”至命令历史文件中

- ◆ 重复前一个命令，有4种方法
 - 重复前一个命令使用上方向键，并回车执行
 - 按 !! 并回车执行
 - 输入 !-1 并回车执行
 - 按 Ctrl+p 并回车执行
- ◆ !:0 执行前一条命令（去除参数）
- ◆ !n 执行history命令输出对应序号n的命令
- ◆ !-n 执行history历史中倒数第n个命令

- ◆ 使用up（向上）和down（向下）键来上下浏览从前输入的命令
- ◆ ctrl-r来在命令历史中搜索命令
 - （reverse-i-search）''：
- ◆ Ctrl+g：从历史搜索模式退出
- ◆ 要重新调用前一个命令中最后一个参数：
 - !\$ 表示
 - Esc, .（点击Esc键后松开，然后点击.键）
 - Alt+.（按住Alt键的同时点击.键）

命令history



- ◆ history [-c] [-d offset] [n]
- ◆ history -anrw [filename]
- ◆ history -ps arg [arg...]
 - c: 清空命令历史
 - d offset: 删除历史中指定的第offset个命令
 - n: 显示最近的n条历史
 - a: 追加本次会话新执行的命令历史列表至历史文件
 - r: 读历史文件附加到历史列表
 - w: 保存历史列表到指定的历史文件
 - n: 读历史文件中未读过的行到历史列表
 - p: 展开历史参数成多行，但不存在历史列表中
 - s: 展开历史参数成一行，附加在历史列表后

获得帮助



马哥教育

IT 人的高薪职业学院

◆ 获取帮助的能力决定了技术的能力！

◆ 多层次的帮助

`whatis`

`command --help`

`man` and `info`

`/usr/share/doc/`

Red Hat documentation

其它网站和搜索

whatis



- ◆ 显示命令的简短描述
- ◆ 使用数据库
- ◆ 刚安装后不可立即使用
- ◆ makewhatis | mandb制作数据库
- ◆ 使用示例：
whatis cal 或 man -f cal

- ◆ 内部命令：
 help COMMAND
 man bash
- ◆ 外部命令：
 - (1) COMMAND --help
 - COMMAND -h
 - (2) 使用手册(manual)
 man COMMAND
 - (3) 信息页
 info COMMAND
 - (4) 程序自身的帮助文档
 README
 INSTALL
 ChangeLog
 - (5) 程序官方文档
 官方网站：Documentation
 - (6) 发行版的官方文档
 - (7) Google

--help选项



- ◆ 显示用法总结和参数列表
- ◆ 使用的大多数，但并非所有的
- ◆ 示例：

date --help

Usage: date [OPTION]... [+FORMAT] or: date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

[] 表示可选项

CAPS或 <> 表示变化的数据

... 表示一个列表

x |y| z 的意思是 “ x 或 y 或 z ”

-abc的 意思是-a -b -c

{ } 表示分组

- ◆ 提供命令帮助的文件
- ◆ 手册页存放在/usr/share/man
- ◆ 几乎每个命令都有man的“页面”
- ◆ man页面分组为不同的“章节”
- ◆ 统称为Linux手册
- ◆ man命令的配置文件：/etc/man.config | man_db.conf
 - MANPATH /PATH/TO/SOMEWHERE: 指明man文件搜索位置
- ◆ man -M /PATH/TO/SOMEWHERE COMMAND: 到指定位置下搜索COMMAND命令的手册页并显示
- ◆ 中文man需安装包man-pages-zh-CN

man 章节



- ◆ 1: 用户命令
- ◆ 2: 系统调用
- ◆ 3: C库调用
- ◆ 4: 设备文件及特殊文件
- ◆ 5: 配置文件格式
- ◆ 6: 游戏
- ◆ 7: 杂项
- ◆ 8: 管理类的命令
- ◆ 9 : Linux 内核API

man 帮助段落说明



◆ 帮助手册中的段落说明：

- NAME 名称及简要说明
- SYNOPSIS 用法格式说明
 - [] 可选内容
 - <> 必选内容
 - a|b 二选一
 - {} 分组
 - ... 同一内容可出现多次
- DESCRIPTION 详细说明
- OPTIONS 选项说明
- EXAMPLES 示例
- FILES 相关文件
- AUTHOR 作者
- COPYRIGHT 版本信息
- REPORTING BUGS bug信息
- SEE ALSO 其它帮助参考

- ◆ 查看man手册页

man [章节] keyword

- ◆ 列出所有帮助

man -a keyword

- ◆ 搜索man手册

man -k keyword 列出所有匹配的页面
使用 whatis 数据库

- ◆ 相当于whatis

man -f keyword

- ◆ 打印man帮助文件的路径

man -w [章节] keyword

◆ man命令的操作方法：使用less命令实现

space, ^v, ^f, ^F: 向文件尾翻页

b, ^b: 向文件首部翻页

d, ^d: 向文件尾部翻半屏

u, ^u: 向文件首部翻半屏

RETURN, ^N, e, ^E or j or ^J: 向文件尾部翻一行
or ^K: 向文件首部翻一行

q: 退出

: 跳转至第#行

1G: 回到文件首部

G : 翻至文件尾部

◆ /KEYWORD:

以KEYWORD指定的字符串为关键字，从当前位置向文件尾部搜索；不区分字符大小写；

n: 下一个

N : 上一个

◆ ?KEYWORD:

以KEYWORD指定的字符串为关键字，从当前位置向文件首部搜索；不区分字符大小写；

n: 跟搜索命令同方向，下一个

N : 跟搜索命令反方向，上一个

- ◆ man常用于命令参考，GNU工具info适合通用文档参考。
- ◆ 没有参数,列出所有的页面
- ◆ info 页面的结构就像一个网站
- ◆ 每一页分为“节点”
- ◆ 链接节点之前 *
- ◆ info [命令]

- ◆ 方向键，PgUp，PgDn 导航
- ◆ Tab键 移动到下一个链接
- ◆ d 显示主题目录
- ◆ Home 显示主题首部
- ◆ Enter进入 选定链接
- ◆ n/p/u/l 进入下/前/上一层/最后一个链接
- ◆ s 文字 文本搜索
- ◆ q 退出 info

- ◆ System->help (centos6)
- ◆ Applications -> documentation->help (centos7)
 - 提供的官方使用指南和发行注记
- ◆ /usr/share/doc目录
 - 多数安装了的软件包的子目录,包括了这些软件的相关原理说明
 - 常见文档：README INSTALL CHANGES
 - 不适合其它地方的文档的位置
 - 配置文件范例
 - HTML/PDF/PS 格式的文档
 - 授权书详情

◆ 第三方应用官方文档

- <http://www.nginx.org>
- <http://tomcat.apache.org>
- <http://httpd.apache.org>
- <http://www.python.org>

◆ 通过发行版官方的文档光盘或网站可以获得

- 安装指南、部署指南、虚拟化指南等
- 红帽知识库和官方在线文档

<http://kbase.redhat.com>

<http://www.redhat.com/docs>

<http://access.redhat.com>

◆ rhn.redhat.com或者本地卫星服务器/代理服务器

- 基于网络管理的服务器
- RHN代理服务器为RHN流量提供缓存
- RHN卫星服务器提供自主的RHN

◆ RHN账户

- 为及其注册和基于网络管理的RHN用户
- 用于系统自动验证的系统ID

◆ sosreport

收集所有系统上的日志信息的工具，并自动打成压缩包，方便技术支持人员和红帽全球支持提供分析问题依据

网站和搜索



马哥教育

IT 人的高薪职业学院

- ◆ <http://tldp.org>
- ◆ <http://www.slideshare.net>
- ◆ <http://www.google.com>
Openstack filetype:pdf
rhca site:redhat.com/docs



马哥教育
IT 人的高薪职业学院

祝大家学业有成

谢 谢

咨询热线 400-080-6560