

## Linux 云计算集群架构师

**学神 IT 教育：从零基础到实战，从入门到精通！**

### 版权声明：

本系列文档为《学神 IT 教育》内部使用教材和教案，只允许 VIP 学员个人使用，禁止私自传播。否则将取消其 VIP 资格，追究其法律责任，请知晓！

### 免责声明：

本课程设计目的只用于教学，切勿使用课程中的技术进行违法活动，学员利用课程中的技术进行违法活动，造成的后果与讲师本人及讲师所属机构无关。倡导维护网络安全人人有责，共同维护网络文明和谐。

### 联系方式：

学神 IT 教育官方网站: <http://www.xuegod.cn>

Linux 云计算架构师进阶学习群 QQ 群: 1072932914



学习顾问：小语老师

学习顾问：边边老师

学神微信公众号

微信扫码添加学习顾问微信，同时扫码关注学神公众号了解最新动态，获取更多学习资料及答疑就业服务！

## 第二章 Linux 基本命令操作

### 本节所讲内容:

- 2.1 Linux 终端介绍 Shell 提示符 Bash Shell 基本语法
- 2.2 基本命令操作
- 2.3 系统时间管理
- 2.4 帮助命令使用
- 2.5 开关机命令及 7 个启动级别
- 2.6 实战: 设置服务器来电后自动开机
- 2.7 实战: 设置服务器定时开机

## 2.1 Linux 终端介绍、Shell 提示符、Bash 基本语法

### 2.1.1 登录 LINUX 终端

两种终端仿真器: 1、GNOME 桌面的 GNOME Terminal ; 2、KDE 桌面的 Konsole Terminal  
远程连接终端工具: Xshell , CRT 。后期会教你安装。

例 1: 通过 tty 命令看到当前所属的虚拟终端或叫伪终端

pts 指的是伪终端, 当打开一个终端时为 pts/0, 再次打开一个终端时即为 pts/1, 在

图形界面使用的就是伪终端, 使用 xshell 远程连接工具也是使用的伪终端。

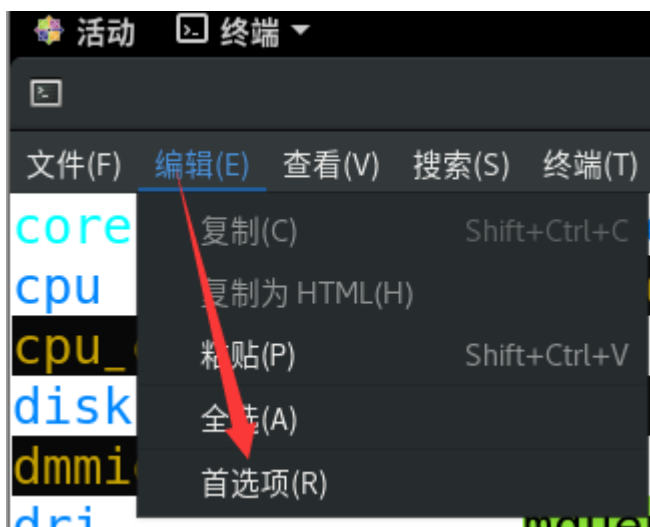
当我们进入第 3 种级别, 多用户级别的字符界面, 使用的是/dev/tty 就是控制终端。

```
[root@xuegod63 ~]# tty
```

```
/dev/pts/0
```

注: ctrl+shift+T 快速打开一个终端; shift+ctrl+加号 终端字体放大; 终端字体缩小: ctrl+减号

修改虚拟终端显示的颜色为: 白底黑字





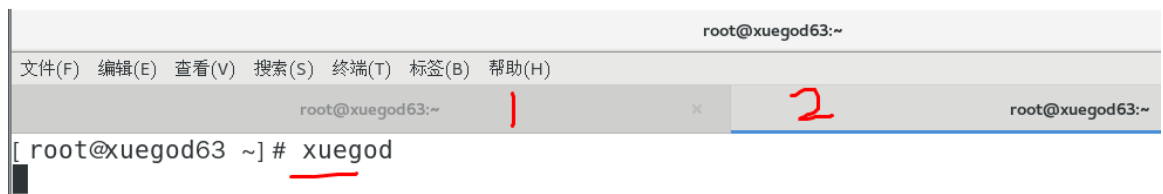
改完后，立即生效，没有保存按钮。直接点关闭就可以了。

## 例 2：不同虚拟终端之间通讯

同时打开两终端，第一个终端执行：

```
[root@xuegod63 ~]# echo xuegod > /dev/pts/1
```

在第二个终端查看：有 xuegod 输出



## 例 3：对所有终端广播消息：系统 10 分钟后关机。

[root@xuegod63 ~]# shutdown +10 #执行此命令后，在其他终端都会收到关机的消息。

centos8 当中本地桌面对广播消息不能很好的接收，xshell 远程登录可以接收到。

[root@xuegod63 ~]# shutdown -c #取消关机

或：

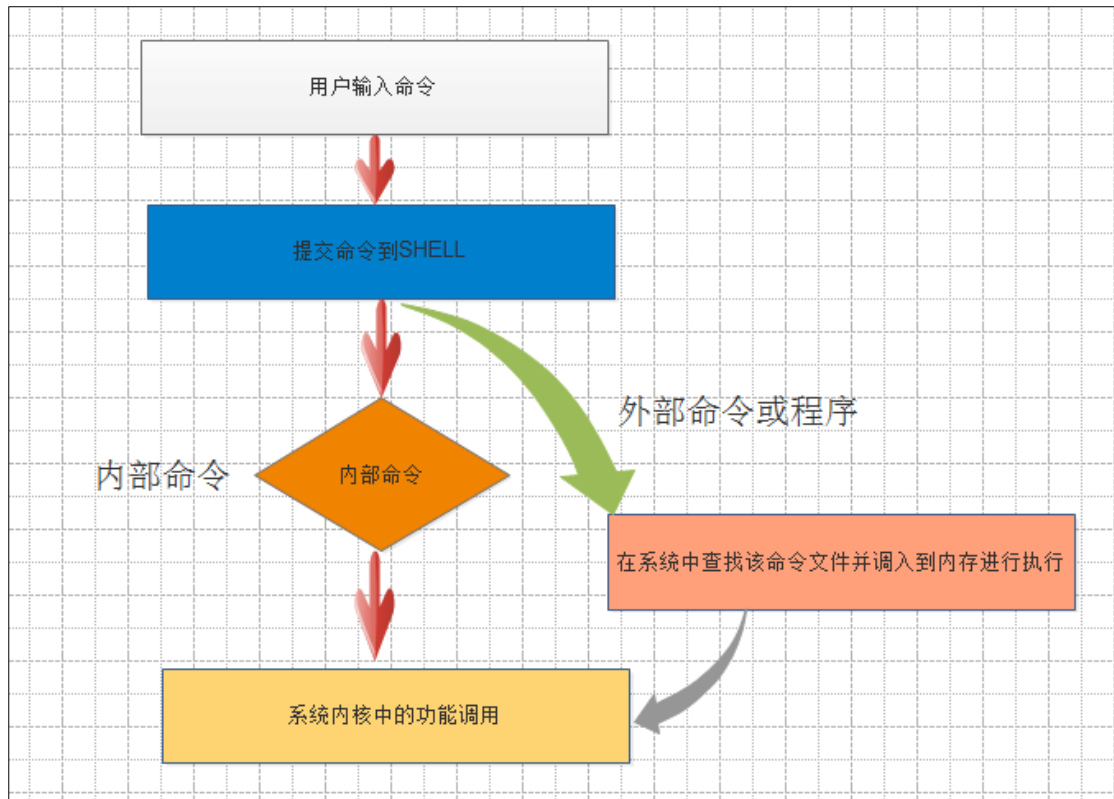
[root@xuegod63 ~]# wall " The system will be shut down in 10 minutes " #广播，所有终端都能收到

## 2.1.2 认识 SHELL

Shell 俗称壳，它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 实际上是一个命令解释器，它通过解释用户输入的命令并把它传输给系统内核去执行。

Shell 有自己的编程语言，它允许用户编写由 shell 命令组成的程序。Shell 编程语言具有普通编程语言的很多特点，比如它也有循环结构和分支控制结构等。



内部命令：在系统启动时就调入内存，是常驻内存的，所以执行效率高

外部命令：是系统软件的功能，用户需要时才从硬盘中读入内存

如何区分内外部命令？

使用 type 命令，语法：type 要检测的命令

```
[root@xuegod63 ~]# type cat
```

cat 是 /usr/bin/cat

```
[root@xuegod63 ~]# type pwd
```

pwd 是 shell 内建

总结：通过 SHELL，我们可以对 LINUX 实现哪些操作或管理：

例如：

对文件的管理（创建、删除、复制、修改）

- 1、对用户的管理（添加、删除）
- 2、相关权限的管理（授权用户对相关文件的管理，比如增删改查）
- 3、对磁盘的管理（分区、raid、lvm）
- 4、对软件的管理
- 5、对网络的管理

### 2.1.3 shell 提示符 “#” 号与 “\$” 号的区别

```
[root@xuegod63 ~]#           #表示是 root 用户登录，管理员账号登陆
```

```
[root@xuegod63 ~]# su - mk    #切换到 mk 普通用户
```

```
[mk@xuegod63 ~]$             #表示普通用户登录
```

上面各位置对应的内容代表的意思如下：

```
[root    @   xuegod63    ~                ]#
```

用户名---@---主机名---当前所在目录(~表示当前用户的家目录)---(# root/\$普通用户)

```
[user1@xuegod63 ~]$ exit #退出当前 shell
```

#### 2.1.4 认识 Bash Shell

查看所有 shell 类型

```
[root@xuegod63 ~]# cat /etc/shells
```

/bin/sh

/bin/bash

/usr/bin/sh

/usr/bin/bash

具体你使用的是哪一个, 取决于你的用户配置。

例: 通过查看/etc/passwd 文件的每一行的最后一个字段, 确认用户使用的 shell 类型

```
[root@xuegod63 ~]# head /etc/passwd #查看 passwd 文件中前 10 行内容
```

```
root:x:0:0:root:/root:/bin/bash
```

注: (第一阶段报错问题, 大概率是 2 点, 第一点检查命令是否字母空格敲错了, 2 检查当前所在路径是否正确。)

## 2.2 基本命令操作

在 LINUX 中使用一个命令, 命令格式如下:

命令 【选项】 【参数】

命令: 具体执行的命令, 比如 pwd,head

选项: 会影响到命令的一些参数

参数: 命令作用的对象

#### 2.2.1 基本命令之--ls

作用: 查看当前目录下有哪些文件 (list)

语法: ls 目录/文件, 如果什么也不加, 那么查看的是当前目录下的内容

常用选项:

命令后面不加任何选项

```
[root@xuegod63 ~]# ls
```

-l 列出文件的详细信息, 如创建者, 创建时间, 文件的读写权限列表等等, 长列表

```
[root@xuegod63 ~]# ls -l
```

总用量 8

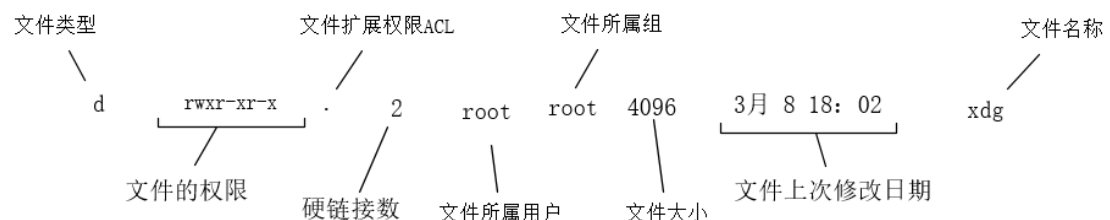
```
-rw-----. 1 root root 1680 9 月 19 12:16 anaconda-ks.cfg
```

```
-rw-r--r--. 1 root root 1728 9 月 19 12:55 initial-setup-ks.cfg
```

```
drwxr-xr-x. 2 root root 6 9 月 19 13:05 公共
```

```
drwxr-xr-x. 2 root root 6 9 月 19 13:05 模板
```

```
drwxr-xr-x. 2 root root 6 9 月 19 13:05 视频
```



第一个字符文件类型中:

d: 目录文件

l: 链接文件

b: 块设备文件

c: 字符设备文件

p: 管道文件

-: 表示普通文件

为什么有的目录文件有不同的颜色呢?

linux 系统中不同的颜色代表了不同的文件类型:

颜色	代表内容	举 例
蓝色	目录	/etc
黑色	文件	/etc/passwd
浅蓝色	链接	/etc/grub2.cfg
红色	压缩包	boot.tar.gz
绿色	可执行文件	/bin/bash
黑底黄字	设备文件	/dev/sda
绿底黑字	粘滞位权限目录文件	/tmp

粘滞位权限是针对目录的, 对文件无效, 也叫防删除位 (目录下创建的文件只有 root、文件创建者、目录所有者才能删除。)

```
[root@xuegod83 ~]# touch file1 file2
[root@xuegod83 ~]# tar czvf file.tar.gz file1 file2
[root@xuegod83 ~]# ls
```

例: 查看/dev 目录下的文件

```
[root@xuegod63 ~]# ls /dev/
```

```
[root@xuegod63 ~]# ls /dev/
autofs      kmsg        raw          tty14
block       log          rtc          tty15
bsg         loop-control rtc0         tty16
```

参数-a : 列出目录下所有的文件, 包括以 “.” 开头的隐藏文件 (linux 下隐藏文件是以 . 开头的, 如果存在 2 个点代表存在着父目录, 1 个点表示当前目录)

```
[root@xuegod63 ~]# ls -a
.          .bash_profile  .dbus          .mozilla  视频  桌面
```

-d 查看目录(不查看里面的内容)

```
[root@xuegod63 ~]# ls -ld /root/
```

```
dr-xr-x---. 15 root root 4096 11 月 15 21:37 /root/
```

-S 以文件的大小进行排序

```
[root@xuegod63 ~]# ls -ls /root/
```

总用量 8

```
-rw-r--r--. 1 root root 1728 9 月 19 12:55 initial-setup-ks.cfg
```

```
-rw-----. 1 root root 1680 9 月 19 12:16 anaconda-ks.cfg
```

ls -l 和 ll 这两个命令是等价的

```
[root@xuegod63 ~]# ll /etc/passwd
-rw-r--r--. 1 root root 2053 9 月 19 12:57 /etc/passwd
[root@xuegod63 ~]# ls -l /etc/passwd
-rw-r--r--. 1 root root 2053 9 月 19 12:57 /etc/passwd
[root@xuegod63 ~]# type ll
ll 是 `ls -l --color=auto' 的别名
```

### 2.2.2 别名的使用

定义一个别名:

```
[root@xuegod63 ~]# alias vimens33='vim /etc/sysconfig/network-scripts/ifcfg-ens160'
```

```
[root@xuegod63 ~]# vimens160
```

删除别名:

```
[root@xuegod63 ~]# unalias vimens160
```

设置别名永久生效:

对当前用户设置永久别名

```
[root@xuegod63 ~]# vim /root/.bashrc #在文件最后, 插入以下内容:
```

```
alias vimens160="vim /etc/sysconfig/network-scripts/ifcfg-ens160"
```

```
[root@xuegod63 ~]# source /root/.bashrc
```

```
[root@xuegod63 ~]# vimens160
```

对当全局用户设置永久别名, 就是让任何人, 都可以使用你创建的别名

```
[root@xuegod63 ~]# vim /etc/bashrc #在文件最后插入
```

```
alias vimens160="vim /etc/sysconfig/network-scripts/ifcfg-ens160"
```

```
[root@xuegod63 ~]# su - user1
```

上一次登录: 四 11 月 16 10:22:03 CST 2017pts/1 上

```
[user1@xuegod63 ~]$ vimens160
```

```
[user1@xuegod63 ~]$ exit
```

```
[root@xuegod83 ~]# type vimens160
```

### 2.2.3 基本命令之--cd

作用: 切换目录 (change directory)

语法: cd 目录

例 1: 直接输入 cd 回到当前用户的宿主 (家) 目录

```
[root@xuegod63 ~]# cd /etc/sysconfig/network-scripts/
```

```
[root@xuegod63 network-scripts]# cd #回到当前用户的宿主 (家) 目录
```

或:

```
[root@xuegod63 ~]# cd ~ #回到当前用户的宿主 (家) 目录
```

```
[root@xuegod63 ~]# pwd
```

```
/root
```

```
[root@xuegod63 ~]# cd .. #表示返回到上级目录位置, 也就是父目录
```

```
[root@xuegod63 /]# pwd
```



```
/
[root@xuegod63 /]# cd .    #表示进入到当前目录
[root@xuegod63 /]# cd /etc/sysconfig/network-scripts/
[root@xuegod63 network-scripts]# cd -    #表示返回切换前的目录
/
```

#### 2.2.4 历史命令之--history

命令: history

4 个快速查找 Linux 历史命令的技巧:

方法 1: 光标上下键

方法 2: **ctrl+r** -》输入某条命令的关键字-》找出来对应的命令

方法 3: **!数字** //执行历史命令中第 N 条命令

方法 4: **!字符串** //搜索历史命令中最近一个以 xxxx 字符开头的命令, 例如!**vim**

#### 3.2.5 Linux 下快捷键

都是用 Ctrl+下面的单词, ^表示 Ctrl

Ctrl+C #终止前台运行的程序, 如: ping g.cn 后, 想停止按下 Ctrl+C

Ctrl+D #退出 等价 exit

Ctrl+L #清屏与执行 clear 命令功能一样

Ctrl+R #搜索历史命令, 可以利用好关键词

!**\$** #引用上一个命令的最后一个参数

Esc + . #引用上一个命令的最后一个参数

```
[root@xuegod63 network-scripts]# cat /etc/hosts
```

```
[root@xuegod63 network-scripts]# vim !$    #相当于执行: vim /etc/hosts
```

补全命令使用 tab 键, Tab 只能补全命令和文件的路径和名字

## 2.3 系统时间管理

在 Linux 中有硬件时钟与系统时钟等两种时钟。硬件时钟是指主机板上的时钟设备, 也就是通常可在 BIOS 画面设定的时钟; 系统时钟则是指 kernel 中的时钟; 所有 Linux 相关指令与函数都是读取系统时钟的设定

当 Linux 启动时, 系统时钟会去读取硬件时钟的设定, 之后系统时钟即独立运作

#### 2.3.1 两种时间

查看硬件时间:

```
[root@xuegod63 etc]# hwclock
```

查看系统时间:

```
[root@xuegod63 mnt]# date
```

时区:

UTC:

整个地球分为二十四时区, 每个时区都有自己的本地时间, 在国际无线电通信场合, 为了统一起见, 使用一个统一的时间, 称为通用协调时(UTC:Universal Time Coordinated)。

GMT:

格林威治标准时间 (Greenwich Mean Time)指位于英国伦敦郊区的皇家格林尼治天文台的标准时间, 因为本初子午线被定义在通过那里的经线(UTC 与 GMT 时间基本相同)。



CST:

中国标准时间 (China Standard Time)

GMT + 8 = UTC + 8 = CST

DST:

夏令时(Daylight Saving Time) 指在夏天太阳升起的比较早时, 将时间拨快一小时, 以提早日光的使用 (中国不使用)。

UTC (Universal Time Coordinated): 世界标准时间

GMT (Greenwich Mean Time): 格林尼治时间

CST (China standard Time): 中国标准时间

DST: (Daylight Saving Time): 夏令时

EDT: (Eastern Daylight Time): 北美东部夏令时间

选择时区的命令:

```
[root@xuegod63 ~]# timedatectl
[root@xuegod63 ~]# timedatectl list-timezones | grep "New_York"
[root@xuegod63 ~]# timedatectl list-timezones | grep "Shanghai"
[root@xuegod63 ~]# timedatectl set-timezone America/New_York
[root@xuegod63 ~]# date
[root@xuegod63 ~]# timedatectl set-timezone Asia/Shanghai
[root@xuegod63 ~]# date
```

tzselect 了解

```
[root@xuegod63 ~]# tzselect                                     #Americas/United States/Eastern/yes
[root@xuegod63 ~]# echo " TZ='America/New_York'; export TZ
" >> /root/.bash_profile
[root@xuegod63 ~]# source /root/.bash_profile
[root@xuegod63 ~]# date
[root@xuegod63 ~]# echo $TZ
[root@xuegod63 ~]# sed -i '/^TZ/d' /root/.bash_profile #删除 TZ 开头的行
```

```
[root@xuegod63 ~]# tzselect                                     #Asia/china/beijing/yes
[root@xuegod63 ~]# TZ="Asia/Shanghai";export TZ
[root@xuegod63 ~]# date
[root@xuegod63 ~]# echo $TZ
[root@xuegod63 ~]# sed -i '/^TZ/d' /root/.bash_profile
```

### 2.3.2 如何修改时间

date 命令相关参数:

date --help

-s, --set=STRING 把时间设为字符串所描述的时间

例: [root@xuegod63 ~]# date -s "2019-11-2 22:30"

2019 年 11 月 02 日 星期五 22:30:00 CST

%F 完整日期格式, 等价于 %Y-%m-%d

```
[root@xuegod63 ~]# date "+%F"
```

2019-11-02

```
[root@xuegod83 ~]# hwclock -s          #以硬件时间为基准, 修改系统时间
```

```
[root@xuegod83 ~]# hwclock -w          以系统时间为基准, 修改硬件时间
```

%y 年份最后两位数位 (00-99)

%Y 年份

%m month (01..12)

%d 按月计的日期(例如: 01)

%M minute (00..59)

%H 小时(00-23)

%S 秒(00-60)

```
[root@xuegod63 ~]# date "+%Y%m%d"
```

20181102

```
[root@xuegod63 ~]# date "+%Y-%m-%d %H:%M:%S"  #在年月日之前可以添加自己想要的符号
```

2018-11-02 22:34:27

```
[root@xuegod63 ~]# date "+%Y/%m/%d %H:%M:%S"
```

2018/11/02 22:34:38

```
[root@xuegod63 ~]# date "+%Y%m%d %H:%M:%S"
```

20181102 22:35:03

-d, --date=STRING #显示由字符串描述的时间, 而不是“当前时间”

```
[root@xuegod63 ~]# date -d "+1months" +%F
```

2019-12-02

### 2.3.3 使用 time 命令测试一个命令运行的时间

time 作用: 一般用来测量一个命令的运行时间

使用方法: time 在后面直接跟上命令和参数

```
[root@xuegod63 ~]# time ls -l /etc/
```

real 0m0.056s

user 0m0.016s

sys 0m0.040s

说明:

real: 实际使用时间

user: 用户状态使用的时间

sys: 内核状态使用的时间

## 2.4 帮助命令使用

遇到命令不知道添加哪个参数, 可以使用命令帮助查看相关介绍, 常用的查看帮助信息命令有如下几个

### 2.4.1 man 命令

查看手册页或命令描述

```
[root@xuegod63 ~]# man find
```

man 命令查看帮助时, 支持它支持上翻下翻, 搜索(直接输入斜线), 退出用 q

### 2.4.2 使用-h 或--help 查看命令选项

```
[root@xuegod63 ~]# find -h #不可以执行
```

```
[root@xuegod63 ~]# find --help
```

## 2.5 开关机命令及 7 个启动级别

常用的几个关机, 重启命令

shutdown

init

reboot

poweroff

### 2.5.1 关机命令之--shutdown

作用: 关机, 重启, 定时关机

语法: shutdown [选项]

参数:

-r => 重新启动计算机

-h => 关机

-h 时间 => 定时关机

例如:

```
[root@xuegod63 ~]# shutdown -h +10 #10 分钟之后关机
```

```
[root@xuegod63 ~]# shutdown -h 23:30 #指定具体的时间点进行关机
```

```
[root@xuegod63 ~]# shutdown -r 22: 22 #22:22 以后重启
```

```
[root@xuegod63 ~]# shutdown -h now #立即关机
```

### 2.5.2 7 个启动级别

作用: 切换系统运行级别

语法: init 0-6

Linux 7 个启动级别:

0 系统停机模式, 系统默认运行级别不能设置为 0, 否则不能正常启动, 机器关的

1 单用户模式, root 权限, 用于系统维护, 禁止远程登陆, 就像 Windows 下的安全模式登录

2 多用户模式, 没有 NFS 网络文件系统 (Network File System) 和网络支持

3 完整的多用户文本模式, 有 NFS 和网络, 登陆后进入控制台命令行模式

4 系统未使用, 保留一般不用, 在一些特殊情况下可以用它来做一些事情。例如在笔记本电脑的电池用尽时, 可以切换到这个模式来做一些设置

5 图形化模式, 登陆后进入图形 GUI 模式, X Window 系

6 重启模式, 默认运行级别不能设为 6, 否则不能正常启动。运行 init 6 机器就会重启

例

```
[root@localhost ~]# init 0    #关机
[root@localhost ~]# init 3    #进入 3 级别字符界面
[root@localhost ~]# init 5    #进入 5 级别图形界面
```

### 2.5.3 设置默认的运行级别

centos8 不再使用/etc/inittab 文件进行默认的启动级别配置, 而使用比 sysvinit 的运行级更为自由的 target 替代。

第 3 运行级用 multi-user.target 替代。

第 5 运行级用 graphical.target 替代。

查看当前默认启动级别:

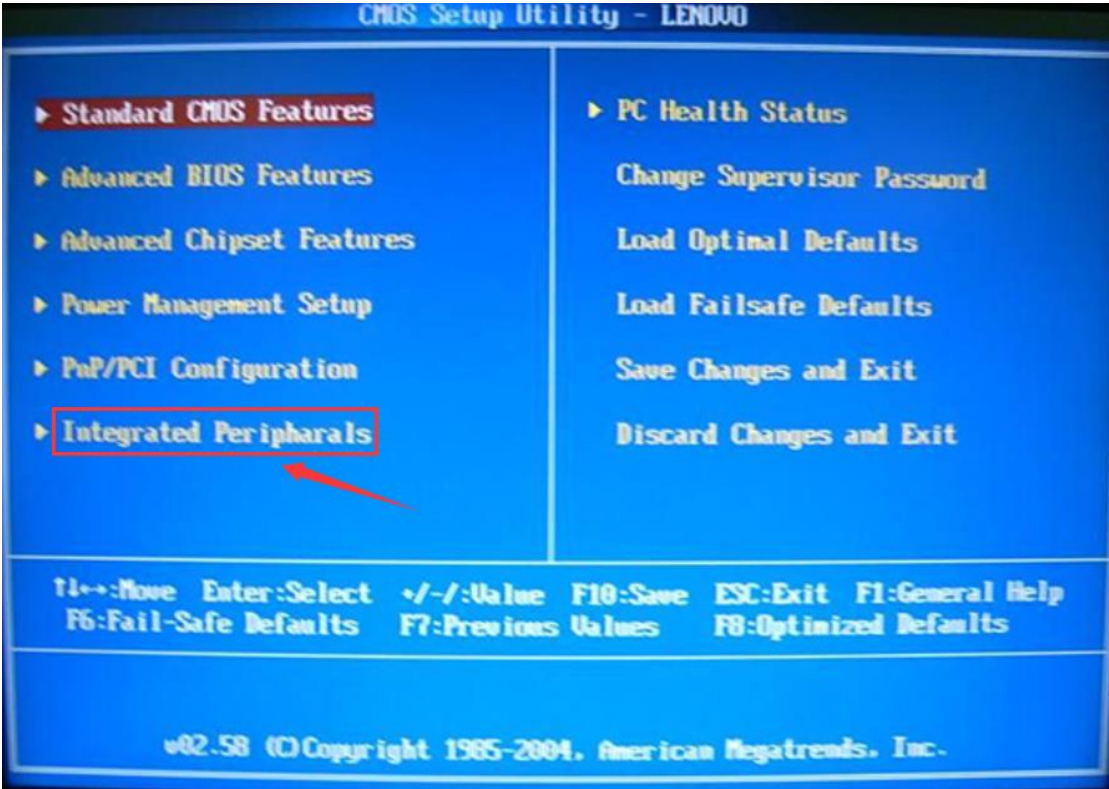
```
[root@xuegod63 ~]# systemctl get-default
[root@xuegod63 ~]# systemctl isolate multi-user.target    #切换级别为字符界面
# isolate [!aiseleit]  隔离, 脱离
[root@xuegod63 ~]# systemctl set-default multi-user.target #设置默认第三启动级别
[root@xuegod63 ~]# systemctl set-default graphical.target #设置默认第五启动级别
[root@xuegod63 ~]# systemctl isolate graphical.target    #切换级别为图形界面
```

```
[root@xuegod63 ~]# runlevel
```

3 5 #表示从 3 级别切换到了 5 级别

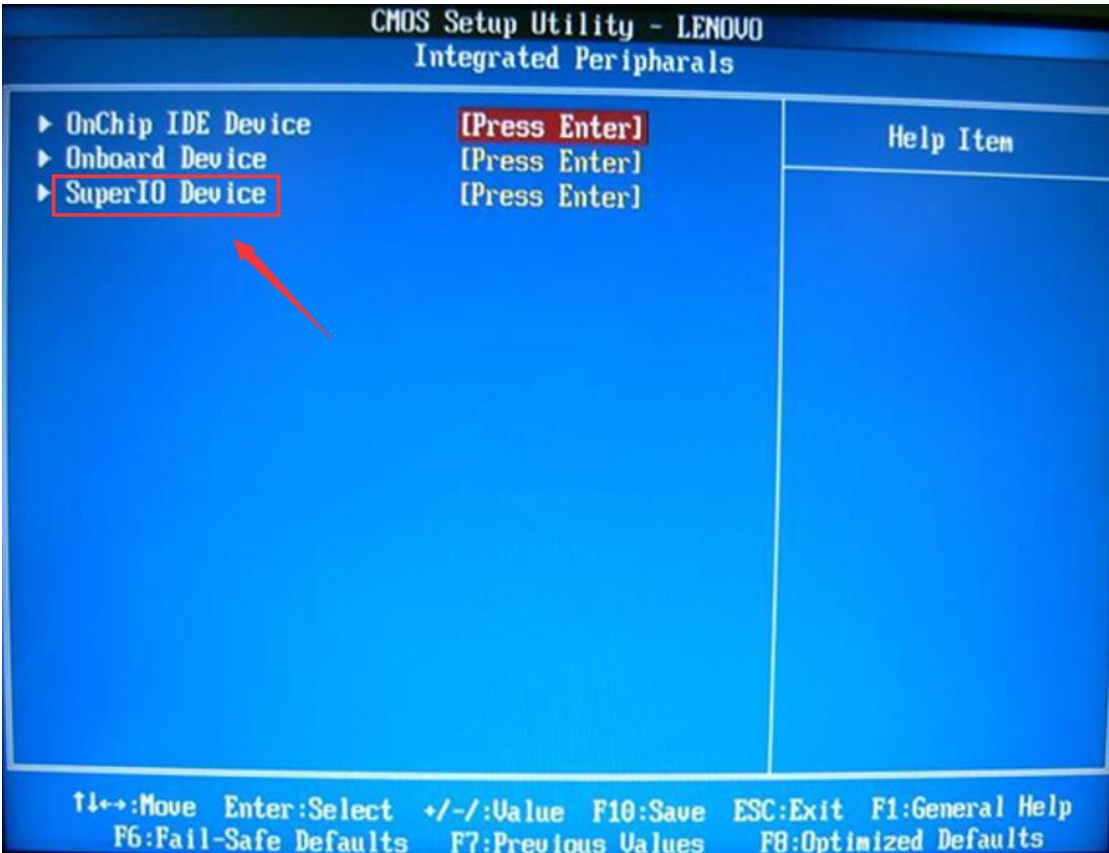
## 2.6 实战: 设置服务器来电后自动开机

进入 bios, 一般是在开机后出现主板画面是按 Delete 这个键, 部分品牌机可能按 F2, F1



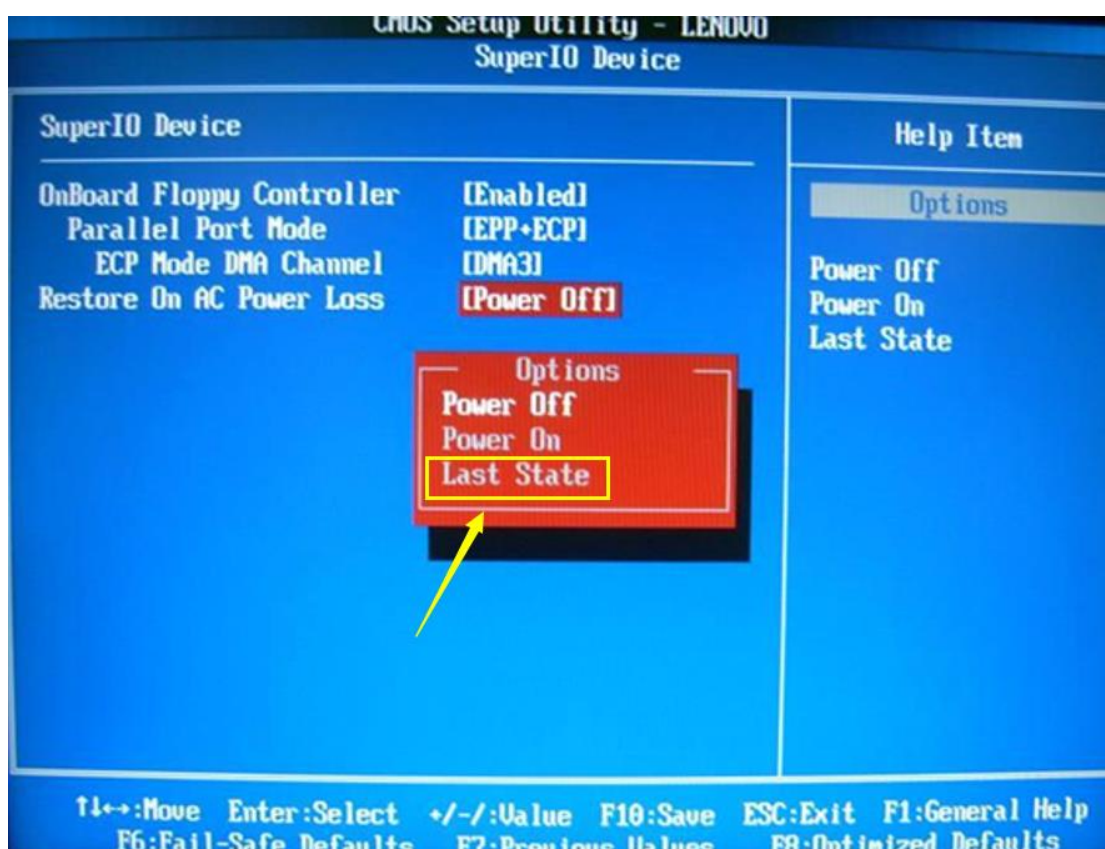
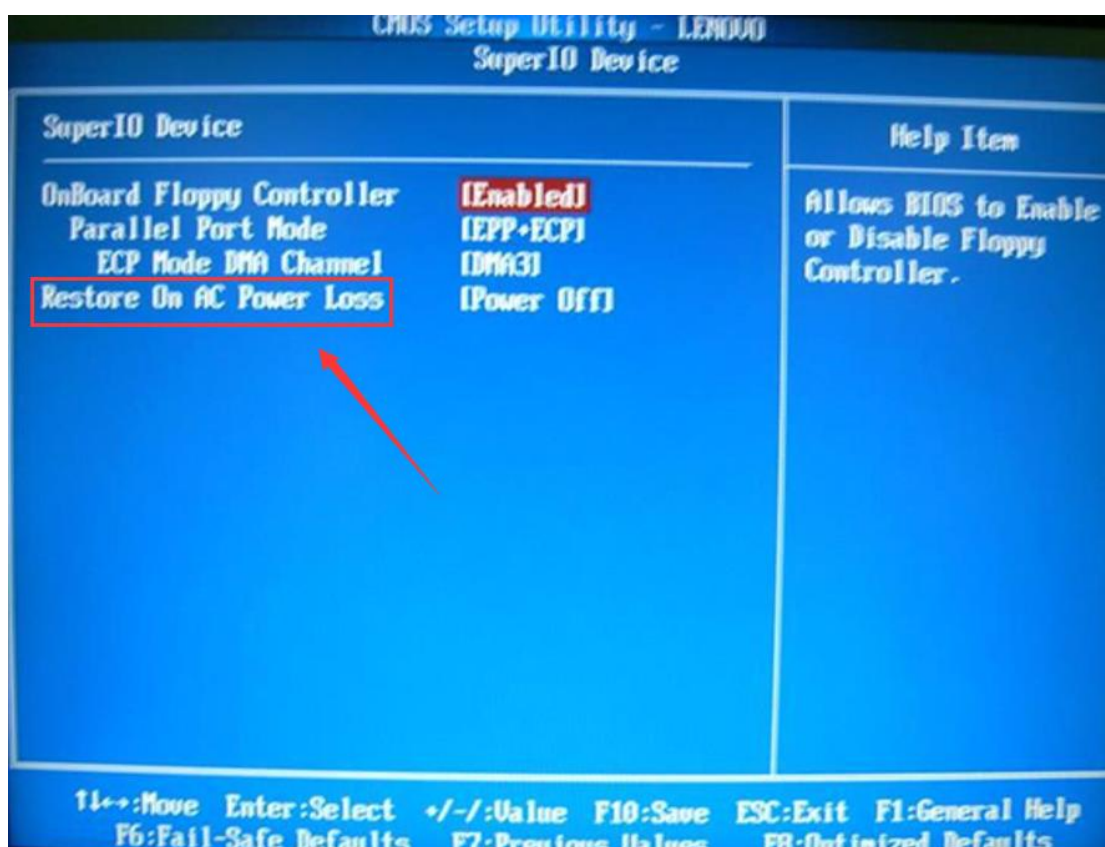
选择 Integrated Peripherals (外围设备设置) 中的 SuperIO Device

Integrated [ˈɪntɪɡreɪtɪd] 集成 ; Peripherals [pəˈrɪfərəl] 外围



将其中的 Restore On AC Power Loss 选项修改: Last State (此选项更好) 或 Power On  
交流电流(Alternating Current,缩写:AC)

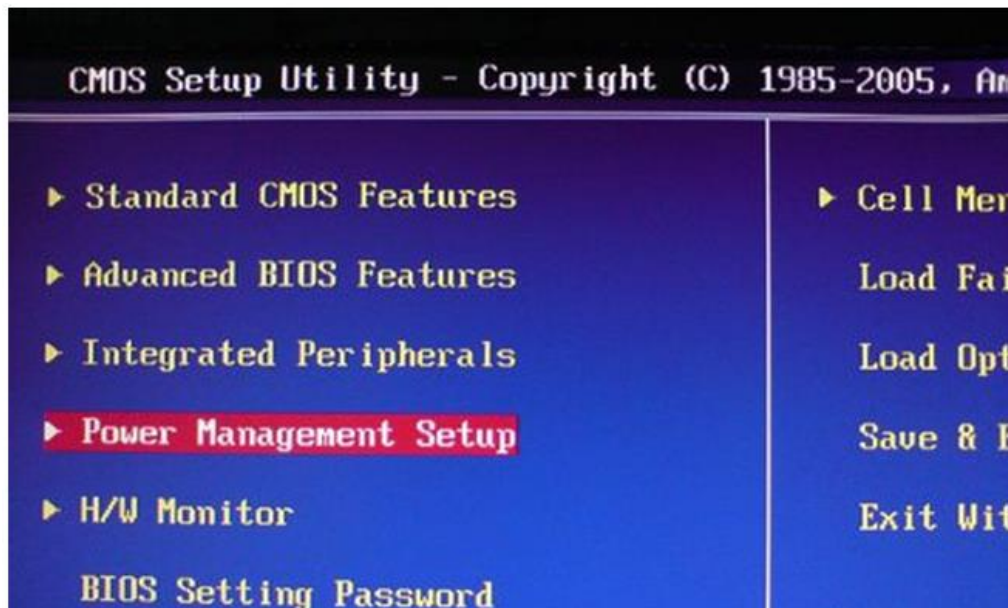




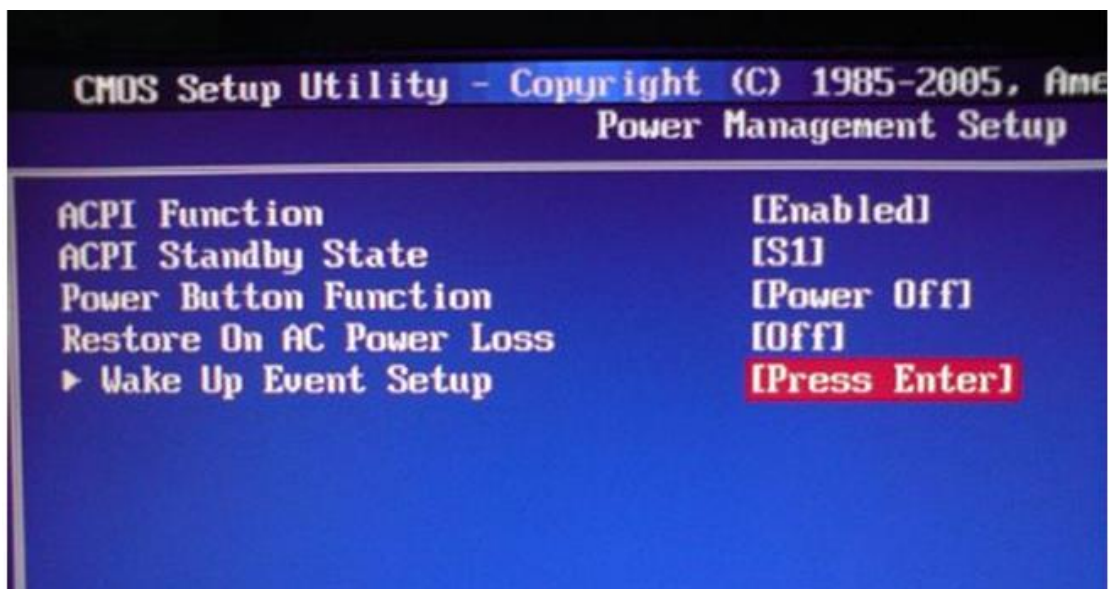
(若要加电不开机选择 Power Off, 若要保持断电前的状态选择 Last State)

## 2.7 设置服务器定时开机

Power Management Setup, 就进入电源管理设置了

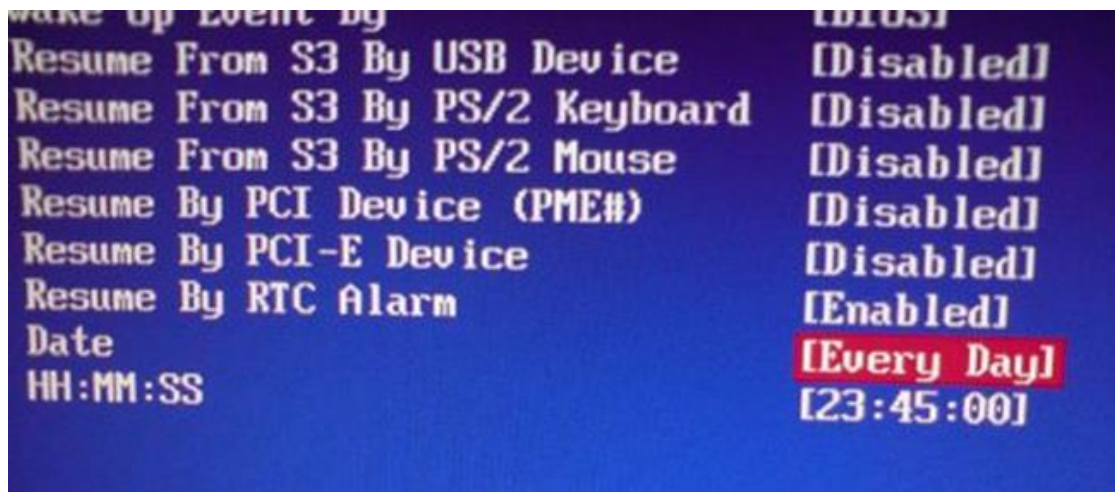


通过回车进入这个设置后, 选择 Wake Up Event Setup, 回车选择 Press Enter



找到 RTC Alarm ([ə'li:ɔ:m] 报警), 将 Disabied 更改为 Enabled, 然后继续回车确定。然后再继续设置时间点和日期





按 F10 保存, 退出。

## 总结:

- 2.1 Linux 终端介绍 Shell 提示符 Bash Shell 基本语法
- 2.2 基本命令的使用: ls、pwd、cd、history
- 2.3 查看系统和 BIOS 硬件时间
- 2.4 Linux 如何获得帮助
- 2.5 开关机命令及 7 个启动级别
- 2.6 实战: 设置服务器来电后自动开机
- 2.7 实战: 设置服务器定时开机