

一、题型

待更新

二、Linux简介

2.1 Linux系统组成部分

- Linux内核。

操作系统的核心部分，包括基本的系统启动核心信息、硬件驱动程序的。内核从应用层接受命令，根据调度算法调度进程并分配系统资源，使得程序能够顺利执行。

- Linux Shell

操作系统的用户界面，提供应用于内核进行交互操作的统一接口。其实质是一个命令解释器。*Shell也是用户接口*

- Linux文件系统

是文件数据在磁盘等存储介质上的存储规则。

- Linux应用程序 **Linux操作系统指的是“linux内核+应用”**

2.2 Linux版本

2.2.1 概念

- **内核版**：由linux带领工作组负责维护，现在主流内核版本依然是2.6.x
- **发行版**：在选定的内核基础上，进行修订，并添加相关功能软件，发行的版本。

2.2.2 内核版本区分

Ps：稳定版or测试（实验）版？

- **Linux的内核版本命名总体有三种规格：**

(1) **纯小数**：0.01~1.0之前；

(2) **A.B.C**：1.0~2.6；A表示主版本号，B表示次版本号（**偶数：稳定版**，**奇数：测试版**），C表示修改更新的次数；如：2.3.5；

(3) time-based方式，两种：

→ **A.B.C.D** (2.6之后)：A、B同上。C表示末版本号，D表示驱动、Bug等更新的次数；

➤ **A.B.C** (3.0之后)：A表示主版本号，B表示末版本号，C表示驱动、Bug等更新的次数；

2.3 虚拟控制台

2.3.1 定义

虚拟文本控制台

- **虚拟文本控制台**，又被称为**虚拟终端(tty)**，提供给用户一个使用命令行的字符界面，用于接收用户输入和反馈计算结果。
- 像Windows下的命令行状态，Linux、UNIX下的字符终端程序，都称为虚拟控制台。

第几个tty就是tty0

tty0, tty1, ..., tty5 (默认对等开启5个tty)

超链接(加法): CSDN查询R62

2.4 Shell

2.4.1 概念

Shell也是用户接口

- Shell是一个命令解释器，将用户输入的命令进行适当的解释，然后提交给内核去执行，并将内核执行的结果显示给用户。Shell还有自己的语言，允许用户编写由Shell命令组成的程序。每个Linux用户都可以拥有自己的用户界面或Shell，Shell也有多种版本。目前主要有下面4种版本的Shell。

(1) BASH: GNU的Bourne Again Shell，是GNU操作系统上默认的Shell。

(2) Bourne Shell: 是贝尔实验室开发的Shell。ps:1与2本质相同。

(3) Korn Shell: 是对Bourne Shell的发展，大部分内容与Bourne Shell兼容。

(4) C Shell: 是SUN公司Shell的BSD版本。

2.4.2 Shell命令格式

- Shell的命令通常具有固定的格式，以方便用户进行操作，其一般格式如下：

命令名 [选项] [参数1] [参数2] ...

eg: cp -r src dsc
① dsc为文件，则不执行
② 本命令用于复制src目录下的all内容到dsc目录
③ -r改为r则只复制src，不复制内容

其中各部分的含义如下：

即所需的参数

命令名：需要提交给系统执行的命令，这些命令是一个可执行文件或Shell脚本文件。

选项：是对命令的特别定义，以短线(-)开始。(也可用--双连字符，eg: -f, --force)

参数：是提供给命令运行的信息或命令执行过程中所使用的文件名。

若多条命令要执行，可将这些命令输入在一行中，各命令之间用分号(;)隔开即可。

2.4.3 命令提示符

命令提示符 (*[~]# 或 [~]\$*)

- 登录Linux系统的文本界面后，会出现以“#”或者“\$”结束的命令提示行。

是用户名
是主机名
是当前目录
命令提示符

[root@localhost root]#

当前用户名 Linux主机名 当前目录 命令提示符

其中“#”是管理员的命令提示符，“\$”是普通用户的命令提示符。(i.e. #是身份提示符)

命令提示符用于指示用户输入命令的位置，只有在命令提示符后面输入的命令系统才会解析执行。

用户	用户家目录
root	/root
其他用户	/home/用户名 eg: wacha

2.5 Systemctl的2种用法及命令格式

2.5.1 用法 1：管理服务

命令

systemctl命令格式： `systemctl <command> <unit>`

说明：

- **unit**：为服务单位名称，即systemctl管理的服务文件后缀为service；
- **command**为操作方式，主要有：

- **start**：启动unit；
- **stop**：关闭unit；
- **restart**：重启unit；
- **reload**：不关闭unit，加载配置信息；

- **enable**：设置开机自启动unit；
- **disable**：关闭开机自启动unit；
- **status**：查看unit状态；

- **mask**：注销unit；

- **umask**：重置unit；

即不重启服务的情况下，加载服务的配置信息

- **is-active**：查看是否在运行；

- **is-enable**：查看是否开机自启动；

- **list-units**：列出已启动的unit；

Ps: unit简记为“服务”。

举例

以httpd.service（用于提供网页服务）为例

1、启动服务（start）

```
sudo systemctl start httpd.service
```

作用：用于启动httpd服务。httpd.service是服务单元名称，sudo是因为启动服务通常需要管理员权限，root用户下无需输入sudo。

2、重启服务（restart）

```
sudo systemctl restart httpd.service
```

作用：如果httpd服务已经在运行，想要重启它，该命令会让httpd服务停止，然后重新启动。

3、不关闭服务加载配置信息（reload）

```
sudo systemctl reload httpd.service
```

作用：当修改了httpd服务的配置文件（比如/etc/httpd/conf/httpd.conf）后，不想重启服务但能让新配置生效。

4、设置开机自启（enable）

```
sudo systemctl enable httpd.service
```

作用：希望httpd服务在系统开机时自动启动。

5、查看服务状态（status）

```
sudo systemctl status httpd.service
```

作用：查看httpd服务当前是运行、停止还是有错误等状态。终端会显示服务的运行状态信息，包括是否正在运行、运行时长、启动时间等内容。

2.5.2 用法 2：设置系统运行的环境

说明：

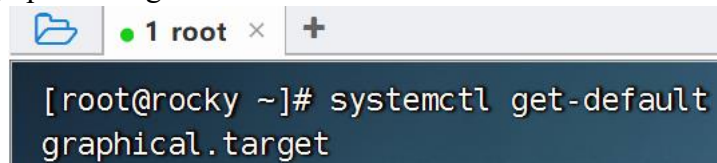
- 当unit为target时，可以操纵当前的操作系统运行在不同的环境。
- command为操作方式，主要有：
 - **get-default**：获得当前的target；
 - **set-default**：设置默认的运行环境target；
 - **isolate**：切换到后接的target，而不需要重启。
- 常见的target有：
 - **graphical.target**：图形界面环境；
 - **multi-user.target**：命令行环境；
 - **emergency.target**：紧急处理环境，需要root登录；
 - **rescue.target**：救援模式，无需root登录；
 - **shutdown.target**：关机模式；

举例

1、获取当前的target（get-default）

```
systemctl get-default
```

作用：执行该命令后，系统会返回当前默认的运行目标。若系统默认运行在图形界面环境，则返回graphical.target；若在命令行环境下，返回multi-user.target。



```
[root@rocky ~]# systemctl get-default
graphical.target
```

2、设置默认的运行环境target（set-default）

将系统默认的运行目标设置为命令行环境，使用命令：

```
sudo systemctl set-default multi-user.target
```

作用：执行此命令后，系统会将默认运行目标设置为multi-user.target，下次重启系统时，就会默认进入命令行环境。

将系统默认运行目标改回图形界面环境，使用命令：

```
sudo systemctl set-default graphical.target
```

2.6 Systemd概述

历史上，Linux的启动往往采用init进程。这种方法有两个缺点：启动时间长、启动脚本复杂。Systemd就是为了解决这些问题而诞生的。Systemd的设计目标是：守护整个系统，为系统的启动和管理提供解决方案。Systemd取代了init，成为系统的第一个进程，PID值为1。其他进程都是它的子进程。守护进程（daemon）的缩写：d

CentOS 7.x以后，RedHat系列放弃沿用System V的init服务流程，转而使用systemd服务管理机制。

systemd机制由systemd守护进程常驻内存，配备唯一的systemctl命令进行管理。

systemd的特点:

- 并行处理所有服务, 加速开机流程;
- 采用on-demand响应启动方式;
- 服务依赖性的自我检查;
- 依daemon功能分类: unit、type;
- 将多个daemons集合称为一个群组;
- 向下兼容旧有的init脚本。

systemd将过去的daemon执行脚本通常为一个服务单位 (unit), 然后按照每个服务单位的功能不同进行分类, 形成不同的类型 (type)。

基本的类型包括:

type	扩展名	主要功能
一般服务类型	.service	主要是系统服务, 包括服务器及网络服务等
内部程序数据交换的socket服务	.socket	主要是IPC的传输信息socket文件功能
快照类型	.snapshot	监控存储系统状态的快照功能
操作环境类型	.target	提供类似运行级别的操作系统执行环境功能, 是一群unit集合
目录挂载服务	.mount 或 .automount	文件系统挂接相关的服务。
文件检测服务	.path	检测特定文件或目录功能, 如打印队列监控。
循环执行服务	.timer	类似anacrontab服务

与systemd相关的目录:

- /usr/lib/systemd/system: 服务的启动脚本目录, 类似于/etc/init.d目录;
- /run/systemd/system: 系统执行过程中产生的服务脚本, 优先级高于 /usr/lib/systemd/system目录中的文件;
- /etc/systemd/system: 管理员根据需求创建的执行脚本, 类似于 /etc/rc.d/rc5.d/Sxx文件, 优先级高于/run/systemd/system目录中的文件。

2.7 Linux系统中磁盘设备命名

设备接口	设备名	设备文件 (对于 /dev/ 设备)
IDE设备 <i>硬盘</i>	hd, 通常允许4个IDE设备, 故分别用hda、hdb、hdc和hdd表示	/dev/hda、/dev/hdb、 /dev/hdb、/dev/hdd
SCSI/SATA/U盘 <i>小型计算机设备</i>	sd, 通常允许16个SCSI设备, 故分别用sda、sdb、...、sdp表示	/dev/sda、/dev/sdb、...、 /dev/sdo、/dev/sdp
SSD (NVMe) <i>固态硬盘</i>	nvme0n, 若干个NVMe设备, 故分别用nvme0n1、nvme0n2、... nvme0nM表示 <i>nvme0n1, nvme0n2, ...</i>	/dev/nvme0n1、/dev/nvme0n2
打印机 <i>sda : sda0, sda1, ... sdb</i>	lp, 最多可以接16个打印机	/dev/lp0、/dev/lp1、...、 /dev/lp14、/dev/lp15
光驱 <i>CPU</i>	sr, 同时又别名cdrom	/dev/sr0、/dev/cdrom

以IDE设备为例（传统MBR分区方式）：

/dev/hda^{hdb}1、/dev/hdb、/dev/hdc、/dev/hdd 通常主机上最多有4个IDE接口

/dev/hda1、/dev/hda2、/dev/hda3、/dev/hda4 第一块IDE硬盘的四个主分区，或3个主分区+1个扩展分区（一块硬盘在MBR分区模式下，最多可以有4个主分区，或3个主分区和1个扩展分区）

/dev/hda5、/dev/hda6、…… 为逻辑分区，即/dev/hda5为第一个逻辑分区

以NVMe设备为例（传统MBR分区方式）：

/dev/nvme0n1、/dev/nvme0n2 通常主机上有2个NVMe接口

/dev/nvme0n1p1、/dev/nvme0n1p2、/dev/nvme0n1p3、/dev/nvme0n1p4 第一块NVMe硬盘的四个主分区，或3个主分区+1个扩展分区（一块硬盘在MBR分区模式下，最多可以有4个主分区，或3个主分区和1个扩展分区）

/dev/nvme0n1p5、/dev/nvme0n1p6、…… 为逻辑分区，即/dev/nvme0n1p5为第一个逻辑分区

2.8 查看系统信息命令（了解）

看系统信息命令

查看Linux内核版本的命令：uname -r或uname -a。

查看文件系统的磁盘空间大小和剩余空间大小的命令：df。

显示系统已经运行了多长时间的命令：uptime。

查看当前系统内存的使用情况的命令：free。

查询有关CPU的详细硬件信息命令：cat /proc/cpuinfo。

查看CPU的使用情况和正在运行的进程情况的命令：top。

查看登录日志信息的命令：last。

查看登录用户信息的命令：w [用户名]。

显示月历或年历的命令：cal。

显示或设置当前日期和时间的命令：date。

→管道符，是单向的，能让2个进程进行通信。会把前一个命令的执行结果作为后一个命令的输入参数。

命令 --help | more

Linux下磁盘空间大小以块(block)为单位，1块默认为1KB。

Linux下蓝色暗：目录

黑色：文件

淡蓝/绿色：链接

绿色：可执行文件

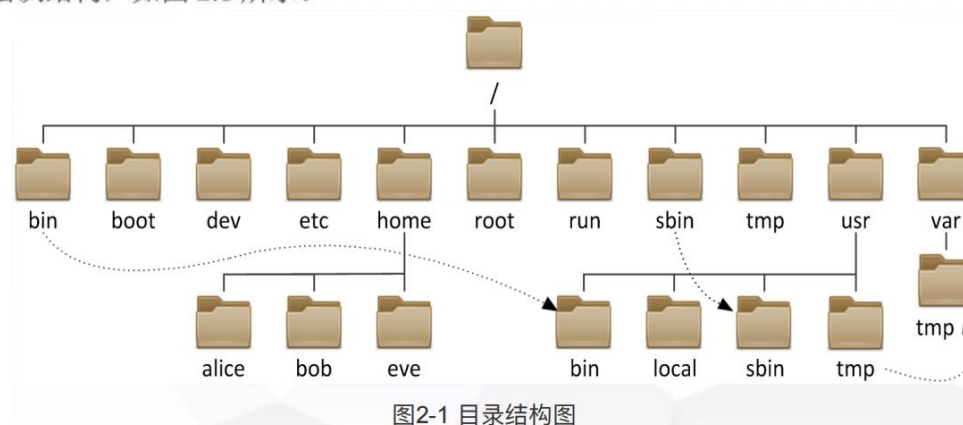
Flags：当前处理器支持的描述。

三、文件系统

3.1 Linux的树形目录结构

3.1.1 概念

Linux 系统中目录的概念，类似于 Windows 系统中的文件夹概念。Linux 系统只有一个根目录“/”，采用树形目录结构来组织和管理文件，每个文件都有文件名，并编排在相应目录下。Linux 系统中，目录也是文件。所有的文件采取分级、分层的方式组织在一起，从而形成了一个树形的层次结构，如图 2.1 所示。



3.1.2 常用目录介绍

/bin: 存放系统常用命令，这个目录中的文件都是可执行的、普通用户可以使用的命令。
/dev: 设备文件（比如声卡、磁盘文件）存储目录。
/boot: 存放 Linux 内核及引导系统程序文件。
/etc: 存放系统配置文件，某些服务器的配置文件也存放于此。
/home: 普通用户主目录的默认存放位置。
/lib: 库文件存放目录。

/tmp: 临时文件目录。程序运行产生的临时文件存放于此。这个目录和/var/tmp 目录相似。

/usr: 系统存放程序的目录。当安装一个 Linux 发行版本官方提供的软件包时，大多安装在这里。如果有涉及服务器配置文件的，会把配置文件安装在/etc 目录中。/usr 目录下包括设计字体目录/usr/share/fonts, 帮助目录/usr/share/man 和/usr/share/doc, 普通用户可执行文件目录/usr/bin 或/usr/local/bin, 超级权限用户 root 可执行命令存放目录/usr/sbin 或/usr/local/sbin 等，以及程序的头文件存放目录/usr/include 等。

Ps: 前 6 个目录用途要记住。此外，usr是user的缩写、/bin与/usr/bin功能相同、/lib与/usr/lib功能相同，细节不赘述，可参考：<https://www.zhihu.com/question/21265424>。