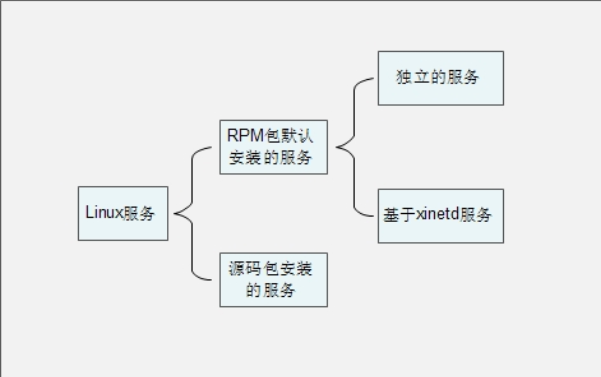
Linux服务管理

**服务分类**



二进制包

1. 快速安装、卸载、升级和管理软件

2. 安装简洁、速度快

3. 经过封装，无法直接获取源代码

4. 功能选择性差，功能定制不灵活

源码包

1. 获得最新的软件版本，及时修复bug

2. 根据用户需要，灵活定制软件功能

**RPM包的服务管理**

RPM包所安装的软件有特定的管理命令，管理方式分为两种：

* 独立服务：

所谓的独立启动，就是可以自行管理启动，和其他的软件的启动与否没有关系，不依赖其他软件的启动，此类软件当被调用时会直接响应，处理请求的速度快。目前Linux中绝大多数服务为独立服务。

手动启动：

1. 启动脚本启动，/etc/init.d/某服务脚本文件

/etc/init.d/httpd start

2. service服务管理工具

service httpd start | stop | restart

自启动：

1. 使用chkconfig服务自启动工具管理

查看软件的自启动：

chkconfig –list | grep 服务名

设置某服务的自启动：

chkconfig [--level 级别] [独立服务名] [on | off]

2. 使用ntsysv工具设置自启动软件

netsysv --level 级别 #设置某级别下软件是否自启动

F1 ：显示服务的说明信息

* 基于xinetd的服务：

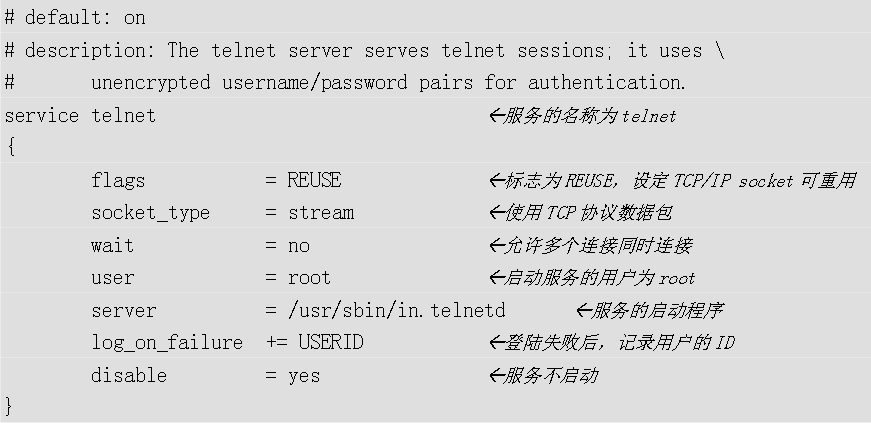
此类服务无法独立启动，必须依赖某一管理服务来启动或者关闭，这个管理服务就是我们xinetd服务了，此服务是Linux的超级守护进程，专门用来管理那些无法自行启动的服务，当客户端请求服务启动时，请求会先被xinetd服务接受，然后xinetd去唤醒对应的服务，打到启动某服务的目的。这类服务不会一直处于开启状态，只有在xinetd服务接受到请求的时候才会唤醒对应服务，处理完后服务关闭，这样大大节省了服务器的资源，但是对应的软件的反应速度没有独立服务反应快。

基于xinetd服务的—telnet

此处我们用telnet服务来举栗子！telnet是远程管理命令，但由于传输过程是明文，so… 我们不选用他，我们使用更安全的ssh，telnet的端口是23，我们借助xinetd来启动下我们的telnet。

1. 手动启动xinetd中的某服务

vim /etc/xinetd.d/telnet



此处我们只需要将disable = yes 修改成 no 即可！

然后重新启动xinetd服务（注意我们现在是借助xinetd来管理的telnet）

2. 设置xinetd内的服务自启动

同样借助chkconfig 和 ntsysv 来实现自启动，但是这个地方我们需要注意的是，我们不是设置某个具体服务的自启动，而是设置的xinetd的自启动。

**源码包的服务管理**

1. 源码包服务的手动启动—指定路径启动

eg. /usr/local/apache2/bin/apachectl start | stop | restart …………

2. 源码包服务的自启动

借助/etc/rc.d/rc.local 或者 /etc/rc.local

写入手动启动命令即可

3. 让系统识别源码包

对照刚才看的RPM包安装的httpd的启动脚本文件我们自己写一个让系统识别我们的源码软件。

注意：脚本中有两句话必须存在！作用让chkconfig识别此脚本，并能执行。

