

Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心



Linux 应用软件安装使用 原理扫盲

博士研究生 刘望桐

2018年12月16日

- 背景知识
 - Linux文件目录机制
 - Linux软件安装
 - Linux软件运行
 - yum、apt-get
 - 其他
- 预期收获
 - 了解Linux文件目录及分区机制
 - 掌握Linux下软件安装原理及运行方法
 - 掌握yum、apt-get的基本使用和配置方法



背景知识

- 电脑中的应用软件主要包括3个部分
 - 可执行的2进制文件——exe
 - 软件执行所需要的数据（真·数据、配置文件等）
 - 动态链接库——dll



- 只要使用的寻址方式恰当，3者可以不在同一个目录下

- 应用软件的安装过程不外乎4个步骤
 - 将软件包含的所有文件解压
 - 编译生成可执行的二进制程序（可选）
 - 将二进制程序、数据、配置文件、动态链接库移动到**恰当**的地方（分门别类放好，但互相知道彼此的位置）
 - 修改系统中与软件相关配置文件和参数
- 在Windows平台下
 - 运行安装软件->选择安装路径->解压、移动文件->修改系统配置文件和参数（注册表、环境变量）->安装完成
 - 绿色版/硬盘版=只需要解压和移动文件

- 应用软件运行的2个重要步骤
 - 执行可执行的二进制程序
 - 手动执行（双击exe）
 - 启动器启动
 - 附带启动
 - 条件自启动
 - 二进制程序找到自己需要配置文件、数据、动态链接库等文件
 - exe当前目录、子目录
 - 配置参数中指定的目录
 - 系统默认目录（环境变量）

- Linux环境与Windows下软件的运行与安装有一定差异
 - 所有文件均**允许**被“执行”，而不局限于exe文件
 - Linux下后缀名与文件类型没有绝对的关联性
 - **文件目录结构**不同，没有“C盘=系统盘”的概念
 - 用户在安装前、中、后均有更多的配置空间，软件可以更加个性化、定制化
 - 安装目录、位置文件目录、库文件目录都可以配置
 - 很多时候运行软件需要通过命令行而非UI界面
 - Linux下的动态链接库是.so文件



Linux文件目录

- Windows的文件目录结构以“我的电脑”为根节点，1级节点为各个分区（C盘、D盘、E盘）
- Linux的文件目录结构以根目录“/”为根节点，1级节点为各个主要目录



- /, 根目录
 - boot, 保存了系统启动过程中所需要各项数据, 例如系统引导镜像
 - home, 保存用户数据
 - root, 保存root用户数据
 - etc, 保存各种系统配置数据
 - usr, 常态化数据、软件目录
 - bin, 可执行文件
 - lib, 动态链接库
 - doc, 可读文件
 - var, 经常变化的数据、软件目录

- 不同于Windows直接按照1级目录（C\D\E盘）进行磁盘分区，Linux的磁盘分区和文件目录是相对独立的
 - 关于分区：2018-06-10，计算机启动流程详解2
- Linux的磁盘分区通过“挂载”的方式和文件目录连接，任意磁盘分区可以挂载在任意文件目录上

```
[root@ff007bit bfs_server]# df -hl
```

文件系统	容量	已用	可用	已用%%	挂载点
/dev/mapper/vg_ff007bit-lv_root	18G	8.9G	7.6G	54%	/
tmpfs	504M	80K	504M	1%	/dev/shm
/dev/sda1	485M	52M	408M	12%	/boot

- 根目录必须挂载1个分区
- 已挂载分区的目录下所有未挂载其他分区的子目录及文件属于同一分区
- 和分区关联的属性有
 - 文件系统
 - 磁盘物理空间位置和大 小
 - 分区管理操作

/ ————— 挂载分区1

| — A

| | — A1 — 挂载分区2

| | — A2

| — B

| — C ————— 挂载分区3

分区1: /、A（不包括A1）、A2、B

分区2: A1

分区3: C



- 通常/boot目录会单独挂载一个分区，类似Windows的C盘，作为系统分区，承担启动、引导操作系统的任务
- 某些具有特殊功能的文件夹需要单独挂载分区，例如备份用文件夹
- 新插入的移动硬盘、U盘均需要挂早到文件目录中才能访问
 - `mount /dev/sda /mnt/usb_disk`
- Linux文件目录机制对软件使用的影响
 - 安装系统时需要慎重进行分区和挂载
 - 虚拟机扩容磁盘后需要对扩容的部分进行分区和挂载
 - 合理选择将软件安装到哪一个文件夹



Linux软件安装

- Linux平台下应用软件安装有2种方式：源码安装、RPM包安装
- 源码安装
 - 下载应用软件源代码，在本地搭建编译环境，编译后安装
 - 优点：定制化程度高、安装后软件与系统更加匹配
 - 缺点：安装过程麻烦，报错信息不宜处理
- RPM包安装
 - 下载应用软件对应RPM包，利用RPM命令安装，或直接使用RPM软件管理器（yum、apt-get）安装
 - 优点：安装、管理便捷
 - 缺点：定制化程度低、不便于安装后调整



- 源码安装
 - 下载源码安装包
 - configure
 - Makefile

- 源 /home/bfs/nginx-1.14.1

名字	扩展	大小	已改变	权限	拥有者
..			2018/12/10 11:09:26	rwxr-xr-x	bfs
auto			2018/11/9 22:26:11	rwxr-xr-x	bfs
conf			2018/11/9 22:26:11	rwxr-xr-x	bfs
contrib			2018/11/9 22:26:11	rwxr-xr-x	bfs
html			2018/11/9 22:26:11	rwxr-xr-x	bfs
man			2018/11/9 22:26:11	rwxr-xr-x	bfs
objs			2018/11/9 22:27:21	rwxrwxr-x	bfs
src			2018/11/9 22:26:11	rwxr-xr-x	bfs
CHANGES		287,441	2018/11/6 21:52:53	rw-r--r--	bfs
configure		2,502	2018/11/6 21:52:46	rwxr-xr-x	bfs
LICENSE		1,397	2018/11/6 21:52:46	rw-r--r--	bfs
Makefile		372	2018/11/9 22:26:54	rw-rw-r--	bfs
README		49	2018/11/6 21:52:46	rw-r--r--	bfs
CHANGES.ru		438,114	2018/11/6 21:52:51	rw-r--r--	bfs

- 源码安装
 - 下载源码安装包
 - configure
 - Makefile
 - ./configure
 - 检测当前机器的相关条件是否满足编译安装的需求（编译器、依赖的其他软件等）
 - 根据检测结果，会在最后提示缺什么，或者一切OK
 - 根据运行configure时附带的参数，将会调整后续编译、安装时的相关参数
 - ./configure --prefix=/usr --with-xen --with-libvirt
 - 通常./configure --help可以查看所有配置参数的含义

```
bfs@ubuntu:~/nginx-1.14.1$ ./configure --help
```

```
--help                print this message

--prefix=PATH         set installation prefix
--sbin-path=PATH      set nginx binary pathname
--modules-path=PATH   set modules path
--conf-path=PATH      set nginx.conf pathname
--error-log-path=PATH set error log pathname
--pid-path=PATH        set nginx.pid pathname
--lock-path=PATH      set nginx.lock pathname
```

- 源码安装
 - 利用Makefile文件编译并安装
 - Makefile是make命令所需要的一个配置文件，是绝大多数Linux平台下源码编译的必需品
 - Makefile中定义了很多编译、安装的相关操作的具体步骤，可以极大提升软件开发、编译、安装的效率
 - 常见命令
 - make
 - make clean
 - make install

```
default:    build

clean:
    rm -rf Makefile objs

build:
    $(MAKE) -f objs/Makefile

install:
    $(MAKE) -f objs/Makefile install

modules:
    $(MAKE) -f objs/Makefile modules

upgrade:
    /home/bfs/nginx/sbin/nginx -t

    kill -USR2 `cat /home/bfs/nginx/logs/nginx.pid`
    sleep 1
    test -f /home/bfs/nginx/logs/nginx.pid.oldbin

    kill -QUIT `cat /home/bfs/nginx/logs/nginx.pid.oldbin`
```

- RPM包安装
 - 下载RPM包
 - 通过RPM管理软件安装
 - `rpm -ivh ***.rpm`
 - i表示安装，v表示显示安装过程，h表示显示进度
 - `yum install [软件]`
 - `apt-get install [软件]`
 - 不同Linux发行版有不同的RPM管理软件
 - rpm/yum: CentOS、Redhat、Suse
 - apt-get/dpkg: Ubuntu、Debian

- RPM包安装
 - 通过RPM包方式安装的软件可以通过rpm管理软件进行管理，包括升级、卸载、重安装
 - RPM包中记录了该软件的依赖软件，利用rpm管理软件的相关命令可以查看依赖软件，或者在安装时直接一起安装
- RPM包的安装过程是完全透明的
 - 方便
 - 不便于后续调整、定制化



Linux软件运行



- 软件运行的3个关键点
 - 找到并运行可执行的二进制代码
 - 软件找到所需的数据
 - 软件找到所需的动态链接库

- 找到并运行可执行的二进制代码
 - 代码在哪里？
 - 各类bin文件夹， /bin， /usr/bin， /usr/local/bin， /var/bin……
 - 安装时通过参数配置好的安装位置
 - 如何运行？
 - [软件名]
 - ./[可执行代码文件]（相对路径执行）
 - /xxx/xxx/xxx/xxx/[可执行代码文件]（绝对路径执行）
 - /etc/rc.d/init.d/httpd start
 - service [软件] start

- 直接通过[软件名]运行
 - python xxx
 - sh xxx
 - vim xxx
- 本质是该软件的相关可执行文件放在了系统变量\$PATH中的目录下

```
[root@iz2zea99qngm2vogtdtw98Z ~]# echo $PATH
/usr/local/inotify/bin:/usr/local/inotify/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/root/bin
```

- 直接通过软件名运行时系统会在PATH的各个目录中寻找同名可执行文件并运行
 - 多个同名可执行文件按在PATH里的先后顺序执行

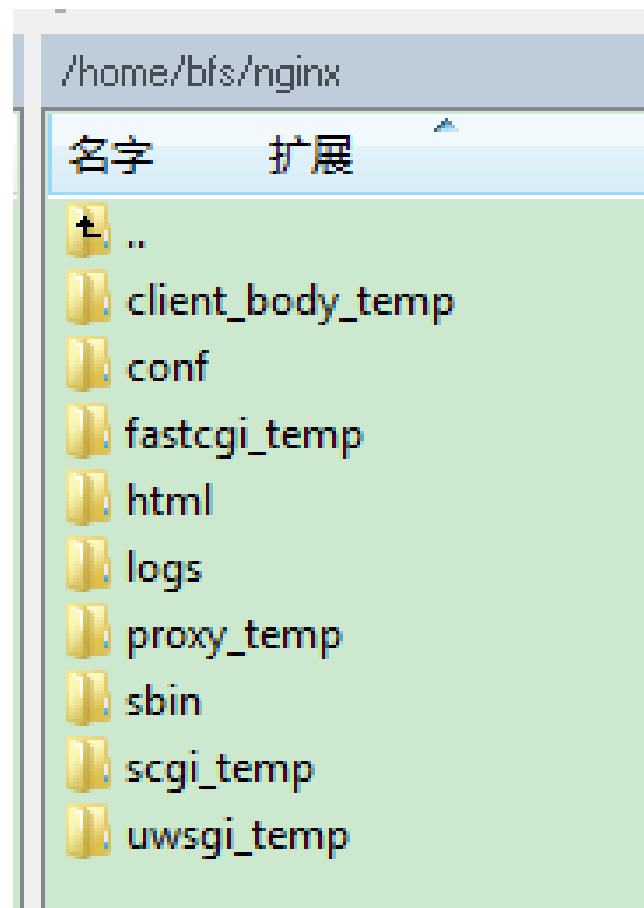


- `/etc/rc.d/init.d/httpd start`
- `service [软件] start`
 - 这2种运行方式的具体使用方法与Linux的发行版有关，不能照搬照抄
 - 其原理本质是绝大多数服务性质（Web服务、FTP服务等）的软件会在`/etc/rc.d/init.d/`下留有一份可执行文件，利用这2种方式会运行该目录下对应的可执行文件
 - 但如果是通过源码编译安装的软件就不一定能这么启动
 - Nginx

- 软件找到所需的数据
 - RPM包安装的软件不需要用户关注其数据所在位置
 - 大部分情况下在 /var、/data、/usr/local等文件夹下
 - 通过 `rpm -ql [软件名]` 命令可以找到与软件相关的文件的位置

```
[root@ff007bit ~]# rpm -ql httpd
/etc/httpd
/etc/httpd/conf
/etc/httpd/conf.d
/etc/httpd/conf.d/README
/etc/httpd/conf.d/welcome.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/conf/magic
/etc/httpd/logs
/etc/httpd/modules
/etc/httpd/run
/etc/logrotate.d/httpd
/etc/rc.d/init.d/htcacheclean
/etc/rc.d/init.d/httpd
/etc/sysconfig/htcacheclean
/etc/sysconfig/httpd
/usr/lib/httpd
/usr/lib/httpd/modules
/usr/lib/httpd/modules/mod_actions.so
/usr/lib/httpd/modules/mod_alias.so
/usr/lib/httpd/modules/mod_asis.so
/usr/lib/httpd/modules/mod_auth_basic.so
/usr/lib/httpd/modules/mod_auth_digest.so
```

- 软件找到所需的数据
 - 源码编译安装的软件的相关数据根据configure时的参数在编译完成后被移动到对应位置
 - 大多数情况下和可执行文件在同一个位置
 - ./configure --prefix=/home/[软件名]则几乎所有软件相关文件都在/home/[软件名]目录下



```
38 ./configure --prefix=/home/bfs/nginx
39 make
40 make install
```

- 软件找到所需的动态链接库
 - 大部分编译安装的软件会在configure步骤时检查所需的运行环境、依赖软件
 - RPM包安装的软件在安装时会检查依赖的软件的安装情况
 - error while loading shared libraries:
libnet.so.1:cannot open shared object file :No such file or directory

- Linux下默认的动态链接库搜索目录为/lib和/usr/lib
- 但某些编译安装的软件会把自己编译生成的动态链接库放到/usr/local/lib、/usr/local/lib64……
 - 少数软件直接将动态链接库放在可执行文件所在的目录下
- 此时运行软件会因为找不到需要的动态链接库从而报错
- 解决方法
 - 直接将动态链接库放到正确的目录
 - find / -name [报错提示的库文件]
 - 通过cp或ln命令在/lib和/usr/lib生成复制或者链接
 - 将存放动态链接库的文件夹添加到系统搜索路径下
 - 修改LD_LIBRARY_PATH环境变量（重启失效）
 - 修改/etc/ld.so.conf文件、修改/etc/ld.so.conf.d/目录下文件、在/etc/ld.so.conf.d/目录下新建文件，最后执行ldconfig
 - google

- 开机自启动
 - 不同Linux发行版的开机自启动配置方法不同
 - 一些开机自启动方法依赖于特定软件
 - chkconfig, CentOS自带, ubuntu需自行安装
 - 在/etc/rc.local、/etc/rc.d/rc.local文件中加入启动软件的命令
- 使用不同的开机自启动方法在细节效果上有差异
 - 有的在用户登录后才启动（~.bash_profile、/etc/profile.d）
 - 有的执行后终端结束，若需要软件常驻后台需在执行命令后加上 &（/xxx/[可执行文件名] &）
 - 有的执行时当前路径在/root，有的在/home有的不知道在哪里……
 - 通过开机自启动的效果和手动执行不一样（执行用户不同）
 - 有的就是死活没法自启动

yum、apt-get的使用



yum、apt-get的使用

- yum和apt-get分别是CentOS和ubuntu下比较常用的RPM包安装工具
 - 提供在线安装能力
 - 能够方便的安装、卸载、升级、查询安装的软件
- yum和apt-get在使用中主要包括配置源和敲命令2个部分

- 源的配置文件
 - yum: `/etc/yum.conf/CentOS-Base.repo`
 - apt-get: `/etc/apt/sources.list`
- 修改方法
 - 配置源最好的方法是从网上查询、下载对应源配置文件，然后覆盖系统初始的源配置文件
 - `http://mirrors.aliyun.com/repo/Centos-7.repo`
 - 手动修改文件，将文件中所有域名相关的部分替换为想要更换的源的地址
 - `http://mirror.bit.edu.cn/centos/$releasever/os/$basearch/`
 - 记得备份原始配置文件
- 本地源
 - 将系统安装镜像内文件拷贝到本地并修改源配置文件
 - 在本机或局域网内搭建一个有源功能的web服务器

- yum的使用

- yum [动作] [软件名] [参数]

- 动作

- install [软件名]、groupinstall [组名]

- list [软件名]、list installed、deplist [软件名]、info [软件名]

- **update**、update [软件名]、check-update

- remove、clean

- 参数

- -y 自动选择yes

- 在有可视化界面的情况下，可以安装yumex对软件进行可视化管理

yum、apt-get的使用



Yum Extender

文件 (F) 编辑 (E) 显示 (V) 配置 (O) 帮助 (H)

软件包

更新 (P) 可利用 (B) 安装 (I) 所有 软件组 (G) 分类 (C)

软件包	版本	构架	概要	软件库	大小
<input type="checkbox"/> ModemManager	1.6.0-2.el7	x86_64	Mobile broadband modem management service	base	756 k
<input type="checkbox"/> ModemManager-glib	1.6.0-2.el7	x86_64	Libraries for adding ModemManager support to applications th	base	232 k
<input type="checkbox"/> NetworkManager	1.8.0-11.el7_4	x86_64	Network connection manager and user applications	updates	1.6 M
<input type="checkbox"/> NetworkManager	1.8.0-11.el7_4	x86_64	Network connection manager and user applications	updates	1.6 M
<input type="checkbox"/> NetworkManager	1.8.0-11.el7_4	x86_64	Network connection manager and user applications	updates	1.6 M
<input type="checkbox"/> NetworkManager-adsl	1.8.0-11.el7_4	x86_64	ADSL device plugin for NetworkManager	updates	153 k
<input type="checkbox"/> NetworkManager-bluetooth	1.8.0-11.el7_4	x86_64	Bluetooth device plugin for NetworkManager	updates	171 k
<input type="checkbox"/> NetworkManager-glib	1.8.0-11.el7_4	x86_64	Libraries for adding NetworkManager support to applications (updates	1.1 M
<input type="checkbox"/> NetworkManager-libreswan	1.2.4-2.el7	x86_64	NetworkManager VPN plug-in for IPsec VPN	base	112 k
<input type="checkbox"/> NetworkManager-ppp	1.8.0-11.el7_4	x86_64	PPP plugin for NetworkManager	updates	157 k

<http://blog.csdn.net/zhaoyanjun6>

- apt-get的使用
 - apt-get [动作] [软件名] [参数]

```
update - 取回更新的软件包列表信息
upgrade - 进行一次升级
install - 安装新的软件包 (注: 软件包名称是 libc6 而非 libc6.deb)
remove - 卸载软件包
autoremove - 卸载所有自动安装且不再使用的软件包
purge - 卸载并清除软件包的配置
source - 下载源码包文件
build-dep - 为源码包配置所需的编译依赖关系
dist-upgrade - 发布版升级, 见 apt-get(8)
dselect-upgrade - 根据 dselect 的选择来进行升级
clean - 删除所有已下载的包文件
autoclean - 删除已下载的旧包文件
check - 核对以确认系统的依赖关系的完整性
changelog - 下载指定软件包, 并显示其 changelog
download - 下载指定的二进制包到当前目录
```



其他

- 部分小型软件不需要“安装”过程，直接./运行即可，这种软件本质上属于源码编译安装的特例
- 部分复杂但常用软件有“脚本安装”例如lamp（linux-apache-mysql-php环境），其本质是通过脚本的方式将软件安装的各个步骤串联，并提供了更为人性化和方便的交互过程
- 安装Linux时磁盘配置过程中会让选择是否启用LVM，启用LVM后如果需要对硬盘进行扩容会更加方便

- <https://www.cnblogs.com/hanson1/articles/7105223.html>
- <https://www.cnblogs.com/tswhq/p/7868314.html>
- <https://segmentfault.com/a/1190000011325357>
- <https://www.cnblogs.com/yangp/p/8506264.html>

谢谢!

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。

