软件管理

一、软件分类

1. **源码包（Source Package）**
   - **特点**：包含原始代码（C/C++等），需手动编译安装。
   - **常见格式**：`.tar.gz`、`.tar.bz2`、`.tar.xz`。
   - **适用场景**：需要最新版本、定制功能或特定编译选项时使用。
2. **封装包（预编译包）**
   - **特点**：已编译为二进制文件，直接安装。
   - 常见格式：
     - **RPM包**：`.rpm`（Red Hat/CentOS/Fedora manager）。
     - **DEB包**：`.deb`（Debian/Ubuntu）。
   - **适用场景**：快速部署稳定版本，依赖自动管理。

二、源码包安装核心知识点

1. **安装流程**

   bash

   ```
   # 解压 → 配置 → 编译 → 安装
   tar -zxf package.tar.gz
   cd package
   ./configure --prefix=/usr/local/webserver
   make
   make install
   ```

2. **关键步骤解析**
   - `./configure`：
     - 检查系统环境（如编译器、依赖库）。
     - 生成 `Makefile` 文件（编译规则）。
     - 常用参数：`--prefix`（指定安装路径）。
   - `make`：根据 `Makefile` 编译源码为二进制文件。
   - `make install`：将编译后的文件复制到指定路径。
3. **路径管理**
   - 源码包默认安装到 `/usr/local/`（可通过 `--prefix` 自定义）。
   - 重要目录：
     - `/usr/local/bin`：可执行文件。
     - `/usr/local/etc`：配置文件。
     - `/usr/local/lib`：库文件。

实验：安装apache 2.2.15

1.将压缩包拷贝到虚拟机中

```
[root@xnha ~]# ls
公共  图片  音乐              file111.txt  httpd-2.2.15          test123
模板  文档  桌面              file1.txt    httpd-2.2.15.tar.gz   test321.tar
视频  下载  anaconda-ks.cfg  file.txt     initial-setup-ks.cfg
```

2.解压 http-2.2.15.tar.gz

```
root@xnha ~]# tar -zxf httpd-2.2.15.tar.gz
[root@xnha ~]# ls
公共  图片  音乐              file111.txt  httpd-2.2.15          test123
模板  文档  桌面              file1.txt    httpd-2.2.15.tar.gz   test321.tar
视频  下载  anaconda-ks.cfg  file.txt     initial-setup-ks.cfg
[root@xnha ~]#
```

3.查看解压的目录

目录中还有不同的目录，不同目录里头有对应的c文件，都是作者写的源码

```
[root@xnha ~]# ls httpd-2.2.15
ABOUT_APACHE  config.layout  INSTALL         NOTICE            srclib
acinclude.m4  configure      InstallBin.dsp  NWGNUmakefile     support
Apache.dsw    configure.in   LAYOUT          os                test
build         docs           libhttpd.dsp    README            VERSIONING
BuildAll.dsp  emacs-style    LICENSE         README.platforms
BuildBin.dsp  httpd.dsp      Makefile.in     README-win32.txt
buildconf     httpd.spec     Makefile.win    ROADMAP
CHANGES       include        modules         server
```

4. 查看帮助手册和安装手册

   readme和install

   ```
   [root@xnha httpd-2.2.15]# less README
   [root@xnha httpd-2.2.15]# less INSTALL
   ```

```
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)

                        Apache HTTP Server

What is it?
-----------

The Apache HTTP Server is a powerful and flexible HTTP/1.1 compliant
web server.  Originally designed as a replacement for the NCSA HTTP
Server, it has grown to be the most popular web server on the
Internet.  As a project of the Apache Software Foundation, the
developers aim to collaboratively develop and maintain a robust,
commercial-grade, standards-based server with freely available
source code.

The Latest Version
------------------

Details of the latest version can be found on the Apache HTTP
server project page under <http://httpd.apache.org/>.

Documentation
-------------

The documentation available as of the date of this release is
included in HTML format in the docs/manual/ directory.  The most
up-to-date documentation for the 2.2.x releases can be found at
```

5. 按照安装手册命令依次进行

`./configure`

```
[root@xnha httpd-2.2.15]# ./configure --prefix=/usr/local/webserver   #--prefix= 定义路径      过程中会检查操作系统是否具备安装插件的功能   并生成安装的编译顺序配置文件
```

```
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
creating config_vars.mk
configure: creating ./config.status
creating modules/aaa/Makefile
creating modules/arch/win32/Makefile
creating modules/cache/Makefile
creating modules/database/Makefile
creating modules/debug/Makefile
creating modules/echo/Makefile
creating modules/experimental/Makefile
creating modules/filters/Makefile
creating modules/ldap/Makefile
creating modules/loggers/Makefile
creating modules/metadata/Makefile
creating modules/proxy/Makefile
creating modules/ssl/Makefile
creating modules/test/Makefile
creating os/unix/Makefile
creating server/mpm/Makefile
creating server/mpm/prefork/Makefile
creating modules/http/Makefile
creating modules/dav/main/Makefile
creating modules/generators/Makefile
creating modules/dav/fs/Makefile
creating modules/dav/lock/Makefile
creating modules/mappers/Makefile
creating Makefile
creating modules/Makefile
creating srclib/Makefile
```

编译顺序配置文件

`make`根据生成出的makefile文件开始编译

```
[root@xnha httpd-2.2.15]# make
```

```
[root@xnha httpd-2.2.15]# make
Making all in srclib
make[1]: 进入目录 `/root/httpd-2.2.15/srclib'
Making all in apr
make[2]: 进入目录 `/root/httpd-2.2.15/srclib/apr'
make[3]: 进入目录 `/root/httpd-2.2.15/srclib/apr'
/bin/sh /root/httpd-2.2.15/srclib/apr/libtool --silent --mode=compile gcc -g -O2 -pthre
ad   -DHAVE_CONFIG_H -D_REENTRANT -D_GNU_SOURCE   -I./include -I/root/httpd-2.2.15/srcl
ib/apr/include/arch/unix -I./include/arch/unix -I/root/httpd-2.2.15/srclib/apr/include/
arch/unix -I/root/httpd-2.2.15/srclib/apr/include  -o passwd/apr_getpass.lo -c passwd/a
pr_getpass.c && touch passwd/apr_getpass.lo
/bin/sh /root/httpd-2.2.15/srclib/apr/libtool --silent --mode=compile gcc -g -O2 -pthre
ad   -DHAVE_CONFIG_H -D_REENTRANT -D_GNU_SOURCE   -I./include -I/root/httpd-2.2.15/srcl
ib/apr/include/arch/unix -I./include/arch/unix -I/root/httpd-2.2.15/srclib/apr/include/
arch/unix -I/root/httpd-2.2.15/srclib/apr/include  -o strings/apr_cpystrn.lo -c strings
/apr_cpystrn.c && touch strings/apr_cpystrn.lo
/bin/sh /root/httpd-2.2.15/srclib/apr/libtool --silent --mode=compile gcc -g -O2 -pthre
ad   -DHAVE_CONFIG_H -D_REENTRANT -D_GNU_SOURCE   -I./include -I/root/httpd-2.2.15/srcl
ib/apr/include/arch/unix -I./include/arch/unix -I/root/httpd-2.2.15/srclib/apr/include/
arch/unix -I/root/httpd-2.2.15/srclib/apr/include  -o strings/apr_strtok.lo -c strings/
```

`make install`将刚才编译的文件拷贝到安装路径下

```
[root@xnha httpd-2.2.15]# ls /usr/local/webserver/
bin    cgi-bin  error   icons    lib   man      modules
build  conf     htdocs  include  logs  manual
```

运行服务`apachectl start`

```
[root@xnha webserver]# cd bin
[root@xnha bin]# apachectl start
bash: apachectl: 未找到命令...
文件搜索失败: Cannot update read-only repo
[root@xnha bin]# ls
ab            apu-1-config  dbmmanage    htcacheclean  htpasswd   logresolve
apachectl     apxs          envvars      htdbm         httpd      rotatelogs
apr-1-config  checkgid      envvars-std  htdigest      httxt2dbm
[root@xnha bin]# ./apachectl start

httpd: Could not reliably determine the server's fully qualified domain name,
using fe80::ac23:8439:4329:9fe2 for ServerName
```

三、封装包（RPM/DEB）管理工具

1. **RPM包管理（Red Hat系）**
   - **安装**：`rpm -ivh package.rpm`
   - **查询**：`rpm -q package`（查看是否安装）。
   - **卸载**：`rpm -e package`。
   - **依赖管理**：`yum install package`（自动解决依赖）。
2. **DEB包管理（Debian系）**
   - **安装**：`dpkg -i package.deb`
   - **查询**：`dpkg -l | grep package`。
   - **卸载**：`dpkg -r package`。
   - **依赖管理**：`apt install package`（自动解决依赖）。

四、源码包 vs 二进制包对比

对比项	源码包	二进制包（RPM/DEB）
灵活性	高（可自定义编译选项）	低（使用预编译配置）
安装速度	慢（需编译）	快（直接安装）
依赖管理	手动解决	自动解决
更新频率	快（紧跟上游发布）	慢（需打包者适配）
适用场景	开发者、需定制功能	快速部署稳定版本

六、维护与管理

1. **源码包卸载**
   - 直接删除安装目录：`rm -rf /安装路径`。
   - 若配置了系统服务，需清理服务文件（如 `/etc/systemd/system/`）。
2. **服务管理**
   - **启动/停止服务**：通过安装目录中的控制脚本（如 `apachectl start`）.
   - **注册系统服务**：创建 `systemd` 配置文件（`.service` 文件）。