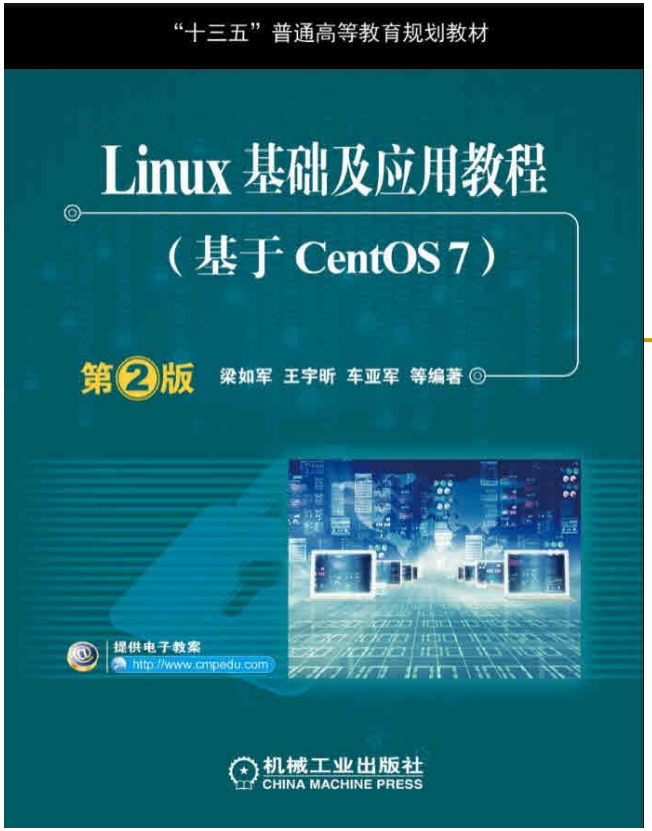


第12章 FTP服务和NFS服务

主讲人： 梁如军

2015-05-05

本章内容要点

- FTP的相关概念
- 配置vsftpd服务器
- NFS的相关概念
- 配置NFS服务器

梁如军（linuxbooks@126.com）

本章学习目标

- 理解FTP协议模型
- 掌握FTP的数据传输模式及使用场合
- 学会配置各种FTP服务器
- 理解NFS和RPC的关系
- 学会配置NFS目录共享
- 掌握NFS相关工具的使用
- 学会挂装NFS文件系统

FTP服务

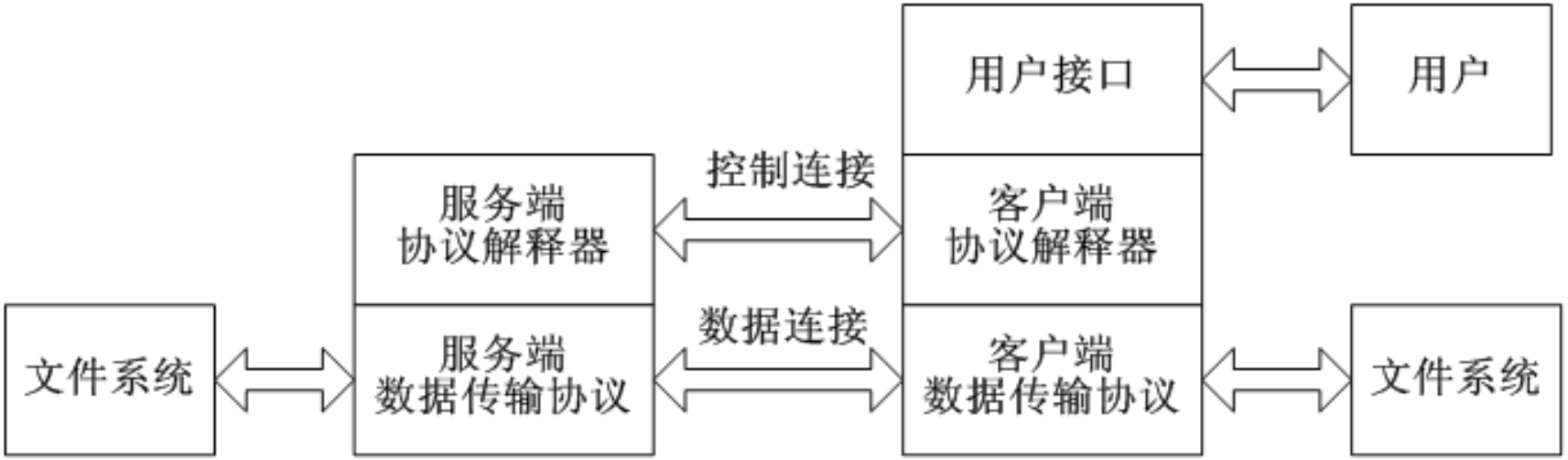
梁如军（linuxbooks@126.com）

FTP协议

- 文件传输协议（File Transfer Protocol，FTP）标准是在RFC959说明的。
- 协议定义了一个在远程计算机系统和本地计算机系统之间传输文件的一个标准。
- FTP运行在OSI模型的应用层， 并利用传输控制协议TCP在不同的主机之间提供可靠的数据传输。
- FTP在文件传输中还支持断点续传功能，可以大幅度地减小CPU和网络带宽的开销。

梁如军（linuxbooks@126.com）

FTP协议模型



梁如军（linuxbooks@126.com）

FTP协议模型（续）

- 用户接口（UI）：提供了一个用户接口并使用客户端协议解释器的服务
- 客户端协议解释器（CPI）：向远程服务器协议机发送命令并且驱动客户数据传输过程
- 服务端协议解释器（SPI）：响应客户协议机发出的命令并驱动服务器端数据传输过程
- 客户端数据传输协议（CDTP）：负责完成和服务器数据传输过程及客户端本地文件系统的通信
- 服务端数据传输协议（SDTP）：负责完成和客户数据传输过程及服务器端文件系统的通信

梁如军（linuxbooks@126.com）

FTP运行原理——两种连接

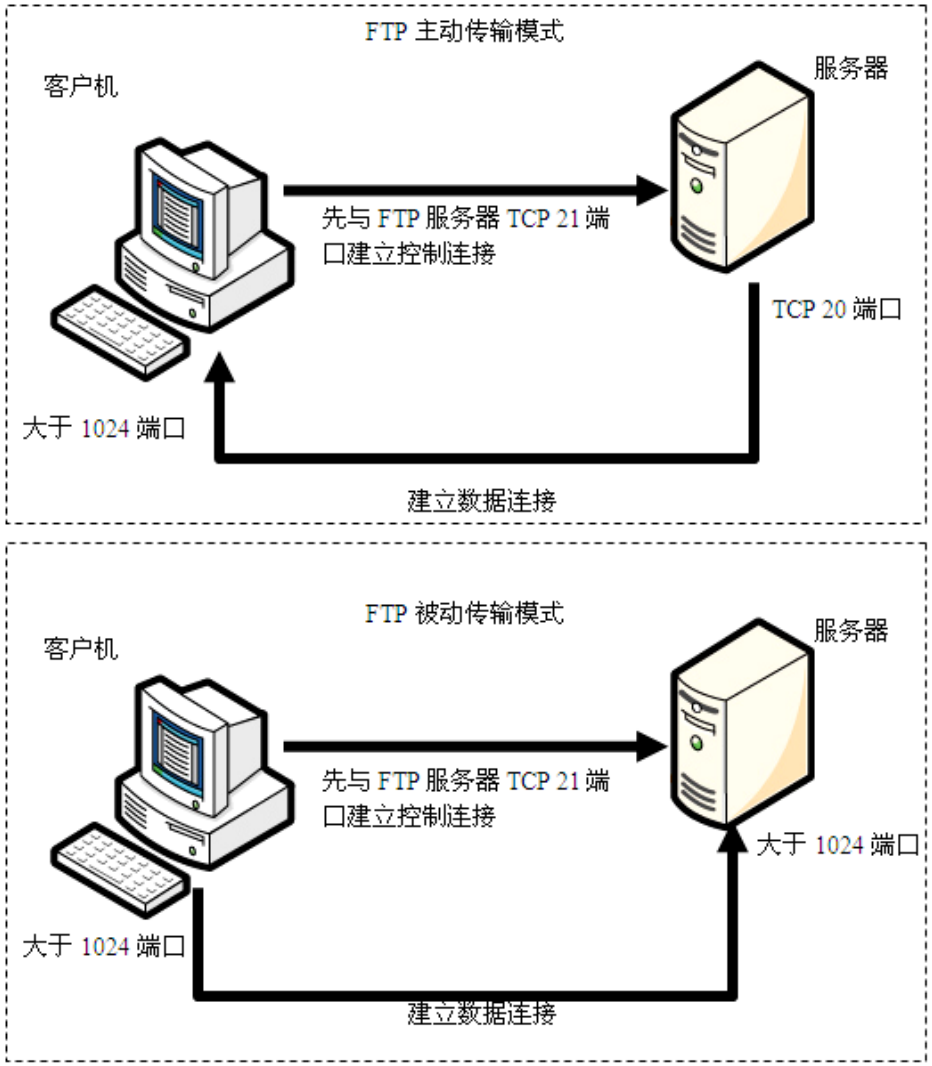
- FTP会话存在有两个独立的TCP连接
  - 由CPI和SPI使用的，被称作控制连接（control connection）
  - 由CDTP和SDTP使用的，被称作数据连接（data connection）
- 两个连接可以选择不同的合适服务质量。如：对控制连接来说需要更小的延迟时间， 对数据连接来说需要更大的数据吞吐量； 而且可以避免实现数据流中的命令的通明性及逃逸。

FTP运行原理——控制连接

- 控制连接主要用来传送在实际通信过程中需要执行的FTP命令以及命令的响应。
- 控制连接只需要很小的网络带宽。
- FTP服务器监听端口号21来等待控制连接建立请求。
- 控制连接建立以后并不立即建立数据连接，而是服务器通过一定的方式来验证客户的身份，以决定是否可以建立数据传输。
- 数据连接是等到要目录列表、传输文件时才临时建立的， 并且每次客户端使用不同的端口号来建立数据连接。 一旦数据传输完毕，就中断这条临时的数据连接。
- 在FTP连接期间，控制连接始终保持通畅的连接状态。 在数据连接存在期间内，控制连接肯定是存在的； 一旦控制连接断开，数据连接会自动关闭。

梁如军（linuxbooks@126.com）

主动模式和被动模式



梁如军（linuxbooks@126.com）

FTP的数据传输模式（1）

- 主动传输模式 （Active FTP）
  - FTP的数据连接和控制连接的方向是相反的。 也就是说，是服务器向客户端发起一个用于数据传输的连接。 客户端的连接端口是由服务器端和客户端通过协商确定的。
  - FTP客户端随机开启一个大于1024的端口N向服务器的21号端口发起连接， 然后开放N+1号端口进行监听，并向服务器发出PORT N+1命令。
  - 服务器接收到命令后，会用其本地的FTP数据端口（通常是20） 来连接客户端指定的端口N+1，进行数据传输。

梁如军（linuxbooks@126.com）

FTP的数据传输模式（2）

- 被动传输模式 （Passive FTP）
  - FTP的数据连接和控制连接的方向是一致的。 也就是说，是客户端向服务器发起一个用于数据传输的连接。 客户端的连接端口是发起这个数据连接请求时使用的端口号。
  - FTP客户端随机开启一个大于1024的端口N向服务器的21号端口发起连接， 同时会开启N+1号端口。然后向服务器发送PASV命令，通知服务器自己处于被动模式。
  - 服务器收到命令后，会开放一个大于1024的端口P进行监听，然后用PORT P 命令通知客户端，自己的数据端口是P。
  - 客户端收到命令后，会通过N+1号端后连接服务器的端口P，然后在两个端口之间进行数据传输。
  - 被动模式的FTP通常用在处于防火墙之后的FTP客户访问外界FTP服务器的情况。

梁如军（linuxbooks@126.com）

FTP服务的使用者（1）

- 本地用户（real用户）
  - 本地用户既可以登录Shell，又可以FTP登录。
  - 本地用户可以通过输入自己的账号和口令来进行授权登录。
  - 当授权访问的本地用户登录系统后，其登录目录为用户自己的家目录（$HOME）。
  - 本地用户既可以下载又可以上传。

梁如军（linuxbooks@126.com）

FTP服务的使用者（2）

- 虚拟用户（guest用户）
  - 如果用户在远程FTP服务器上拥有账号， 且此账号只能用于文件传输服务，则称此用户为虚拟用户或Guest用户。
  - 虚拟用户可以通过输入自己的账号和口令来进行授权登录。
  - 当授权访问的虚拟用户登录系统后，其登录目录为服务器为其指定的目录。
  - 通常情况下，虚拟用户既可以下载又可以上传。

梁如军（linuxbooks@126.com）

FTP服务的使用者（3）

- 匿名用户（anonymous用户）
  - 如果用户在远程FTP服务器上没有账号，则称此用户为匿名用户。
  - 若FTP服务器提供匿名访问功能，则匿名用户可以通过输入账号（anonmous或ftp） 和口令（用户自己的E-Mail地址）来进行登录。
  - 当匿名用户登录系统后，其登录目录为匿名FTP服务器的根目录（/var/ftp）。
  - 一般情况下匿名FTP服务器只提供下载功能，不提供上传服务或者使上传受到一定的限制。

梁如军（linuxbooks@126.com）

Vsftpd简介

- 是一个安全、高速、稳定的FTP服务器。
- 可设定多个基于IP的虚拟FTP服务器。
- 匿名FTP服务更是十分容易。
- 不执行任何外部程序，从而减少了安全隐患。
- 支持虚拟用户，且支持每个虚拟用户具有独立的配置。
- 可以设置为从xinetd启动，或者是独立ftp服务器两种运行方式。
- 支持PAM 或 xinetd / tcp_wrappers的认证方式。
- 支持带宽限制等。

vsftpd的安装和启动

- vsftpd的安装

  # yum install vsftpd

- 管理vsftpd服务

  # systemctl {start|stop|status|restart} vsftpd

  # systemctl {enable|disable} vsftpd

  - 管理基于不同配置文件的多个vsftpd服务

  # systemctl {start|stop|status|restart} vsftpd.target

梁如军（linuxbooks@126.com）

CentOS下的vsftpd服务概览

- 软件包：vsftpd
- 服务类型：由Systemd启动的守护进程
- 配置单元： /usr/lib/systemd/system/vsftpd.service
- 守护进程：/usr/sbin/vsftpd
- 端口：21 (ftp), 20 (ftp-data)
- 配置文件
  - 主配置文件：/etc/vsftpd/vsftpd.conf
  - 用户访问控制配置文件：/etc/vsftpd/{ftpusers, user_list}
  - PAM配置文件：/etc/pam.d/vsftpd
- 相关软件包和内核模块：
  - tcp_wrappers
  - ip_conntrack_ftp, ip_nat_ftp

梁如军（linuxbooks@126.com）

vsftpd 默认的主配置文件

- 允许匿名用户和本地用户登录
- 匿名用户的登录名为 ftp 或 anonymous，口令为一个Email地址
- 匿名用户不能离开匿名服务器目录/var/ftp，且只能下载不能上传
- 本地用户的登录名为本地用户名，口令为此本地用户的口令
- 本地用户可以离开自家目录切换至有权访问的其他目录，并在权限允许的情况下进行上传/下载
- 写在文件/etc/vsftpd/ftpusers中的本地用户禁止登录
- 要使用户在下载文件时能够续传文件，必须保证文件对其他用户有读的权限。否则，当续传时不能读取已传的服务器上的文件

梁如军（linuxbooks@126.com）

vsftpd配置文件的常用参数1

- 设置空闲的用户会话的中断时间
  - **idle_session_timeout=600**
- 设置空闲的数据连接的的中断时间
  - **data_connection_timeout=120**
- 设置客户端空闲时的自动中断/激活连接的时间
  - **connect_timeout=60**
    - 客户端空闲1分钟后自动中断连接
  - **accept_timeout=60**
    - 客户端中断1分钟后自动激活连接

梁如军（linuxbooks@126.com）

vsftpd配置文件的常用参数2

- 关于被动模式的数据连接
  - **pasv_enable=Yes**
  - **pasv_min_port=50000**
  - **pasv_max_port=60000**
- 设置用户类型的访问
  - **local_enable=<YES/NO>**
  - **guest_enable=<YES/NO>**
  - **anonymous_enable=<YES/NO>**

梁如军（linuxbooks@126.com）

vsftpd配置文件的常用参数3

- 设置chroot

	chroot_local_user=YES	chroot_local_user=NO
chroot_list_enable=YES	1.所有用户都被限制在其主目录下 2.使用chroot_list_file指定的用户列表，这些用户作为“例外”，不受限制	1.所有用户都不被限制其主目录下 2.使用chroot_list_file指定的用户列表，这些用户作为“例外”，受到限制
chroot_list_enable=NO	1.所有用户都被限制在其主目录下 2.不使用chroot_list_file指定的用户列表，没有任何“例外”用户	1.所有用户都不被限制其主目录下 2.不使用chroot_list_file指定的用户列表，没有任何“例外”用户

梁如军（linuxbooks@126.com）

vsftpd配置文件的常用参数4

- vsftpd匿名用户上传配置
  - **anon_upload_enable = Yes**
  - **anon_mkdir_write_enable = Yes**
  - **anon_world_readable_only = No**
  - **anon_other_write_enable = Yes**
- 注意
  - anon_upload_enable=YES 仅能上传。
  - anon_mkdir_write_enable=YES 仅能创建目录。
  - anon_other_write_enable=YES 同时开放文件更名、删除文件等权限。

梁如军（linuxbooks@126.com）

vsftpd配置文件的常用参数5

- 配置最大传输速率限制
  - **local_max_rate**
  - **anon_max_rate**
- 每客户和最大的连接数限制
  - **max_per_ip**
  - **max_clients**

梁如军（linuxbooks@126.com）

vsftpd配置文件的常用参数6

- 限制指定的本地用户不能访问，而其他本地用户可访问
  - **userlist_enable=YES**
  - **userlist_deny=YES**
  - **userlist_file= /etc/vsftpd/user_list**
- 限制指定的本地用户可以访问，而其他本地用户不可访问
  - **userlist_enable= YES**
  - **userlist_deny= NO**
  - **userlist_file= /etc/vsftpd/user_list**

梁如军（linuxbooks@126.com）

使用vsftpd的分离配置文件

- 对不同的本地用户实施不同的配置
  - **user_config_dir=/etc/vsftpd/userconf/**
- 对不同的 主机/网络 实施不同的配置
  - 主配置文件
    - **tcp_wrappers=YES**
  - tcp_wrappers的配置文件
    - **vsftpd: 主机表: setenv VSFTPD_LOAD_CONF 配置文件**

梁如军（linuxbooks@126.com）

配置vsftpd服务器

- 配置高安全级别的匿名服务器
- 配置允许匿名用户上传的FTP服务器
- 将本地用户限制在其自家目录中
- 对不同的本地用户实施不同的配置
- 配置基于本地用户的访问控制
- 配置基于主机的访问控制
- 对不同的主机/网络的访问实施不同的配置
- 配置使用虚拟用户的FTP服务
- 配置基于ssl的ftp服务

请参考教材的操作步骤

梁如军（linuxbooks@126.com）

NFS服务

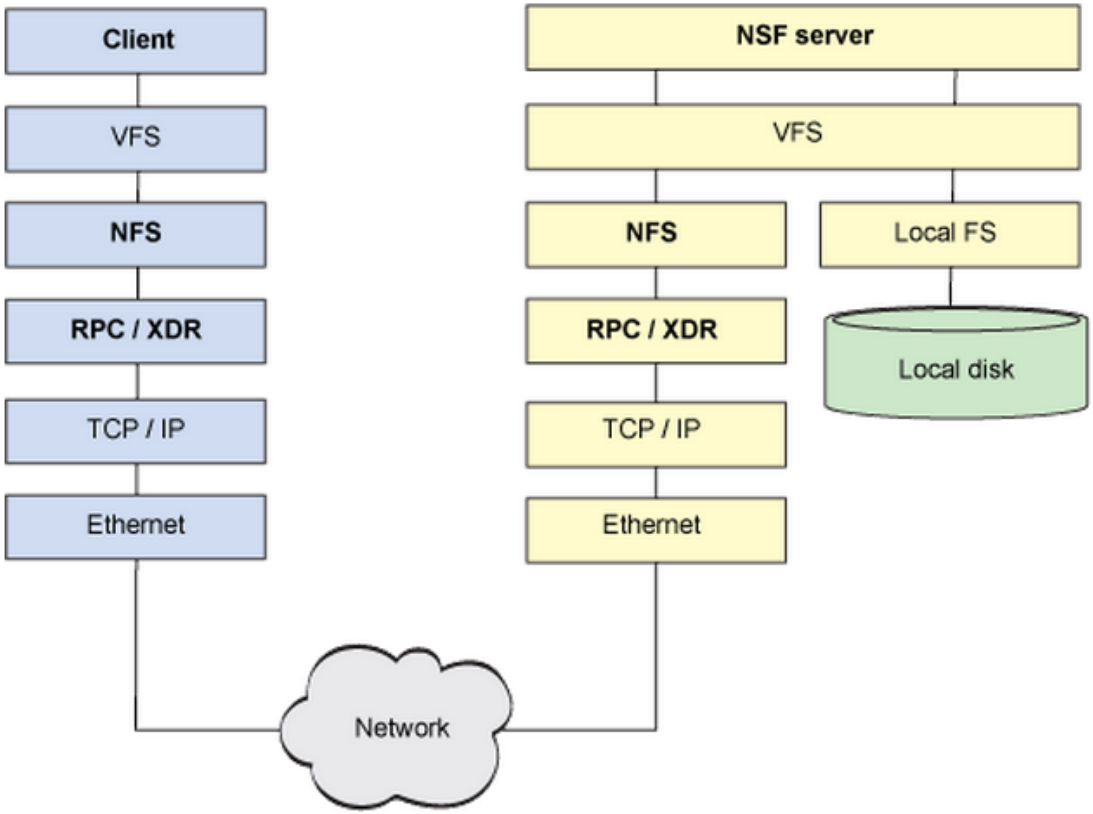
梁如军（linuxbooks@126.com）

NFS简介

- 网络文件系统（Network File System, NFS）采用客户/服务器工作模式。
- NFS是分布式计算系统的一个组成部分，可实现在异种网络上共享和装配远程文件系统。
- NFS提供了一种在类UNIX系统上共享文件的方法。
- NFS还可以结合远程网络启动实现
  - 无盘工作站（PXE启动系统，所有数据均在服务器的磁盘阵列上）
  - 瘦客户工作站（本地启动系统，本地磁盘存储了常用的系统工具，而所有/home目录的用户数据被放在NFS服务器上并且在网络上处处可用）

NFS协议模型

- NFS协议提供了一种远程文件系统规范



NFS协议版本

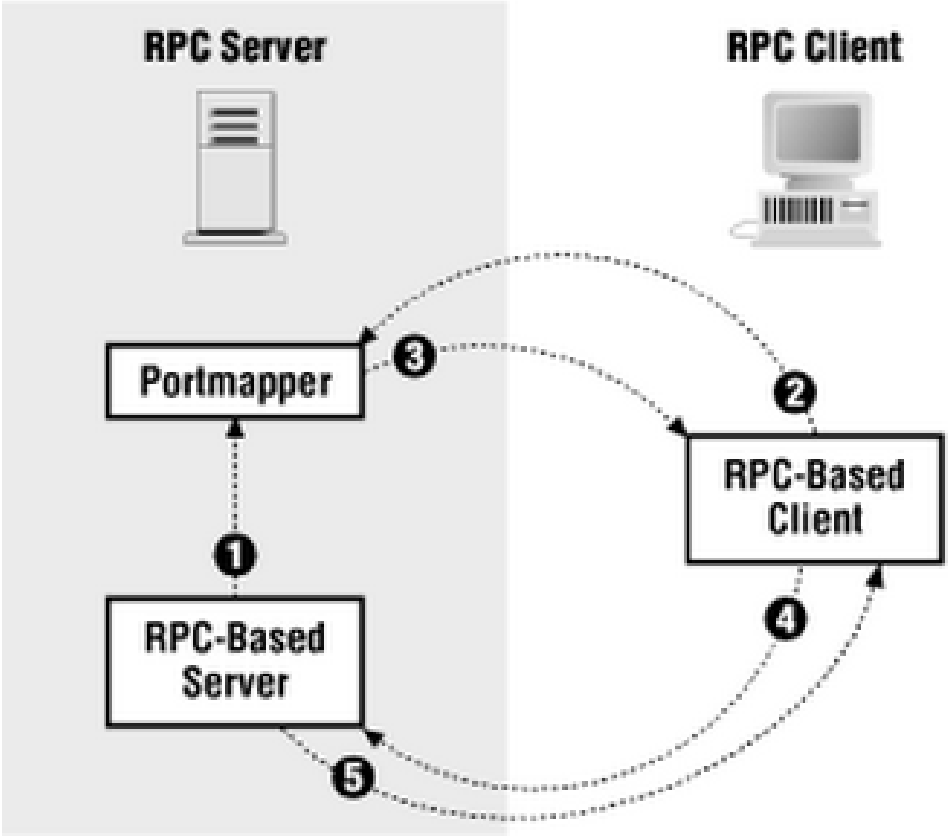
协议版本	说　明	与RPC协同	传输协议	RPC标准
NFS V2	诞生于20世纪80年代的协议标准	需要	UDP	RFC1094
NFS V3	具有更好的可扩展性、支持大文件(超过2GB)、异步写入以及使用TCP传输协议	需要	TCP/UDP	RFC1813
NFS V4	内置了远程挂装和文件锁定协议支持，支持通过 kerberos 进行安全用户身份验证	无需	TCP	RFC 3530
NFS V4.1	支持更高扩展性和更高性能的并行NFS（pNFS）	无需	TCP	RFC 5661

- RHEL /CentOS 7 支持NFS V3、NFS V4客户端，默认使用NFS V4协议。

RPC

- NFS v3协议本身并没有网络传输功能，而是基于远程过程调用（Remote Procedure Call，RPC）协议实现的
- RPC提供了一个面向过程的远程服务的接口。
- RPC可以通过网络从远程主机程序上请求服务，而不需要了解底层网络技术的协议。
- RPC工作在OSI模型的会话层，它可以为遵从RPC协议应用层协议提供端口注册功能。
- 事实上，有很多服务（NFS和NIS等）都可以向RPC注册端口。
- RPC 使用网络端口111 来监听客户端的请求。

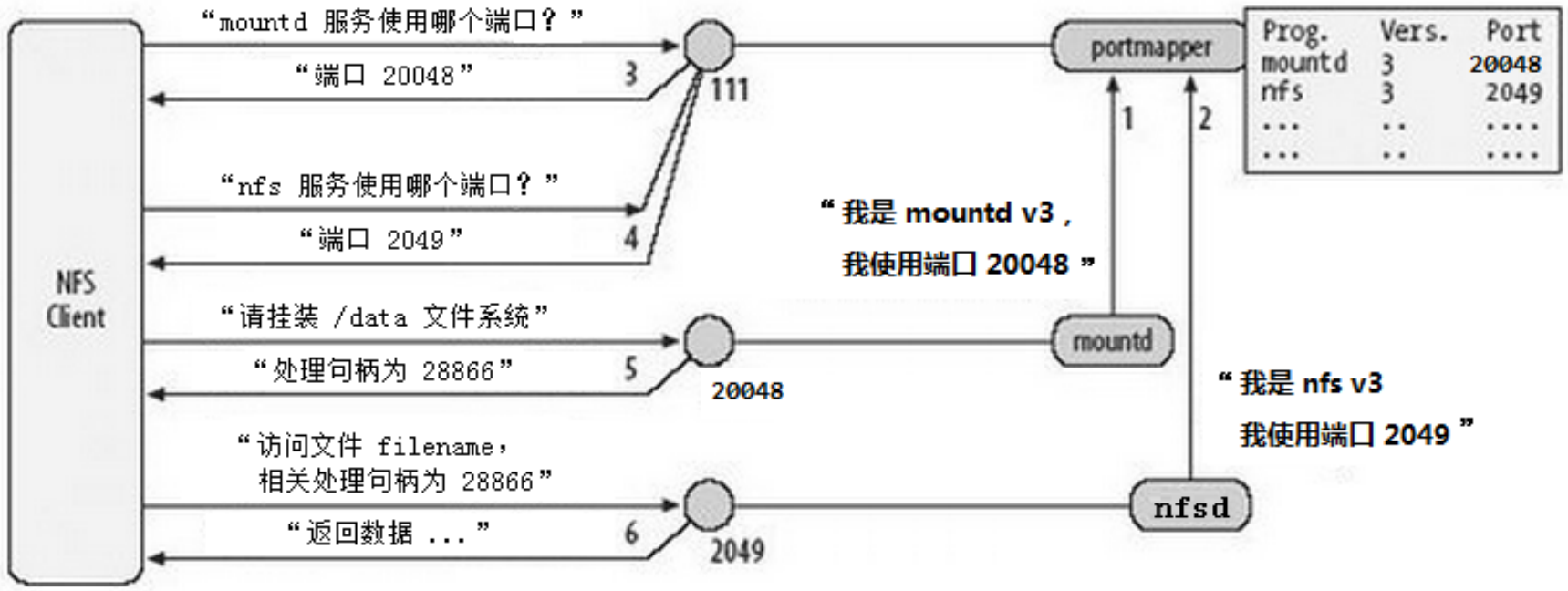
RPC协议模型



1. 基于 RPC 的服务在启动时向 Portmapper 注册端口
2. 基于 RPC 的客户端联系服务器端 Portmapper 询问某服务的端口号
3. Portmapper 告知客户端某基于 RPC 服务使用的端口号
4. 基于 RPC 的客户端访问被告知的某基于 RPC 服务的端口号
5. 基于 RPC 的服务端响应客户端请求

梁如军（linuxbooks@126.com）

NFS V3与 RPC



- Portmapper服务（rpcbind）监听在 111端口

梁如军（linuxbooks@126.com）

NFS v3的守护进程

- NFS v3的不同功能由不同的守护进程提供。
- NFS v3的每个功能就会由RPC固定或随机分配的端口进行监听。

- **rpc.nfsd**：基本的NFS守护进程（2049端口），主要负责登录权限检测。
- **rpc.mountd**：负责管理NFS的文件系统，对客户端存取服务器的文件进行一系列的管理。
- **rpc.rquotad**：提供远程磁盘限额服务。
- **rpc.lockd**：用于管理文件的锁定，防止多个客户端同时写入某个文件时产生的冲突。
- **rpc.statd**：用来检查共享目录的一致性。

梁如军（linuxbooks@126.com）

NFS v4

- NFS v4内置了远程挂装和文件锁定等协议支持，因此NFSv4不再需要与rpcbind、rpc.mountd、rpc.statd和lockd互动。
- 在 CentOS 7 中，当 NFS 服务器端使用 exportfs 命令时仍然需要 rpc.mountd 守护进程，但它不参与跨越网络线路的操作。
- NFSv4的nfs服务仍然监听在tcp:2049端口。

RHEL/CentOS 7下的NFS

- 与NFS相关的RPM包
  - **nfs-utils**： NFS 的主要组件。包含有 rpc.nfsd 及 rpc.mountd 这两个NFS的核心守护进程及其相关文档、执行文件等
  - **rpcbind**：提供RPC的端口映射的守护进程及其相关文档、执行文件等
- 与NFS相关的工具
  - **exportfs**：在NFS服务器端，维护 NFS 共享资源的命令
  - **showmount**：用来在NFS客户端查看服务器共享的目录
  - **nfsstat**：显示NFS的状态统计信息
  - **rpcinfo**：显示由RPC维护的端口映射，显示已经注册的RPC服务列表。

梁如军（linuxbooks@126.com）

安装与启动NFS

- 安装
  - **# yum install nfs-utils rpcbind**
- 开机启动
  - **# systemctl enable rpcbind**
  - **# systemctl enable nfs-server**
- 启动
  - **# systemctl start rpcbind**
  - **# systemctl start nfs**

梁如军（linuxbooks@126.com）

NFS服务概览

- 软件包：nfs-utils
- 服务类型：由Systemd启动的守护进程
- 配置单元： /usr/lib/systemd/system/nfs.service
- 守护进程：rpc.nfsd, rpc.mountd, ……
- 端口：2049(nfsd), 其它端口由rpcbind(111)分配
- 配置文件：/etc/exports
- 相关软件包：rpcbind（必须）、tcp_wrappers

梁如军（linuxbooks@126.com）

NFS服务端

- 主配置文件 /etc/exports
- exportfs命令
- 配置NFS服务固定端口

NFS主配置文件 /etc/exports

共享目录 [主机表1（参数项）] [主机表2（参数项）]

- 主机表：与 TCPWappers 的书写方式类似。
- 参数项：控制共享目录的访问权限，用户映射等。
  - ro：设置共享目录为只读的权限
  - rw：设置共享目录为可读写的权限
  - root_squash：将root用户或其所属组映射成匿名用户或组（nfsnobody），这是默认值
  - no_root_squash：将root用户或其所属组映射成匿名用户或组，这样设置很不安全不建议使用
  - all_squash：将所有远程访问的普通用户或组都映像成匿名用户或组，适合公用目录
  - no_all_squash：不将所有远程访问的普通用户或组都映像成匿名用户或组，这是默认值

/etc/exports配置文件举例


```
/var/ftp/pub          *(ro)
/var/ftp/yum          192.168.0.0/24(ro) 192.168.1.0/24(ro)
/kickstart/centos     192.168.0.0/24(ro)
/var/ftp/incoming     192.168.0.0/24(rw,all_squash,anonuid=14,anongid=50)
/srv/www              www?.ls-al.me(ro)
/srv/public           192.168.1.0/24(rw)    *(ro)
/backup               192.168.1.0/24(rw,no_root_squash)
```


梁如军（linuxbooks@126.com）

exportfs命令

- 用于维护NFS共享的目录列表。
- 当修改了/etc/exports之后，无需重新启动nfs服务，可以使用exportfs命令使改动立刻生效。
- exportfs命令格式为：
  - exportfs [-aruv]
    - -a ： 全部挂载或卸载 /etc/exports 配置文件中的设置
    - -r ： 重新挂载 /etc/exports 中的设置，同步更新/var/lib/nfs/xtab的内容。
    - -u ： 卸载共享目录。
    - -v ： 在显示输出列表同时，显示设定参数。

梁如军（linuxbooks@126.com）

NFS与防火墙

- NFS V4
  - 仅开启对tcp:2049端口访问即可
- NFS V3
  - 开启对tcp:2049端口
  - 同时开启对rpcbind（111端口）的访问
  - 配置其他基于RPC的NFS V3相关服务使用固定端口
  - 开启对配置的固定端口的访问

配置NFS V3服务固定端口

- rquotad,mountd, statd 和 lockd 可以被强制使用一个静态端口
- /etc/sysconfig/nfs

```
RPCRQUOTADOPTS="-p 30001"
LOCKD_TCPPORT=30002
LOCKD_UDPPORT=30002
RPCMOUNTDOPTS="-p 30003"
STATDARG="-p 30004"
```


梁如军（linuxbooks@126.com）

NFS客户端

- 查看NFS服务器共享目录
- NFS文件系统的挂载与卸载
- 在启动时挂载NFS文件系统

梁如军（linuxbooks@126.com）

showmount命令

- 查看NFS服务器上所有的共享目录
  - 格式：**showmount -e [<Hostname> | <IP>]**
  - 例如：**# showmount -e 192.168.0.252**

- 查看服务器上哪些共享目录已经被客户端挂载
  - 格式：**showmount -d [<Hostname> | <IP>]**
  - 例如：**# showmount -d 192.168.0.252**

梁如军（linuxbooks@126.com）

NFS文件系统挂载与卸载

- 挂载
  - **mount -t nfs [-o 参数] 服务器地址:/共享目录 /本机挂载点**
  - 例如：将NFS服务器（192.168.0.252）的共享目录/backup挂载到本地的/backup的命令为：
  - **# mount -t nfs 192.168.0.252:/backup /backup**
- 卸载
  - **umount /本机挂载点**
  - 例如：要卸载本地已挂载的NFS文件系统
  - **# umount /backup**

梁如军（linuxbooks@126.com）

在启动时挂载NFS文件系统

- /etc/fstab

```
192.168.0.252:/backup      /backup       nfs   hard, intr   0   0
192.168.0.252:/var/ftp/pub  / var/ftp/pub  nfs   hard, intr   0   0
```


梁如军（linuxbooks@126.com）

本章思考题

- 简述FTP的数据传输模式及使用场合。
- FTP的使用者分为哪几类？
- vsftpd在RHEL/CentOS 7中的默认配置提供了哪些功能？
- 简述NFS与RPC的关系。
- NFS的常用工具有哪些？其用途和使用方法？

梁如军（linuxbooks@126.com）

本章实验

- 学会配置高安全级别的匿名FTP服务器。
- 学会配置允许匿名用户上传的FTP服务器。
- 学会配置vsftpd的最大传输速率限制和每客户的连接数限制。
- 学会配置vsftpd的基于本地用户的访问控制。
- 学会配置vsftpd的基于主机的访问控制。
- 学会对不同的主机或网络地址的访问实施不同的配置。
- 学会配置vsftpd基于虚拟用户的FTP服务
- 学会配置基于SSL的FTP服务
- 学会配置NFS的共享目录。
- 学会使用mount命令挂装NFS共享目录。
- 学会通过修改/etc/fstab文件在启动时挂装NFS文件系统。

梁如军（linuxbooks@126.com）

进一步学习

- 学习基于vsftpd的虚拟用户的FTP服务器配置。
- 了解、学习另一种Linux下常用的FTP服务器pure-ftpd的配置方法。
- 学习autofs守护进程的功能和用途，学会配置autofs自动挂装NFS文件系统。
- 学习使用跨平台的FTP客户工具Filezilla（http://filezilla-project.org/）。

梁如军（linuxbooks@126.com）