Pure-FTPd + LDAP + MySQL + PGSQL + Virtual-Users + Quota How To

Netkiller(陈景峰)

目录

1 准备工作

```
[root@linux root]# wget ftp://ftp.pureftpd.org/pub/pure-ftpd/releases/pure-ftpd-1.0.15.tar.gz
[root@linux root]# wget http://home.9812.net/linux/download/myphp/site-2.1.0.tar.gz
mysql : http://www.mysql.com
pgsql: http://www.postgresql.org
openldap: http://www.openldap.org
```

1.1 安装 MySQL 数据库

```
[root@linux mysql]$ cat install
rpm -Uvh MySQL-server-4.0.13-0.i386.rpm
rpm -Uvh MySQL-client-4.0.13-0.i386.rpm
rpm -Uvh MySQL-devel-4.0.13-0.i386.rpm
rpm -Uvh MySQL-shared-4.0.13-0.i386.rpm
rpm -Uvh MySQL-shared-compat-4.0.13-0.i386.rpm

[root@linux root]# service mysql start
```

1.2 安装 PostgreSQL 数据库

```
[root@linux pgsql]$ cat install
rpm -Uvh --nodeps postgresql-libs-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-devel-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-server-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-contrib-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-docs-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-jdbc-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-pl-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-python-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-tcl-?.?.?-1PGDG.i386.rpm
rpm -Uvh --nodeps postgresql-test-?.?.?-1PGDG.i386.rpm

[root@linux root]# rpm -qa|grep post
[root@linux root]# service postgresql start
```

1.3 安装 OpenLDAP

```
[root@linux ldap]$ cat install
```

```
rpm -ivh openldap-servers-2.0.25-1.i386.rpm
rpm -ivh openldap-clients-2.0.25-1.i386.rpm
rpm -ivh openldap-2.0.25-1.i386.rpm
rpm -ivh openldap12-1.2.13-5.i386.rpm
rpm -ivh openldap-devel-2.0.25-1.i386.rpm

[root@linux root]# service ldap start
```

2 安装 Pure-FTPd

```
[root@linux root]# tar zxvf pure-ftpd-1.0.15.tar.gz
[root@linux root]# cd pure-ftpd-1.0.15

[root@linux pure-ftpd-1.0.15]#./configure \
--prefix=/usr/local/pureftpd \
--with-ldap \
--with-mysql \
--with-pgsql \
--with-puredb \
--with-shadow \
--with-pam \
--with-paranoidmsg \
--with-welcomemsg \
--with-uploadscript \
--with-cookie \
--with-virtualchroot \
--with-virtualhosts \
--with-virtualroot \
--with-diraliases \
--with-quotas \
--with-sysquotas \
--with-ratios \
--with-ftpwho \
--with-throttling \
--with-language=simplified-chinese

[root@linux pure-ftpd-1.0.15]#make
[root@linux pure-ftpd-1.0.15]#make check
[root@linux pure-ftpd-1.0.15] #make install

[root@linux pure-ftpd-1.0.15]# cd configuration-file
[root@linux configuration-file]# chmod u+x pure-config.pl
[root@linux configuration-file]# cp pure-config.pl /usr/local/pureftpd/bin
```

```
[root@linux configuration-file]# cp pure-ftpd.conf /usr/local/pureftpd/etc
[root@linux configuration-file]# cd ..
[root@linux pure-ftpd-1.0.15]# cp pureftpd* /usr/local/pureftpd/etc/
```

2.1 安装选项

--prefix=/usr/local/pureftpd \	软件安装到/usr/local/pureftpd 目录下
--with-ldap \	启用 LDAP 认证
--with-mysql \	启用 MySQL 认证
--with-pgsql \	启用 PgSQL 认证（Postgresql 这里我用的是最新版 7.3.3）
--with-puredb \	启用 puredb 认证 Pureftpd 自带的 Virtual-Users
--with-shadow \	启用 UNIX Shadow 认证就是系统用户
--with-pam \	启用 PAM 模块认证,PAM 是一种为通用设计的认证模块。 常见 PAM 模块有 pam-mysql、pam-pgsql、pam-ldap……
--with-paranoidmsg \	
--with-welcomemsg \	登录 FTP 显示欢迎信息
--with-uploadscript \	上载脚本
--with-cookie \	作用 cookie
--with-virtualchroot \	chroot 模式
--with-virtualhosts \	
--with-virtualroot \	
--with-diraliases \	
--with-quotas \	启用 PureFtpd 自身 Quota 功能
--with-sysquotas \	允许使用操作系统的 Quota(磁盘限额)
--with-ratios \	上传、下载比率如：1:5
--with-ftpwho \	使用 pure-ftpwho 命令查看线上用户
--with-throttling \	频宽可设限.
--with-largefile \	载超过 2G 的文件.
--with-language=simplified-chinese	Socket 会话显示出来的信息的语言.缺省为英语. simplified-chinese 简体中文 traditional-chinese BIG5 繁体中文

```
          *** CuteFTP Pro 3.2 - build Jul  1 2003 ***

状态:>    正在获取列表“”...
状态:>    正在解析主机名 mail.9812.net...
状态:>    已解析主机名 mail.9812.net: ip = 202.103.190.130。
状态:>    正在连接到 ftp 服务器 mail.9812.net:21 (ip = 202.103.190.130)...
状态:>    Socket 已连接。正在等候欢迎消息...
          220---------- 欢迎来到 Pure-FTPd ----------
          220-您是第 1 个使用者，最多可达 50 个连线
          220-现在本地时间是 23:36  K欧    鞑嚎? 21。
```

```
        220 在 15 分钟内没有活动，您被会断线。
状态:>  已连接。正在验证...
命令:>  USER netkiller
        331 使用者 netkiller OK. 需要密码.
命令:>  PASS *****
        230-使用者 netkiller 有群组存取于:   chen
        230-这个伺服器支援 FXP 传输
        230 OK. 目前限制的目录是 /
状态:>  登录成功。
命令:>  PWD
        257 "/" 是您目前的位置
状态:>  Home directory: /
命令:>  FEAT
        211-Extensions supported:
         EPRT
         IDLE
         MDTM
         SIZE
         REST STREAM
         MLST type*;size*;sizd*;modify*;UNIX.mode*;UNIX.uid*;UNIX.gid*;unique*;
         MLSD
         TVFS
         ESTP
         PASV
         EPSV
         SPSV
         ESTA
        211 End.
状态:>  该站点支持 features。
状态:>  该站点支持 SIZE。
状态:>  该站点可以续传中断的下载。
命令:>  REST 0
        350 重新开始于 0
命令:>  PASV
        227 Entering Passive Mode (202,103,190,130,179,187)
命令:>  LIST
状态:>  正在连接 ftp 数据 socket 202.103.190.130:46011...
        150 接受资料连线
        226-Options: -a -l
        226 总共 48 符合
状态:>  传送完成。
```

3 配置 pure-ftpd.conf

在这里我全使用默认值，只修改下面几项。（注：Pureftpd 可以同时支持 ldap,mysql,pgsql,puredb 认证）

```
# LDAP configuration file (see README.LDAP)
LDAPConfigFile                    /usr/local/pureftpd/etc/pureftpd-ldap.conf

# MySQL configuration file (see README.MySQL)
MySQLConfigFile                   /usr/local/pureftpd/etc/pureftpd-mysql.conf


# Postgres configuration file (see README.PGSQL)
PGSQLConfigFile                   /usr/local/pureftpd/etc/pureftpd-pgsql.conf

# PureDB user database (see README.Virtual-Users)
PureDB                            /usr/local/pureftpd/etc/pureftpd.pdb
```

3.1 配置文件详解

ChrootEveryone yes

chroot 每一个用户,等同于 Proftpd 中的 DefaultRoot~，可以限制用户在某个地方活动，增强服务器的安全性。使用过 wu-ftpd 的使用都应该知道 cd /会发生什么！

TrustedGID 50

#以上两者要一起用

BrokenClientsCompatibility no

MaxClientsNumber   50

#最大链接数

Daemonize   yes

#Fork in background 以守护进程方式在后台运行

MaxClientsPerIP 5

#每个 ip 最多链接数，最好设小点。

VerboseLog no

#是否要把所有 client 端的指令都 log 下来

DisplayDotFiles no

#显示开头的文件

AnonymousOnly no

#是否只让匿名登录

NoAnonymous yes

#不开放匿名登入

SyslogFacility ftp

#应该是对日志做一下过滤 (auth, authpriv, daemon, ftp, security, user, local*)可以让日志只记录想要的信息

DontResolve yes

#不反向解释客户端的 ip

```
MaxIdleTime 5
#最大闲置時間
LDAPConfigFile                    /usr/local/pureftpd/etc/pureftpd-ldap.conf
#使用 LDAP 认证，
MySQLConfigFile                   /usr/local/pureftpd/etc/pureftpd-mysql.conf
#使用 MySQL 认证
PGSQLConfigFile                   /usr/local/pureftpd/etc/pureftpd-pgsql.conf
#使用 PGSQL 认证
PureDB /ftp/etc/pureftpd.pdb
#使用者资料的 DB 存放地点 [由于我是用 PureFTPD 的内建 DB.固有此选项]
ExtAuth                           /var/run/ftpd.sock
#pure-authd socket 路径 (详细请看 README.Authentication-Modules)
PAMAuthentication                yes
#开启 PAM 认证
UnixAuthentication yes
#如果你想要有简单的 Unix(/etc/passwd)的认证的話
LimitRecursion               2000   8
#ls 最多列出 3000 个文件.最深 8 层
AnonymousCanCreateDirs          no
#匿名用户可以创建目录
MaxLoad 4
#当 system load 超过 4 時.使用者将不能再下载
PassivePortRange              30000 50000
#被动连接应答范围
ForcePassiveIP                    192.168.0.1
#不会译：(
AnonymousRatio                     1 10
#Anonymous 连接上传/下载比率
UserRatio                         1 10
#用户上传/下载比率（注：如果使用 ldap,mysql,pgsql,pam 不要启用该功能，否则你在 ldap
等中设置的 Ratio 无校）
AntiWarez no
#上传的文件不能被下载(owner is ftp).等到 local admin 确认
Bind                              127.0.0.1,8021
#要绑定和 ip/port,在你的系统中有两个 FTP Server 这样你其中一个 FTP 就要使用其它端口。
#格式-> 127.0.0.1,21 如果只写 port 表 All ip,port
AnonymousBandwidth                 8
#Anonymous 带宽，单位 KB/s
UserBandwidth                     8
#用户带宽，单位 KB/s
Umask 133:022
#上传文件的 Umask.(<umask for files>:<umask for dirs>)
MinUID 1000
# UID 至少多少才能登录
```

```
AllowUserFXP yes
#支不支持 FXP
AllowAnonymousFXP                 no
#Anonymous 支不支持 FXP
ProhibitDotFilesWrite no
ProhibitDotFilesRead no
#(”.”)开头的文件能不能被读/写,UNIX Like 下以点开头的文件是隐藏文件 ls –a 才能列出
#Pureftpd Quota 模式下做产生” .ftpquota”文件。
AutoRename no
#上传文件若有相同文件名自动改名(file.1,file.2...)
AnonymousCantUpload               no
#匿名用户上传文件
TrustedIP 10.1.1.1
#锁 IP.
LogPID
#Log 文件添加 PID
AltLog stats:/ftp/etc/log/pureftpd.log
#log 存放地点，注日志有几种常用的格式
#clf  类似 apache 格式，stats UNIX log 格式，w3c  标准 W3C 格式，可能是 HTML 格式
NoChmod yes
#不给 Chmod 指令的权限
KeepAllFiles yes
#使用者可续传.但不可删除文件
CreateHomeDir no
#如果 user 的 home 不存在自动建立
Quota                                    1000:10
#Quota <文件数>:<容量 Megabytes >，FTP 限制 10M 空间，可以上传 1000 个文件（注：如
果使用 ldap,mysql,pgsql,pam 不要启用该功能，否则你在 ldap 等中设置的 Quota 无校）
PIDFile /ftp/etc/log/pure-ftpd.pid
#记录 pure-ftpd 的 PID 文件
CallUploadScript yes
#呼叫 UploadScript
MaxDiskUsage 99
#当硬盘使用率到多少時将停止上传
NoRename yes
#用户不能重命名文件名
CustomerProof yes
PerUserLimits 3:20
#<每个账号最多可登入几次:Anonymous 最多可同時登入几次>
```

4 运行 pureftpd

```
[root@linux bin]# pure-config.pl ../etc/pure-ftpd.conf
```

5 MySQL 模块

5.1 创建 MySQL 数据库

```

CREATE DATABASE pureftpd;
grant all on pureftpd.* to pureftpd@localhost identified by 'qKiscCbwbXAkWp.'

DROP TABLE IF EXISTS `users`;
CREATE TABLE `users` (
    `id` int(32) unsigned NOT NULL auto_increment,
    `User` varchar(16) NOT NULL default '',
    `Password` varchar(64) NOT NULL default '',
    `Uid` varchar(11) NOT NULL default '-1',
    `Gid` varchar(11) NOT NULL default '-1',
    `Dir` varchar(128) NOT NULL default '',
    `QuotaSize` smallint(5) NOT NULL default '0',
    `QuotaFiles` int(11) NOT NULL default '0',
    `ULBandwidth` smallint(5) NOT NULL default '0',
    `DLBandwidth` smallint(5) NOT NULL default '0',
    `ULRatio` smallint(6) NOT NULL default '0',
    `DLRatio` smallint(6) NOT NULL default '0',
    `comment` tinytext NOT NULL,
    `ipaccess` varchar(15) NOT NULL default '*',
    `status` enum('0','1') NOT NULL default '0',
    `create_date` datetime NOT NULL default '0000-00-00 00:00:00',
    `modify_date` datetime NOT NULL default '0000-00-00 00:00:00',
    PRIMARY KEY    (`id`,`User`),
    UNIQUE KEY `User` (`User`)
) TYPE=MyISAM AUTO_INCREMENT=5 ;

INSERT INTO `users` VALUES (5, 'test', encrypt('test'), '505', '505', '/tmp', 0, 0, 0, 0, 0, 0, '', '*', '1',
'2003-06-26 18:04:33', '2003-06-26 18:04:33');
```

5.2 配置 pureftpd-mysql.conf

```
# Mandatory : user to bind the server as.

MYSQLUser           pureftpd

```

```
# Mandatory : user password. You must have a password.

MYSQLPassword      qKiscCbwbXAkWp.

# Mandatory : database to open.

MYSQLDatabase      pureftpd

# Mandatory : how passwords are stored
# Valid values are : "cleartext", "crypt", "md5" and "password"
# ("password" = MySQL password() function)
# You can also use "any" to try "crypt", "md5" *and* "password"

#MYSQLCrypt            cleartext
MYSQLCrypt            crypt

# Query to execute in order to fetch the password

MYSQLGetPW            SELECT Password FROM users WHERE User="\L"

# Query to execute in order to fetch the system user name or uid

MYSQLGetUID          SELECT Uid FROM users WHERE User="\L"

# Query to execute in order to fetch the system user group or gid

MYSQLGetGID          SELECT Gid FROM users WHERE User="\L"

# Query to execute in order to fetch the home directory

MYSQLGetDir          SELECT Dir FROM users WHERE User="\L"

# Optional : query to get the maximal number of files
# Pure-FTPd must have been compiled with virtual quotas support.

MySQLGetQTAFS    SELECT QuotaFiles FROM users WHERE User="\L"

# Optional : query to get the maximal disk usage (virtual quotas)
# The number should be in Megabytes.
# Pure-FTPd must have been compiled with virtual quotas support.

MySQLGetQTASZ    SELECT QuotaSize FROM users WHERE User="\L"

# Optional : ratios. The server has to be compiled with ratio support.
```

```
MySQLGetRatioUL SELECT ULRatio FROM users WHERE User="\L"
MySQLGetRatioDL SELECT DLRatio FROM users WHERE User="\L"

# Optional : bandwidth throttling.
# The server has to be compiled with throttling support.
# Values are in KB/s .

MySQLGetBandwidthUL SELECT ULBandwidth FROM users WHERE User="\L"
MySQLGetBandwidthDL SELECT DLBandwidth FROM users WHERE User="\L"
```

5.3 配置文件详解

```
MYSQLServer        127.0.0.1
#MYSQL 服务器的 IP
MYSQLPort          3306
#MYSQL  端口号
MYSQLSocket        /var/lib/mysql/mysql.sock
#使用 UNIX.sock 本地连接
```

注：MYSQLServer 与 MYSQLSocket 选择一种即可

```
MYSQLUser          pureftpd
#MYSQLUser 数据用户名
MYSQLPassword      123456
#MYSQL 数据库用户的密码
MYSQLDatabase      pureftpd
#FTP 数据数据库
MYSQLCrypt         crypt
#密码加密方式"cleartext", "crypt", "md5" and "password"
# cleartext  明文，crypt，md5,password 是 Backend password(‘your-passwd’)函数（MYSQL 数据库所使用的 password（）函数）
MYSQLGetPW         SELECT Password FROM users WHERE User="\L"
#  密码字段，我使用 users 表中的 Password 做为密码字段
MYSQLGetUID        SELECT Uid FROM users WHERE User="\L"
#UID 用户 ID 字段
MYSQLDefaultUID 1000
#默认的 UID  （注：如何开启该选项，MYSQLGetUID 将失去作用）
MYSQLGetGID        SELECT Gid FROM users WHERE User="\L"
#GID 组 ID 字段
MYSQLDefaultGID 1000
#默认的 GID  （注：如何开启该选项，MYSQLGetGID 将失去作用）
MYSQLGetDir        SELECT Dir FROM users WHERE User="\L"
#FTP 用户目录如/home/web/www-9812-net
```

```
MySQLGetQTAFS    SELECT QuotaFiles FROM users WHERE User="\L"
#磁盘限额，文件数限制。如 1000，允许用户上传 1 千个文件
MySQLGetQTASZ    SELECT QuotaSize FROM users WHERE User="\L"
#磁盘限额，FTP 用户空间限制（单位为 M），如：100M
MySQLGetRatioUL SELECT ULRatio FROM users WHERE User="\L"
MySQLGetRatioDL SELECT DLRatio FROM users WHERE User="\L"
#上传/下载比率。MySQLGetRatioUL 为上传比，MySQLGetRatioDL 下载比。如：1：5
MySQLGetBandwidthUL SELECT ULBandwidth FROM users WHERE User="\L"
MySQLGetBandwidthDL SELECT DLBandwidth FROM users WHERE User="\L"
#下传/下载带宽（单位 KB/s）。MySQLGetBandwidthUL 上传带宽，MySQLGetBandwidthDL
下载带宽。如上传 500KB/s,下载 50KB/s
MySQLForceTildeExpansion 1
MySQLTransactions On
#不会翻译
```

5.4 测试 pureftpd

```
启动 pureftpd
[root@linux root]# /usr/local/pureftpd/bin/pure-config.pl
/usr/local/pureftpd/etc/pure-ftpd.conf

测试 pureftpd
[root@linux root]ncftp ftp://test:test@localhost:21
```

6 PGSQL 模块

6.1 配置 PostgreSQL 数据库

6.1.1 postgresql.conf

```
[root@linux root]# vi /var/lib/pgsql/data/postgresql.conf
tcpip_socket = true
```

6.1.2 pg_hba.conf

```
[root@linux root]# vi /var/lib/pgsql/data/pg_hba.conf
host     all        all        127.0.0.1        255.255.255.255     md5
local    all        all                                             trust
```

加入上面几行

6.1.3 Restart PostgreSQL

```
[root@linux root]# service postgresql restart
Starting postgresql service:                                    [   OK   ]
```

6.2 创建 PostgreSQL 数据库

```
[root@linux root]# su postgres
bash-2.05$ createdb
bash-2.05$ psql -l
              List of databases
     Name      |   Owner    | Encoding
-----------+----------+-----------
  postgres    | postgres | SQL_ASCII
  template0 | postgres | SQL_ASCII
  template1 | postgres | SQL_ASCII
(5 rows)

bash-2.05$ psql
postgres=# CREATE USER pureftpd WITH PASSWORD ' pureftpd ';
CREATE USER
postgres=# CREATE DATABASE pureftpd WITH OWNER = pureftpd TEMPLATE = template0
ENCODING = 'EUC_CN';
CREATE DATABASE
postgres=# \q
bash-2.05$
bash-2.05$ psql -l
              List of databases
     Name      |   Owner    | Encoding
-----------+----------+-----------
  postgres    | postgres | SQL_ASCII
  pureftpd    | pureftpd | EUC_CN
  template0 | postgres | SQL_ASCII
  template1 | postgres | SQL_ASCII
(5 rows)

bash-2.05$ createlang plpgsql pureftpd

bash-2.05$ psql -u pureftpd
psql: Warning: The -u option is deprecated. Use -U.
```

```
User name: pureftpd
Password:
Welcome to psql 7.3.2, the PostgreSQL interactive terminal.

Type:   \copyright for distribution terms
        \h for help with SQL commands
        \? for help on internal slash commands
        \g or terminate with semicolon to execute query
        \q to quit

pureftpd=>


DROP TABLE users CASCADE;
DROP SEQUENCE users_id_seq CASCADE;
CREATE TABLE "users" (
    id integer DEFAULT nextval('users_id_seq'::text) NOT NULL,
    "User" character varying(16) NOT NULL default '',
    status smallint default 0,
    "Password" character varying(64) NOT NULL default '',
    "Uid" character varying(11) DEFAULT -1 NOT NULL,
    "Gid" character varying(11) DEFAULT -1 NOT NULL,
    "Dir" character varying(128) NOT NULL,
    "comment" text,
    ipaccess character varying(15) DEFAULT '*' NOT NULL,
    "ULBandwidth" smallint default 0,
    "DLBandwidth" smallint default 0,
    "QuotaSize" integer DEFAULT 0,
    "QuotaFiles" integer DEFAULT 0,
    ULRatio smallint default 0,
    DLRatio smallint default 0,
    create_date timestamp with time zone DEFAULT now() NOT NULL,
    modify_date timestamp without time zone DEFAULT now() NOT NULL
);

CREATE SEQUENCE users_id_seq;
CREATE INDEX users_index ON users (id,"User");
ALTER TABLE ONLY users ADD CONSTRAINT users_pkey PRIMARY KEY (id);
ALTER TABLE ONLY users ADD CONSTRAINT users_id_key UNIQUE (id, "User");

pureftpd=> \d
                List of relations
 Schema |       Name       |   Type   |  Owner
--------+--------------+----------+----------
```

```
  public | users          | table    | pureftpd
  public | users_id_seq | sequence | pureftpd
(2 rows)

pureftpd=>
```

6.3 配置 pureftpd-pgsql.conf

```
# If PostgreSQL listens to a TCP socket
PGSQLServer       localhost
# *or* if PostgreSQL can only be reached through a local Unix socket
# PGSQLServer       /tmp
# PGSQLPort         .s.PGSQL.5432

# Mandatory : user to bind the server as.
PGSQLUser         pureftpd

# Mandatory : user password. You *must* have a password.
PGSQLPassword     pureftpd

# Mandatory : database to open.
PGSQLDatabase     pureftpd

# Mandatory : how passwords are stored
# Valid values are : "cleartext", "crypt", "md5" or "any"
#PGSQLCrypt         cleartext
PGSQLCrypt         crypt

PGSQLGetPW         SELECT Password FROM users WHERE User='\L'

# Query to execute in order to fetch the system user name or uid
PGSQLGetUID        SELECT Uid FROM users WHERE User='\L'

# Query to execute in order to fetch the system user group or gid
PGSQLGetGID        SELECT Gid FROM users WHERE User='\L'

# Query to execute in order to fetch the home directory
PGSQLGetDir        SELECT Dir FROM users WHERE User='\L'



# Optional : query to get the maximal number of files
# Pure-FTPd must have been compiled with virtual quotas support.
PGSQLGetQTAFS    SELECT QuotaFiles FROM users WHERE User='\L'
```

```
# Optional : query to get the maximal disk usage (virtual quotas)
# The number should be in Megabytes.
# Pure-FTPd must have been compiled with virtual quotas support.
PGSQLGetQTASZ    SELECT QuotaSize FROM users WHERE User='\L'

# Optional : ratios. The server has to be compiled with ratio support.
PGSQLGetRatioUL SELECT ULRatio FROM users WHERE User='\L'
PGSQLGetRatioDL SELECT DLRatio FROM users WHERE User='\L'

# Optional : bandwidth throttling.
# The server has to be compiled with throttling support.
# Values are in KB/s .

PGSQLGetBandwidthUL SELECT ULBandwidth FROM users WHERE User='\L'
PGSQLGetBandwidthDL SELECT DLBandwidth FROM users WHERE User='\L'
```

6.4 配置文件详解

```
PGSQLServer        127.0.0.1
#PGSQL 服务器的 IP
PGSQLPort          5432
#MYSQL  端口号

PGSQLServer        /tmp
PGSQLPort          .s.PGSQL.5432
#使用 UNIX .sock 本地连接,/tmp/.s.PGSQL.5432
```

注：PGSQLServer   127.0.0.1 与 PGSQLServer   /tmp 选择一种即可

```
PGSQLUser          system
#数据用户名
PGSQLPassword      system
#数据库用户的密码
PGSQLDatabase      system
# FTP 数据数据库
PGSQLCrypt         cleartext
#密码加密方式"cleartext", "crypt", "md5" and "password"
# cleartext  明文，crypt，md5,password 是 Backend password(‘your-passwd’)函数（MYSQL 数据库所使用的 password（）函数）

PGSQLGetPW         SELECT Password FROM users WHERE User='\L'
```

```
#  密码字段，我使用 users 表中的 Password 做为密码字段
PGSQLGetUID          SELECT Uid FROM users WHERE User='\L'
#UID 用户 ID 字段
PGSQLDefaultUID 1000
#默认的 UID （注：如何开启该选项，PGSQLGetUID 将失去作用）
PGSQLGetGID          SELECT Gid FROM users WHERE User='\L'
#GID 组 ID 字段
PGSQLDefaultGID 1000
#默认的 GID （注：如何开启该选项，MYSQLGetGID 将失去作用）
PGSQLGetDir          SELECT Dir FROM users WHERE User='\L'
#FTP 用户目录如/home/web/www-9812-net
# PGSQLGetQTAFS    SELECT QuotaFiles FROM users WHERE User='\L'
#磁盘限额，文件数限制。如 1000，允许用户上传 1 千个文件
# PGSQLGetQTASZ    SELECT QuotaSize FROM users WHERE User='\L'
#磁盘限额，FTP 用户空间限制（单位为 M），如：100M
PGSQLGetRatioUL SELECT ULRatio FROM users WHERE User='\L'
PGSQLGetRatioDL SELECT DLRatio FROM users WHERE User='\L'
#上传/下载比率。MySQLGetRatioUL 为上传比，MySQLGetRatioDL 下载比。如：1：5
PGSQLGetBandwidthUL SELECT ULBandwidth FROM users WHERE User='\L'
PGSQLGetBandwidthDL SELECT DLBandwidth FROM users WHERE User='\L'
#下传/下载带宽（单位 KB/s）。MySQLGetBandwidthUL 上传带宽，MySQLGetBandwidthDL 下载带宽。如上传 500KB/s,下载 50KB/s
```

6.5 测试 pureftpd

```
启动 pureftpd
[root@linux root]# /usr/local/pureftpd/bin/pure-config.pl
/usr/local/pureftpd/etc/pure-ftpd.conf

测试 pureftpd
[root@linux root]ncftp ftp://test:test@localhost:21
```

7 LDAP 模块

OpenLDAP 使用 Berkeley DB (一个层次型数据库，注意：与 RDBMS 不同) 存储数据

7.1 配置 OpenLDAP

```
[root@linux root]vi /etc/openldap/slapd.conf
include              /etc/openldap/schema/pureftpd.schema
```

```
suffix                "dc=gdfz,dc=com"
rootdn                 "cn=Manager,dc=gdfz,dc=com"
rootpw                  {crypt}sa0hRW/W3DLvQ
```

```
[root@linux root]service ldap restart
```

7.2 rootdn 的结构

rootdn:dc=gdfz,dc=com

```
|-----cn=one, dc=gdfz,dc=com
|       |--- objectClass: posixAccount
|       |---cn: joe
|       |---uid: joe
|       |---uidNumber: 500
|       |---gidNumber: 500
|       |---homeDirectory: /home/joe
|       |---userPassword: {crypt}saO3qRXM8wjUE
|---- cn=xxx-1, dc=gdfz,dc=com
|       |--- …………………………
|       |--- …………………………
|---- cn=xxx-n, dc=gdfz,dc=com
|---- ou=two, dc=gdfz,dc=com
|       |---- cn=one,ou=two, dc=gdfz,dc=com
|       |       |--- objectClass: posixAccount
|       |       |---cn: joe
|       |       |---uid: joe
|       |       |---uidNumber: 500
|       |       |---gidNumber: 500
|       |       |---homeDirectory: /home/joe
|       |       |---userPassword: {crypt}saO3qRXM8wjUE
|       |---- cn=two,ou=two, dc=gdfz,dc=com
|       |       |--- …………………………
|       |       |--- …………………………
|       |---- cn=there,ou=two, dc=gdfz,dc=com
|---- ou=other, dc=gdfz,dc=com
        |---- cn=one,ou=other, dc=gdfz,dc=com
        |---- cn=two,ou=other, dc=gdfz,dc=com
```

7.3 创建 dn

```
[root@linux root]# cat base-dn.ldif
dn: dc=gdfz,dc=com
objectClass: person
```

```
cn: gdfz
sn: gdfz
ldapadd -x -D "cn=manager,dc=gdfz,dc=com" -w [你的 rootpw 密码] -f base-dn.ldif
[root@linux etc]# cat pureftpd.ldif
dn: cn=joe,dc=gdfz,dc=com
objectClass: posixAccount
cn: joe
uid: joe
uidNumber: 500
gidNumber: 500
homeDirectory: /home/joe
userPassword: {crypt}saO3qRXM8wjUE
[root@linux root]#ldapadd -x -D "cn=manager,dc=gdfz,dc=com" -w [你的 rootpw 密码] -f pureftpd.ldif

[root@linux root]# cat pureftpd.ldif
dn: uid=chen,dc=gdfz,dc=com
objectClass: posixAccount
cn: chen
uid:chen
uidnumber:501
gidNumber:501
homeDirectory: /home/chen
userPassword: {crypt}$1$chen$y13/Ao8O3O/9jhSSCPFZg0
objectClass: PureFTPdUser
FTPStatus: enabled
FTPQuotaFiles: 50
FTPQuotaMBytes: 10
FTPDownloadBandwidth: 50
FTPUploadBandwidth: 50
FTPDownloadRatio: 5
FTPUploadRatio: 1
[root@linux root]# ldapadd -x -D "cn=manager,dc=gdfz,dc=com" -w [你的 rootpw 密码] -f pureftpd.ldif
```

7.4 pureftpd-ldap.conf

```
LDAPServer localhost
# Optional : server port. Default : 389

LDAPPort     389

# Mandatory : the base DN to search accounts from. No default.
```

```
LDAPBaseDN dc=gdfz,dc=com


# Optional : who we should bind the server as.
#                    Default : binds anonymously
LDAPBindDN cn=Manager,dc=gdfz,dc=com


# Password if we don't bind anonymously
# This configuration file should be only readable by root
LDAPBindPW chen
```

7.5 配置文件详解

LDAPServer localhost
#LDAP 服务器地址
LDAPPort   389
#LDAP 端口号
LDAPBaseDN dc=gdfz,dc=com
#基本 DN
LDAPBindDN cn=Manager,dc=gdfz,dc=com
#绑定 DN，LDAP 管理员
LDAPBindPW chen
#管理员密码
LDAPDefaultUID 500
LDAPDefaultGID 100
#默认的 UID，GID （注：如果设置该 uidnumber:501，gidNumber:501 设置将无效）
LDAPFilter (&(objectClass=posixAccount)(uid=\L))
#过滤 LDAP 条目，当你使用 ldapsearch 检索条目时不做过滤，会列出所有条目，如果你的数据量很大，输入所有条目要很久，所以要对你的 DN 做过滤，将 FTP 服务器用的条目过滤出来。LDAPFilter (&(objectClass=posixAccount)(uid=\L)) 类似 RDBMS 中游标（游标请看 PostgreSQL 手册 45.7. 游标 http://www.pgsqldb.org/pgsqldoc-cvs/plpgsql-cursors.html）。
LDAPHomeDir homeDirectory
#FTP 的用户目录
LDAPVersion 3
#LDAP 版本，目前主流的 LDAP 服务器都是 v3 版,如：ActiveDirectory,OpenLDAP,Novell NDS,SUN ONE LDAP……

7.6 测试 pureftpd

```
启动 pureftpd
```

```
[root@linux root]# /usr/local/pureftpd/bin/pure-config.pl
/usr/local/pureftpd/etc/pure-ftpd.conf

测试 pureftpd
[root@linux root]ncftp ftp://chen:passwd@localhost:21
```

8 Virtual-Users

```
pure-pw  使用方法

[root@linux bin]# ./pure-pw

Usage :

pure-pw useradd <login> [-f <passwd file>] -u <uid> [-g <gid>]
                        -D/-d <home directory> [-c <gecos>]
                        [-t <download bandwidth>] [-T <upload bandwidth>]
                        [-n <max number of files>] [-N <max Mbytes>]
                        [-q <upload ratio>] [-Q <download ratio>]
                        [-r <allow client ip>/<mask>] [-R <deny client ip>/<mask>]
                        [-i <allow local ip>/<mask>] [-I <deny local ip>/<mask>]
                        [-y <max number of concurrent sessions>]
                        [-z <hhmm>-<hhmm>] [-m]

pure-pw usermod <login> -f <passwd file> -u <uid> [-g <gid>]
                        -D/-d <home directory> -[c <gecos>]
                        [-t <download bandwidth>] [-T <upload bandwidth>]
                        [-n <max number of files>] [-N <max Mbytes>]
                        [-q <upload ratio>] [-Q <download ratio>]
                        [-r <allow client ip>/<mask>] [-R <deny client ip>/<mask>]
                        [-i <allow local ip>/<mask>] [-I <deny local ip>/<mask>]
                        [-y <max number of concurrent sessions>]
                        [-z <hhmm>-<hhmm>] [-m]

pure-pw userdel <login> [-f <passwd file>] [-m]

pure-pw passwd   <login> [-f <passwd file>] [-m]

pure-pw show     <login> [-f <passwd file>]

pure-pw mkdb     [<puredb database file> [-f <passwd file>]]

pure-pw list     [-f <passwd file>]
```

```
-d <home directory> : chroot user (recommended)
-D <home directory> : don't chroot user
-<option> '' : set this option to unlimited
-m : also update the /usr/local/pureftpd/etc/pureftpd.pdb database
For a 1:10 ratio, use -q 1 -Q 10
To allow access only between 9 am and 6 pm, use -z 0900-1800


*WARNING* : that pure-ftpd server hasn't been compiled with puredb support

添加 9812 用户，用户目录/home/www/9812.net/,使用 web 用户的 uid 与 gid
[root@linux bin]# ./pure-pw useradd 9812 -u web -d /home/www/9812.net/
Password:
Enter it again:
[root@linux bin]#

[root@linux etc]# cat pureftpd.passwd
qqqq:$1$suA.WBZ0$Uu/05AtMi/4cNdhg9gKjP/:505:505::/home/web/./::::::::::::
9812:$1$4.iPvGE0$lY5CEVYLde.Mb9QWNu.so0:505:505::/home/www/9812.net/./::::::::::::

生成 pureftpd.pdb
[root@linux etc]# ../bin/pure-pw mkdb

[root@linux etc]# ls
pure-config.pl    pure-ftpd.conf    pureftpd-ldap.conf    pureftpd-mysql.conf    pureftpd.passwd
pureftpd.pdb    pureftpd-pgsql.conf

启动 pureftpd
[root@linux root]# /usr/local/pureftpd/bin/pure-config.pl /usr/local/pureftpd/etc/pure-ftpd.conf

测试 pureftpd
[root@linux root]ncftp ftp://9812:passwd@localhost:21
```

9 配置文件实例

9.1 pure-ftpd.conf

```
################################################################
#                                                              #
#            Configuration file for pure-ftpd wrappers        #
#                                                              #
```

```
############################################################

# If you want to run Pure-FTPd with this configuration
# instead of command-line options, please run the
# following command :
#
# /usr/local/pureftpd/sbin/pure-config.pl /usr/local/pureftpd/etc/pure-ftpd.conf
#
# RPM binary files use another configuration file by default :
# /etc/sysconfig/pure-ftpd
#
# Please don't forget to have a look at documentation at
# http://www.pureftpd.org/documentation.html for a complete list of
# options.

# Cage in every user in his home directory

ChrootEveryone              yes



# If the previous option is set to "no", members of the following group
# won't be caged. Others will be. If you don't want chroot()ing anyone,
# just comment out ChrootEveryone and TrustedGID.

# TrustedGID                    100



# Turn on compatibility hacks for broken clients

BrokenClientsCompatibility  no



# Maximum number of simultaneous users

MaxClientsNumber            50



# Fork in background

Daemonize                   yes
```

```
# Maximum number of sim clients with the same IP address

MaxClientsPerIP                 8



# If you want to log all client commands, set this to "yes".
# This directive can be duplicated to also log server responses.

VerboseLog                      no



# List dot-files even when the client doesn't send "-a".

DisplayDotFiles                 yes



# Don't allow authenticated users - have a public anonymous FTP only.

AnonymousOnly                   no



# Disallow anonymous connections. Only allow authenticated users.

NoAnonymous                     no



# Syslog facility (auth, authpriv, daemon, ftp, security, user, local*)
# The default facility is "ftp". "none" disables logging.

SyslogFacility                  ftp



# Display fortune cookies

# FortunesFile                  /usr/share/fortune/zippy
# Don't resolve host names in log files. Logs are less verbose, but
# it uses less bandwidth. Set this to "yes" on very busy servers or
# if you don't have a working DNS.

DontResolve                 yes



# Maximum idle time in minutes (default = 15 minutes)

MaxIdleTime                 15



# LDAP configuration file (see README.LDAP)

# LDAPConfigFile                /etc/pureftpd-ldap.conf
LDAPConfigFile                /usr/local/pureftpd/etc/pureftpd-ldap.conf



# MySQL configuration file (see README.MySQL)

# MySQLConfigFile               /etc/pureftpd-mysql.conf
MySQLConfigFile               /usr/local/pureftpd/etc/pureftpd-mysql.conf


# Postgres configuration file (see README.PGSQL)

# PGSQLConfigFile               /etc/pureftpd-pgsql.conf
PGSQLConfigFile               /usr/local/pureftpd/etc/pureftpd-pgsql.conf


# PureDB user database (see README.Virtual-Users)

# PureDB                        /etc/pureftpd.pdb
PureDB                        /usr/local/pureftpd/etc/pureftpd.pdb


# Path to pure-authd socket (see README.Authentication-Modules)
# ExtAuth                                    /var/run/ftpd.sock



# If you want to enable PAM authentication, uncomment the following line

# PAMAuthentication                   yes



# If you want simple Unix (/etc/passwd) authentication, uncomment this

# UnixAuthentication                  yes



# Please note that LDAPConfigFile, MySQLConfigFile, PAMAuthentication and
# UnixAuthentication can be used only once, but they can be combined
# together. For instance, if you use MySQLConfigFile, then UnixAuthentication,
# the SQL server will be asked. If the SQL authentication fails because the
# user wasn't found, another try # will be done with /etc/passwd and
# /etc/shadow. If the SQL authentication fails because the password was wrong,
# the authentication chain stops here. Authentication methods are chained in
# the order they are given.



# 'ls' recursion limits. The first argument is the maximum number of
# files to be displayed. The second one is the max subdirectories depth

LimitRecursion                   2000 8



# Are anonymous users allowed to create new directories ?

AnonymousCanCreateDirs           no



# If the system is more loaded than the following value,
# anonymous users aren't allowed to download.

MaxLoad                          4
```

```
# Port range for passive connections replies. - for firewalling.

# PassivePortRange          30000 50000



# Force an IP address in PASV/EPSV/SPSV replies. - for NAT.
# Symbolic host names are also accepted for gateways with dynamic IP
# addresses.

# ForcePassiveIP                192.168.0.1



# Upload/download ratio for anonymous users.

# AnonymousRatio                1 10



# Upload/download ratio for all users.
# This directive superscedes the previous one.

# UserRatio                     1 10



# Disallow downloading of files owned by "ftp", ie.
# files that were uploaded but not validated by a local admin.

AntiWarez                       yes



# IP address/port to listen to (default=all IP and port 21).

# Bind                          127.0.0.1,21

Bind                            127.0.0.1,8021
```

```
# Maximum bandwidth for anonymous users in KB/s

# AnonymousBandwidth            8



# Maximum bandwidth for *all* users (including anonymous) in KB/s
# Use AnonymousBandwidth *or* UserBandwidth, both makes no sense.

# UserBandwidth                 8



# File creation mask. <umask for files>:<umask for dirs> .
# 177:077 if you feel paranoid.

Umask                           133:022



# Minimum UID for an authenticated user to log in.

MinUID                          100



# Allow FXP transfers for authenticated users only.

AllowUserFXP                    yes



# Allow anonymous FXP for anonymous and non-anonymous users.

AllowAnonymousFXP               no



# Users can't delete/write files beginning with a dot ('.')
# even if they own them. If TrustedGID is enabled, this group
# will have access to dot-files, though.

ProhibitDotFilesWrite           no
```

```
# Prohibit *reading* of files beginning with a dot (.history, .ssh...)

ProhibitDotFilesRead        no



# Never overwrite files. When a file whoose name already exist is uploaded,
# it get automatically renamed to file.1, file.2, file.3, ...

AutoRename                  no



# Disallow anonymous users to upload new files (no = upload is allowed)

AnonymousCantUpload         no



# Only connections to this specific IP address are allowed to be
# non-anonymous. You can use this directive to open several public IPs for
# anonymous FTP, and keep a private firewalled IP for remote administration.
# You can also only allow a non-routable local IP (like 10.x.x.x) to
# authenticate, and keep a public anon-only FTP server on another IP.

#TrustedIP                  10.1.1.1



# If you want to add the PID to every logged line, uncomment the following
# line.

#LogPID                     yes



# Create an additional log file with transfers logged in a Apache-like format :
# fw.c9x.org - jedi [13/Dec/1975:19:36:39] "GET /ftp/linux.tar.bz2" 200 21809338
# This log file can then be processed by www traffic analyzers.

# AltLog                    clf:/var/log/pureftpd.log
# Create an additional log file with transfers logged in a format optimized
# for statistic reports.

# AltLog                                stats:/var/log/pureftpd.log
#AltLog                                 stats:/var/log/pureftpd.log



# Create an additional log file with transfers logged in the standard W3C
# format (compatible with most commercial log analyzers)

# AltLog                                w3c:/var/log/pureftpd.log



# Disallow the CHMOD command. Users can't change perms of their files.

#NoChmod                                yes



# Allow users to resume and upload files, but *NOT* to delete them.

#KeepAllFiles                           yes



# Automatically create home directories if they are missing

#CreateHomeDir                          yes



# Enable virtual quotas. The first number is the max number of files.
# The second number is the max size of megabytes.
# So 1000:10 limits every user to 1000 files and 10 Mb.

#Quota                                  1000:10
```

```
# If your pure-ftpd has been compiled with standalone support, you can change
# the location of the pid file. The default is /var/run/pure-ftpd.pid

#PIDFile                     /var/run/pure-ftpd.pid



# If your pure-ftpd has been compiled with pure-uploadscript support,
# this will make pure-ftpd write info about new uploads to
# /var/run/pure-ftpd.upload.pipe so pure-uploadscript can read it and
# spawn a script to handle the upload.

#CallUploadScript yes



# This option is useful with servers where anonymous upload is
# allowed. As /var/ftp is in /var, it save some space and protect
# the log files. When the partition is more that X percent full,
# new uploads are disallowed.

MaxDiskUsage                 99



# Set to 'yes' if you don't want your users to rename files.

#NoRename yes



# Be 'customer proof' : workaround against common customer mistakes like
# 'chmod 0 public_html', that are valid, but that could cause ignorant
# customers to lock their files, and then keep your technical support busy
# with silly issues. If you're sure all your users have some basic Unix
# knowledge, this feature is useless. If you're a hosting service, enable it.

CustomerProof yes




# Per-user concurrency limits. It will only work if the FTP server has
# been compiled with --with-peruserlimits (and this is the case on
# most binary distributions) .
# The format is : <max sessions per user>:<max anonymous sessions>
# For instance, 3:20 means that the same authenticated user can have 3 active
# sessions max. And there are 20 anonymous sessions max.

# PerUserLimits 3:20
```

9.2 pureftpd-ldap.conf

```

############################################
#                                          #
# Sample Pure-FTPd LDAP configuration file. #
# See README.LDAP for explanations.        #
#                                          #
############################################


# Optional : name of the LDAP server. Default : localhost

#LDAPServer ldap.c9x.org
LDAPServer localhost


# Optional : server port. Default : 389

LDAPPort    389


# Mandatory : the base DN to search accounts from. No default.

#LDAPBaseDN cn=Users,dc=c9x,dc=org
LDAPBaseDN dc=gdfz,dc=com


# Optional : who we should bind the server as.
#            Default : binds anonymously

#LDAPBindDN cn=Manager,dc=c9x,dc=org
LDAPBindDN cn=Manager,dc=gdfz,dc=com


# Password if we don't bind anonymously
# This configuration file should be only readable by root

#LDAPBindPW r00tPaSsw0rD
LDAPBindPW chen


# Optional : default UID, when there's no entry in an user object

# LDAPDefaultUID 500


# Optional : default GID, when there's no entry in an user object

# LDAPDefaultGID 100


# Filter to use to find the object that contains user info
# \L is replaced by the login the user is trying to log in as
# The default filter is (&(objectClass=posixAccount)(uid=\L))

# LDAPFilter (&(objectClass=posixAccount)(uid=\L))


# Attribute to get the home directory
# Default is homeDirectory (the standard attribute from posixAccount)

# LDAPHomeDir homeDirectory


# LDAP protocol version to use
# Version 3 (default) is mandatory with recent releases of OpenLDAP.

# LDAPVersion 3
```

9.3 pureftpd-mysql.conf

```

################################################
#                                              #
# Sample Pure-FTPd Mysql configuration file. #
# See README.MySQL for explanations.           #
#                                                  #
##############################################


# Optional : MySQL server name or IP. Don't define this for unix sockets.

#MYSQLServer     127.0.0.1

# Optional : MySQL port. Don't define this if a local unix socket is used.

#MYSQLPort       3306

# Optional : define the location of mysql.sock if the server runs on this host.

MYSQLSocket     /var/lib/mysql/mysql.sock

# Mandatory : user to bind the server as.

MYSQLUser       pureftpd

# Mandatory : user password. You must have a password.

MYSQLPassword   qKiscCbwbXAkWp.

# Mandatory : database to open.

MYSQLDatabase   pureftpd

# Mandatory : how passwords are stored
# Valid values are : "cleartext", "crypt", "md5" and "password"
# ("password" = MySQL password() function)
# You can also use "any" to try "crypt", "md5" *and* "password"

#MYSQLCrypt      cleartext
MYSQLCrypt      crypt

# In the following directives, parts of the strings are replaced at
# run-time before performing queries :
#
# \L is replaced by the login of the user trying to authenticate.
# \I is replaced by the IP address the user connected to.
# \P is replaced by the port number the user connected to.
# \R is replaced by the IP address the user connected from.
# \D is replaced by the remote IP address, as a long decimal number.
```

```
#
# Very complex queries can be performed using these substitution strings,
# especially for virtual hosting.

# Query to execute in order to fetch the password

MYSQLGetPW            SELECT Password FROM users WHERE User="\L"

# Query to execute in order to fetch the system user name or uid

MYSQLGetUID          SELECT Uid FROM users WHERE User="\L"

# Optional : default UID - if set this overrides MYSQLGetUID

#MYSQLDefaultUID 1000

# Query to execute in order to fetch the system user group or gid

MYSQLGetGID          SELECT Gid FROM users WHERE User="\L"

# Optional : default GID - if set this overrides MYSQLGetGID

#MYSQLDefaultGID 1000

# Query to execute in order to fetch the home directory

MYSQLGetDir          SELECT Dir FROM users WHERE User="\L"

# Optional : query to get the maximal number of files
# Pure-FTPd must have been compiled with virtual quotas support.

MySQLGetQTAFS    SELECT QuotaFiles FROM users WHERE User="\L"

# Optional : query to get the maximal disk usage (virtual quotas)
# The number should be in Megabytes.
# Pure-FTPd must have been compiled with virtual quotas support.

MySQLGetQTASZ    SELECT QuotaSize FROM users WHERE User="\L"



# Optional : ratios. The server has to be compiled with ratio support.

# MySQLGetRatioUL SELECT ULRatio FROM users WHERE User="\L"
# MySQLGetRatioDL SELECT DLRatio FROM users WHERE User="\L"
# Optional : bandwidth throttling.
# The server has to be compiled with throttling support.
# Values are in KB/s .

MySQLGetBandwidthUL SELECT ULBandwidth FROM users WHERE User="\L"
MySQLGetBandwidthDL SELECT DLBandwidth FROM users WHERE User="\L"

# Enable ~ expansion. NEVER ENABLE THIS BLINDLY UNLESS :
# 1) You know what you are doing.
# 2) Real and virtual users match.

# MySQLForceTildeExpansion 1


# If you upgraded your tables to transactionnal tables (Gemini,
# BerkeleyDB, Innobase...), you can enable SQL transactions to
# avoid races. Leave this commented if you are using the
# traditionnal MyIsam databases or old (< 3.23.x) MySQL versions.

# MySQLTransactions On
```

9.4 pureftpd-pgsql.conf

```
##################################################
#                                                #
# Sample Pure-FTPd PostgreSQL configuration file. #
# See README.PGSQL for explanations.             #
#                                                #
##################################################


# If PostgreSQL listens to a TCP socket
#PGSQLServer     localhost
PGSQLServer     localhost
#PGSQLPort       5432
PGSQLPort       5432
```

```
# *or* if PostgreSQL can only be reached through a local Unix socket
# PGSQLServer          /tmp
# PGSQLPort            .s.PGSQL.5432

# Mandatory : user to bind the server as.
#PGSQLUser             postgres
PGSQLUser             pureftpd

# Mandatory : user password. You *must* have a password.
#PGSQLPassword      rootpw
PGSQLPassword      pureftpd

# Mandatory : database to open.
#PGSQLDatabase      pureftpd
PGSQLDatabase      pureftpd

# Mandatory : how passwords are stored
# Valid values are : "cleartext", "crypt", "md5" or "any"
#PGSQLCrypt            cleartext
PGSQLCrypt            crypt

# In the following directives, parts of the strings are replaced at
# run-time before performing queries :
#
# \L is replaced by the login of the user trying to authenticate.
# \I is replaced by the IP address the user connected to.
# \P is replaced by the port number the user connected to.
# \R is replaced by the IP address the user connected from.
# \D is replaced by the remote IP address, as a long decimal number.
#
# Very complex queries can be performed using these substitution strings,
# especially for virtual hosting.


# Query to execute in order to fetch the password

PGSQLGetPW            SELECT Password FROM users WHERE User='\L'


# Query to execute in order to fetch the system user name or uid

PGSQLGetUID          SELECT Uid FROM users WHERE User='\L'
```

```
# Optional : default UID - if set this overrides PGSQLGetUID

#PGSQLDefaultUID 1000


# Query to execute in order to fetch the system user group or gid

PGSQLGetGID          SELECT Gid FROM users WHERE User='\L'


# Optional : default GID - if set this overrides PGSQLGetGID

#PGSQLDefaultGID 1000


# Query to execute in order to fetch the home directory

PGSQLGetDir          SELECT Dir FROM users WHERE User='\L'


# Optional : query to get the maximal number of files
# Pure-FTPd must have been compiled with virtual quotas support.

# PGSQLGetQTAFS    SELECT QuotaFiles FROM users WHERE User='\L'


# Optional : query to get the maximal disk usage (virtual quotas)
# The number should be in Megabytes.
# Pure-FTPd must have been compiled with virtual quotas support.

# PGSQLGetQTASZ    SELECT QuotaSize FROM users WHERE User='\L'


# Optional : ratios. The server has to be compiled with ratio support.

# PGSQLGetRatioUL SELECT ULRatio FROM users WHERE User='\L'
# PGSQLGetRatioDL SELECT DLRatio FROM users WHERE User='\L'


# Optional : bandwidth throttling.
# The server has to be compiled with throttling support.
# Values are in KB/s .

# PGSQLGetBandwidthUL SELECT ULBandwidth FROM users WHERE User='\L'
# PGSQLGetBandwidthDL SELECT DLBandwidth FROM users WHERE User='\L'
```

9.5 pureftpd.passwd

```
[root@linux etc]# cat pureftpd.passwd
qqqq:$1$suA.WBZ0$Uu/05AtMi/4cNdhg9gKjP/:505:505::/home/web/./::::::::::
9812:$1$4.iPvGE0$lY5CEVYLde.Mb9QWNu.so0:505:505::/home/www.9812.net/./::::::::::
```

10 FAQ

10.1 不能访问 http://www.pureftpd.org/

http://www.pureftpd.org/ 网站被我们政府封了，你可以使用代理服务器
代理服务器列表：http://www.salala.com/proxy_index.htm

10.2 目录与 OpenSource RDBMS 比较

性能：
  读速度：OpenLDAP > MySQL > PostgreSQL
  写入/修改：MySQL > PostgreSQL > OpenLDAP
集群：OpenLDAP > PostgreSQL> MySQL（不支持集群）
海量存储：PostgreSQL > OpenLDAP（分布式存储）> MySQL

10.3 产生 Crypt 密码

10.3.1 使用 C 产生

```
[root@linux root]# cat crypt.c
/*
Netkiller 2003-06-27 crypt.c
char *crypt(const char *key, const char *salt);
*/

#include <unistd.h>
main(){
	char key[256];
	char salt[64];
	char passwd[256];
```

```
        printf("key:");
        scanf("%s",&key);
        printf("salt:");
        scanf("%s",&salt);

        sprintf(passwd,"passwd:%s\n",crypt(key,salt));

        printf(passwd);
}

[root@linux root]# gcc -o crypt -s crypt.c –lcrypt
[root@linux root]# ./crypt
key:chen
salt:salt
passwd:sa0hRW/W3DLvQ
[root@linux root]#
```

10.3.2 使用 PHP 产生

```
# cat des.php
<html>
<p>DES  密码产生器</p>
<form method=post action=des.php>
<p>password:<input name=passwd type=text size=20></p>
<input type=submit value=submit>
</form>
<?
$enpw=crypt($passwd);
echo "password is: $enpw";
?>

[root@linux root]# wget http://home.9812.net/linux/download/myphp/site-2.1.0.tar.gz
[root@linux root]#tar zxvf site-2.1.0.tar.gz
[root@linux root]#cp –r site /usr/local/apache/htdocs
[root@linux root]#lynx http://localhost/site
```

10.3.3 使用 perl 产生

```
perl -e 'print("userPassword: ".crypt("secret","salt")."\n");'
```

产生的 DES 密码，同样也可以用于 OpenLDAP 的管理员密码

```
# vi /etc/openldap/slapd.conf
rootpw                    {crypt}ijFYNcSNctBYg
```

10.3.4 使用 SQL 语句产生

```
select encrypt('password');

mysql> select encrypt('password');
+---------------------+
| encrypt('password') |
+---------------------+
| WXvvG0CWY7v5I       |
+---------------------+
1 row in set (0.00 sec)

mysql>
```

10.3.5 使用 Java 产生

第一种方法：
Crypt.java

```
Import netkiller. Security;
Crypt pw = new Crypt();
String passwd = pw.crypt(“passwd”,”salt”);
System.out.println(passwd);
```

关于 JAVA 的 Crypt 包请与我联系

第二种方法：
使用 PostgreSQL JDBC 中提供的 org.postgresql.util.UnixCrypt 产生 crypt。

```
Class   postgresql.util.UnixCrypt
java.lang.Object
        |
        +----postgresql.util.UnixCrypt
```

公共类 UnixCrypt 扩展 Object
这个类为我们提供了在通过网络流传输口令时的加密的功能
包含静态方法用于加密口令和与 Unix 加密的口令比较.
参阅 John Dumas 的 Java Crypt (加密)页面获取原始代码.
http://www.zeh.com/local/jfd/crypt.html

方法

public static final String crypt(String salt, String original)

加密给出了明文口令和一个"种子"("salt")的口令.

参数:

salt - 一个两字符字串代表的所用的种子， 用以向加密引擎说明加密的不同方

式．如果你要生成一个新的密文那么这个值应该是随机生成的.

original - 待加密口令.

返回:

一个字串, 先是 2 字符的种子, 然后跟着密文口令.

方法：

1. 安装 PostgreSQL JDBC，请到 http://www.postgresql.org 下载
2. 将 JDBC 的.jar 文件加到 JAVA 的 CLASSPATH 中
3. 新建 JAVA 文件。
4. 编译 javac crypt.java
5. 运行 JAVA CLASS 文件 java your-package.your-class
   java crypt

```
import org.postgresql.util.UnixCrypt;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.IOException;

public class crypt {
    public static void main(String[] args) throws IOException {
    String password;
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter the password to encrypt. Your password"+
                " will be echoed on the screen,");
    System.out.println("please ensure nobody is looking.");
    System.out.print("password :>");
    password=br.readLine();
    System.out.println(UnixCrypt.crypt(password));
    };
};
```

10.4 产生 MD5 字串

10.4.1 使用 C 产生

```
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#include <time.h>
#include <errno.h>
#include <string.h>
#include <sys/socket.h>
```

```
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include "../md5/md5.h"

#define T1 0xd76aa478
#define T2 0xe8c7b756
#define T3 0x242070db
#define T4 0xc1bdceee
#define T5 0xf57c0faf
#define T6 0x4787c62a
#define T7 0xa8304613
#define T8 0xfd469501
#define T9 0x698098d8
#define T10 0x8b44f7af
#define T11 0xffff5bb1
#define T12 0x895cd7be
#define T13 0x6b901122
#define T14 0xfd987193
#define T15 0xa679438e
#define T16 0x49b40821
#define T17 0xf61e2562
#define T18 0xc040b340
#define T19 0x265e5a51
#define T20 0xe9b6c7aa
#define T21 0xd62f105d
#define T22 0x02441453
#define T23 0xd8a1e681
#define T24 0xe7d3fbc8
#define T25 0x21e1cde6
#define T26 0xc33707d6
#define T27 0xf4d50d87
#define T28 0x455a14ed
#define T29 0xa9e3e905
#define T30 0xfcefa3f8
#define T31 0x676f02d9
#define T32 0x8d2a4c8a
#define T33 0xfffa3942
#define T34 0x8771f681
#define T35 0x6d9d6122
#define T36 0xfde5380c
#define T37 0xa4beea44
#define T38 0x4bdecfa9
```

```
#define T39 0xf6bb4b60
#define T40 0xbebfbc70
#define T41 0x289b7ec6
#define T42 0xeaa127fa
#define T43 0xd4ef3085
#define T44 0x04881d05
#define T45 0xd9d4d039
#define T46 0xe6db99e5
#define T47 0x1fa27cf8
#define T48 0xc4ac5665
#define T49 0xf4292244
#define T50 0x432aff97
#define T51 0xab9423a7
#define T52 0xfc93a039
#define T53 0x655b59c3
#define T54 0x8f0ccc92
#define T55 0xffeff47d
#define T56 0x85845dd1
#define T57 0x6fa87e4f
#define T58 0xfe2ce6e0
#define T59 0xa3014314
#define T60 0x4e0811a1
#define T61 0xf7537e82
#define T62 0xbd3af235
#define T63 0x2ad7d2bb
#define T64 0xeb86d391

static void md5_process(md5_state_t *pms, const md5_byte_t *data /*[64]*/)
{
md5_word_t
a = pms->abcd[0], b = pms->abcd[1],
c = pms->abcd[2], d = pms->abcd[3];
md5_word_t t;

#ifndef ARCH_IS_BIG_ENDIAN
# define ARCH_IS_BIG_ENDIAN 1 /* slower, default implementation */
#endif
#if ARCH_IS_BIG_ENDIAN

/*
* On big-endian machines, we must arrange the bytes in the right
* order. (This also works on machines of unknown byte order.)
*/
md5_word_t X[16];
```

```
const md5_byte_t *xp = data;
int i;

for (i = 0; i < 16; ++i, xp += 4)
X[i] = xp[0] + (xp[1] << 8) + (xp[2] << 16) + (xp[3] << 24);

#else /* !ARCH_IS_BIG_ENDIAN */

/*
* On little-endian machines, we can process properly aligned data
* without copying it.
*/
md5_word_t xbuf[16];
const md5_word_t *X;

if (!((data - (const md5_byte_t *)0) & 3)) {
/* data are properly aligned */
X = (const md5_word_t *)data;
} else {
/* not aligned */
memcpy(xbuf, data, 64);
X = xbuf;
}
#endif

#define ROTATE_LEFT(x, n) (((x) << (n)) | ((x) >> (32 - (n))))

/* Round 1. */
/* Let [abcd k s i] denote the operation
a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */
#define F(x, y, z) (((x) & (y)) | (~(x) & (z)))
#define SET(a, b, c, d, k, s, Ti)\
t = a + F(b,c,d) + X[k] + Ti;\
a = ROTATE_LEFT(t, s) + b
/* Do the following 16 operations. */
SET(a, b, c, d, 0, 7, T1);
SET(d, a, b, c, 1, 12, T2);
SET(c, d, a, b, 2, 17, T3);
SET(b, c, d, a, 3, 22, T4);
SET(a, b, c, d, 4, 7, T5);
SET(d, a, b, c, 5, 12, T6);
SET(c, d, a, b, 6, 17, T7);
SET(b, c, d, a, 7, 22, T8);
SET(a, b, c, d, 8, 7, T9);
SET(d, a, b, c, 9, 12, T10);
SET(c, d, a, b, 10, 17, T11);
SET(b, c, d, a, 11, 22, T12);
SET(a, b, c, d, 12, 7, T13);
SET(d, a, b, c, 13, 12, T14);
SET(c, d, a, b, 14, 17, T15);
SET(b, c, d, a, 15, 22, T16);
#undef SET

/* Round 2. */
/* Let [abcd k s i] denote the operation
a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */
#define G(x, y, z) (((x) & (z)) | ((y) & ~(z)))
#define SET(a, b, c, d, k, s, Ti)\
t = a + G(b,c,d) + X[k] + Ti;\
a = ROTATE_LEFT(t, s) + b
/* Do the following 16 operations. */
SET(a, b, c, d, 1, 5, T17);
SET(d, a, b, c, 6, 9, T18);
SET(c, d, a, b, 11, 14, T19);
SET(b, c, d, a, 0, 20, T20);
SET(a, b, c, d, 5, 5, T21);
SET(d, a, b, c, 10, 9, T22);
SET(c, d, a, b, 15, 14, T23);
SET(b, c, d, a, 4, 20, T24);
SET(a, b, c, d, 9, 5, T25);
SET(d, a, b, c, 14, 9, T26);
SET(c, d, a, b, 3, 14, T27);
SET(b, c, d, a, 8, 20, T28);
SET(a, b, c, d, 13, 5, T29);
SET(d, a, b, c, 2, 9, T30);
SET(c, d, a, b, 7, 14, T31);
SET(b, c, d, a, 12, 20, T32);
#undef SET

/* Round 3. */
/* Let [abcd k s t] denote the operation
a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */
#define H(x, y, z) ((x) ^ (y) ^ (z))
#define SET(a, b, c, d, k, s, Ti)\
t = a + H(b,c,d) + X[k] + Ti;\
a = ROTATE_LEFT(t, s) + b
/* Do the following 16 operations. */
SET(a, b, c, d, 5, 4, T33);
SET(d, a, b, c, 8, 11, T34);
SET(c, d, a, b, 11, 16, T35);
SET(b, c, d, a, 14, 23, T36);
SET(a, b, c, d, 1, 4, T37);
SET(d, a, b, c, 4, 11, T38);
SET(c, d, a, b, 7, 16, T39);
SET(b, c, d, a, 10, 23, T40);
SET(a, b, c, d, 13, 4, T41);
SET(d, a, b, c, 0, 11, T42);
SET(c, d, a, b, 3, 16, T43);
SET(b, c, d, a, 6, 23, T44);
SET(a, b, c, d, 9, 4, T45);
SET(d, a, b, c, 12, 11, T46);
SET(c, d, a, b, 15, 16, T47);
SET(b, c, d, a, 2, 23, T48);
#undef SET

/* Round 4. */
/* Let [abcd k s t] denote the operation
a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */
#define I(x, y, z) ((y) ^ ((x) | ~(z)))
#define SET(a, b, c, d, k, s, Ti)\
t = a + I(b,c,d) + X[k] + Ti;\
a = ROTATE_LEFT(t, s) + b
/* Do the following 16 operations. */
SET(a, b, c, d, 0, 6, T49);
SET(d, a, b, c, 7, 10, T50);
SET(c, d, a, b, 14, 15, T51);
SET(b, c, d, a, 5, 21, T52);
SET(a, b, c, d, 12, 6, T53);
SET(d, a, b, c, 3, 10, T54);
SET(c, d, a, b, 10, 15, T55);
SET(b, c, d, a, 1, 21, T56);
SET(a, b, c, d, 8, 6, T57);
SET(d, a, b, c, 15, 10, T58);
SET(c, d, a, b, 6, 15, T59);
SET(b, c, d, a, 13, 21, T60);
SET(a, b, c, d, 4, 6, T61);
SET(d, a, b, c, 11, 10, T62);
SET(c, d, a, b, 2, 15, T63);
SET(b, c, d, a, 9, 21, T64);
#undef SET

/* Then perform the following additions. (That is increment each
of the four registers by the value it had before this block
was started.) */
pms->abcd[0] += a;
pms->abcd[1] += b;
pms->abcd[2] += c;
pms->abcd[3] += d;
}

void md5_init(md5_state_t *pms)
{
pms->count[0] = pms->count[1] = 0;
pms->abcd[0] = 0x67452301;
pms->abcd[1] = 0xefcdab89;
pms->abcd[2] = 0x98badcfe;
pms->abcd[3] = 0x10325476;
}

void md5_append(md5_state_t *pms, const md5_byte_t *data, int nbytes)
{
const md5_byte_t *p = data;
int left = nbytes;
int offset = (pms->count[0] >> 3) & 63;
md5_word_t nbits = (md5_word_t)(nbytes << 3);

if (nbytes <= 0) return;

/* Update the message length. */
pms->count[1] += nbytes >> 29;
pms->count[0] += nbits;
if (pms->count[0] < nbits) pms->count[1]++;

/* Process an initial partial block. */
if (offset) {
int copy = (offset + nbytes > 64 ? 64 - offset : nbytes);

memcpy(pms->buf + offset, p, copy);
if (offset + copy < 64) return;
p += copy;
left -= copy;
md5_process(pms, pms->buf);
}

/* Process full blocks. */
for (; left >= 64; p += 64, left -= 64)
```

```
md5_process(pms, p);

/* Process a final partial block. */
if (left)
memcpy(pms->buf, p, left);
}

void md5_finish(md5_state_t *pms, md5_byte_t digest[16])
{
static const md5_byte_t pad[64] = {
0x80, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
};
md5_byte_t data[8];
int i;

/* Save the length before padding. */
for (i = 0; i < 8; ++i)
data[i] = (md5_byte_t)(pms->count[i >> 2] >> ((i & 3) << 3));
/* Pad to 56 bytes mod 64. */
md5_append(pms, pad, ((55 - (pms->count[0] >> 3)) & 63) + 1);
/* Append the length. */
md5_append(pms, data, 8);
for (i = 0; i < 16; ++i)
digest[i] = (md5_byte_t)(pms->abcd[i >> 2] >> ((i & 3) << 3));
}

void md5_passwd(char *oldpasswd, char *md5_passwd)
{
md5_state_t state;
md5_byte_t digest[16];
int di;

md5_init(&state);
md5_append(&state, (const md5_byte_t *)oldpasswd, strlen(oldpasswd));
md5_finish(&state, digest);

sprintf(md5_passwd,"\0");
for(di=0; di<16; di++)
sprintf(md5_passwd,"%s%02x",md5_passwd,digest[di]);

}
```

```
main(int argc, char **argv)
{
char md5p[33];

if (argc<1 || argc>2 ) perror("error param");
md5_passwd(argv[1], md5p);
printf("pass=%s, md5pass=%s\n", argv[1], md5p);
}
```

10.4.2 使用 PHP 产生

```
# cat md5.php
<html>
<p>MD5  密码产生器</p>
<form method=post action=des.php>
<p>password:<input name=passwd type=text size=20></p>
<input type=submit value=submit>
</form>
<?
$enpw=md5($passwd);
echo "password is: $enpw";
?>
```

```
[root@linux root]# wget http://home.9812.net/linux/download/myphp/site-2.1.0.tar.gz
[root@linux root]#tar zxvf site-2.1.0.tar.gz
[root@linux root]#cp –r site /usr/local/apache/htdocs
```

10.4.3 使用 SQL 语句产生

```
select md5('password');
```

```
[chen@linux chen]$ mysql
Welcome to the MySQL monitor.    Commands end with ; or \g.
Your MySQL connection id is 11947 to server version: 4.0.13-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select md5('chen');
+----------------------------------+
| md5('chen')                      |
+----------------------------------+
| a1a8887793acfc199182a649e905daab |
+----------------------------------+
1 row in set (0.00 sec)

mysql>


mysql> select md5('chen') as passwd;
+----------------------------------+
| passwd                           |
+----------------------------------+
| a1a8887793acfc199182a649e905daab |
+----------------------------------+
1 row in set (0.00 sec)

mysql>
```

10.4.4 使用 Java 产生

```
/************************************************
MD5 算法的 Java Bean
@author:Topcat Tuppin
Last Modified:10,Mar,2001
*************************************************/
package netkiller.security;
import java.lang.reflect.*;
/*************************************************
md5 类实现了 RSA Data Security, Inc.在提交给 IETF
的 RFC1321 中的 MD5 message-digest 算法。
*************************************************/

public class MD5 {
    /* 下面这些 S11-S44 实际上是一个 4*4 的矩阵，在原始的 C 实现中是用#define 实现的，
    这里把它们实现成为 static final 是表示了只读，切能在同一个进程空间内的多个
    Instance 间共享*/
        static final int S11 = 7;
        static final int S12 = 12;
        static final int S13 = 17;
        static final int S14 = 22;

        static final int S21 = 5;
        static final int S22 = 9;
        static final int S23 = 14;
        static final int S24 = 20;

        static final int S31 = 4;
        static final int S32 = 11;
        static final int S33 = 16;
        static final int S34 = 23;

        static final int S41 = 6;
        static final int S42 = 10;
        static final int S43 = 15;
        static final int S44 = 21;

        static final byte[] PADDING = { -128, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
        /*  下面的三个成员是 MD5 计算过程中用到的 3 个核心数据，在原始的 C 实现中
           被定义到 MD5_CTX 结构中

         */
        private long[] state = new long[4];    // state (ABCD)
        private long[] count = new long[2];    // number of bits, modulo 2^64 (lsb first)
        private byte[] buffer = new byte[64]; // input buffer

    /* digestHexStr 是 MD5 的唯一一个公共成员，是最新一次计算结果的
       16 进制 ASCII 表示.
    */
        public String digestHexStr;

        /* digest,是最新一次计算结果的 2 进制内部表示，表示 128bit 的 MD5 值.
    */
        private byte[] digest = new byte[16];

    /*
      getMD5ofStr 是类 MD5 最主要的公共方法，入口参数是你想要进行 MD5 变换的字符
串
      返回的是变换完的结果，这个结果是从公共成员 digestHexStr 取得的．
    */
        public String getMD5ofStr(String inbuf) {
                md5Init();
                md5Update(inbuf.getBytes(), inbuf.length());
                md5Final();
                digestHexStr = "";
                for (int i = 0; i < 16; i++) {
```

```
            digestHexStr += byteHEX(digest[i]);
        }
        return digestHexStr;

    }
    //  这是 MD5 这个类的标准构造函数，JavaBean 要求有一个 public 的并且没有参数
的构造函数
    public MD5() {
        md5Init();

        return;
    }




    /* md5Init 是一个初始化函数，初始化核心变量，装入标准的幻数  */
    private void md5Init() {
        count[0] = 0L;
        count[1] = 0L;
        ///* Load magic initialization constants.

        state[0] = 0x67452301L;
        state[1] = 0xefcdab89L;
        state[2] = 0x98badcfeL;
        state[3] = 0x10325476L;

        return;
    }
    /* F, G, H ,I  是 4 个基本的 MD5 函数，在原始的 MD5 的 C 实现中，由于它们是
    简单的位运算，可能出于效率的考虑把它们实现成了宏，在 java 中，我们把它们
     实现成了 private 方法，名字保持了原来 C 中的。  */

    private long F(long x, long y, long z) {
        return (x & y) | ((~x) & z);

    }
    private long G(long x, long y, long z) {
        return (x & z) | (y & (~z));

    }
    private long H(long x, long y, long z) {
        return x ^ y ^ z;
    }
```

```
  private long I(long x, long y, long z) {
          return y ^ (x | (~z));
  }

/*
    FF,GG,HH 和 II 将调用 F,G,H,I 进行近一步变换
    FF, GG, HH, and II transformations for rounds 1, 2, 3, and 4.
    Rotation is separate from addition to prevent recomputation.
*/

  private long FF(long a, long b, long c, long d, long x, long s,
          long ac) {
          a += F (b, c, d) + x + ac;
          a = ((int) a << s) | ((int) a >>> (32 - s));
          a += b;
          return a;
  }

  private long GG(long a, long b, long c, long d, long x, long s,
          long ac) {
          a += G (b, c, d) + x + ac;
          a = ((int) a << s) | ((int) a >>> (32 - s));
          a += b;
          return a;
  }
  private long HH(long a, long b, long c, long d, long x, long s,
          long ac) {
          a += H (b, c, d) + x + ac;
          a = ((int) a << s) | ((int) a >>> (32 - s));
          a += b;
          return a;
  }
  private long II(long a, long b, long c, long d, long x, long s,
          long ac) {
          a += I (b, c, d) + x + ac;
          a = ((int) a << s) | ((int) a >>> (32 - s));
          a += b;
          return a;
  }
  /*
   md5Update 是 MD5 的主计算过程，inbuf 是要变换的字节串，inputlen 是长度，这个
   函数由 getMD5ofStr 调用，调用之前需要调用 md5init，因此把它设计成 private 的
```

```
*/
private void md5Update(byte[] inbuf, int inputLen) {

        int i, index, partLen;
        byte[] block = new byte[64];
        index = (int)(count[0] >>> 3) & 0x3F;
        // /* Update number of bits */
        if ((count[0] += (inputLen << 3)) < (inputLen << 3))
                count[1]++;
        count[1] += (inputLen >>> 29);

        partLen = 64 - index;

        // Transform as many times as possible.
        if (inputLen >= partLen) {
                md5Memcpy(buffer, inbuf, index, 0, partLen);
                md5Transform(buffer);

                for (i = partLen; i + 63 < inputLen; i += 64) {

                        md5Memcpy(block, inbuf, 0, i, 64);
                        md5Transform (block);
                }
                index = 0;

        } else

                i = 0;

        ///* Buffer remaining input */
        md5Memcpy(buffer, inbuf, index, i, inputLen - i);

}

/*
  md5Final 整理和填写输出结果
*/
private void md5Final () {
        byte[] bits = new byte[8];
        int index, padLen;

        ///* Save number of bits */
        Encode (bits, count, 8);
        ///* Pad out to 56 mod 64.
        index = (int)(count[0] >>> 3) & 0x3f;
        padLen = (index < 56) ? (56 - index) : (120 - index);
        md5Update (PADDING, padLen);

        ///* Append length (before padding) */
        md5Update(bits, 8);

        ///* Store state in digest */
        Encode (digest, state, 16);

    }

    /*  md5Memcpy 是一个内部使用的 byte 数组的块拷贝函数，从 input 的 inpos 开始把 len 长度的
      字节拷贝到 output 的 outpos 位置开始
    */

    private void md5Memcpy (byte[] output, byte[] input,
        int outpos, int inpos, int len)
    {
        int i;

        for (i = 0; i < len; i++)
            output[outpos + i] = input[inpos + i];
    }

    /*
      md5Transform 是 MD5 核心变换程序，有 md5Update 调用，block 是分块的原始字节
    */
    private void md5Transform (byte block[]) {
        long a = state[0], b = state[1], c = state[2], d = state[3];
        long[] x = new long[16];

        Decode (x, block, 64);

        /* Round 1 */
        a = FF (a, b, c, d, x[0], S11, 0xd76aa478L); /* 1 */
        d = FF (d, a, b, c, x[1], S12, 0xe8c7b756L); /* 2 */
        c = FF (c, d, a, b, x[2], S13, 0x242070dbL); /* 3 */
        b = FF (b, c, d, a, x[3], S14, 0xc1bdceeeL); /* 4 */
        a = FF (a, b, c, d, x[4], S11, 0xf57c0fafL); /* 5 */
        d = FF (d, a, b, c, x[5], S12, 0x4787c62aL); /* 6 */
```

```
c = FF (c, d, a, b, x[6], S13, 0xa8304613L); /* 7 */
b = FF (b, c, d, a, x[7], S14, 0xfd469501L); /* 8 */
a = FF (a, b, c, d, x[8], S11, 0x698098d8L); /* 9 */
d = FF (d, a, b, c, x[9], S12, 0x8b44f7afL); /* 10 */
c = FF (c, d, a, b, x[10], S13, 0xffff5bb1L); /* 11 */
b = FF (b, c, d, a, x[11], S14, 0x895cd7beL); /* 12 */
a = FF (a, b, c, d, x[12], S11, 0x6b901122L); /* 13 */
d = FF (d, a, b, c, x[13], S12, 0xfd987193L); /* 14 */
c = FF (c, d, a, b, x[14], S13, 0xa679438eL); /* 15 */
b = FF (b, c, d, a, x[15], S14, 0x49b40821L); /* 16 */

/* Round 2 */
a = GG (a, b, c, d, x[1], S21, 0xf61e2562L); /* 17 */
d = GG (d, a, b, c, x[6], S22, 0xc040b340L); /* 18 */
c = GG (c, d, a, b, x[11], S23, 0x265e5a51L); /* 19 */
b = GG (b, c, d, a, x[0], S24, 0xe9b6c7aaL); /* 20 */
a = GG (a, b, c, d, x[5], S21, 0xd62f105dL); /* 21 */
d = GG (d, a, b, c, x[10], S22, 0x2441453L); /* 22 */
c = GG (c, d, a, b, x[15], S23, 0xd8a1e681L); /* 23 */
b = GG (b, c, d, a, x[4], S24, 0xe7d3fbc8L); /* 24 */
a = GG (a, b, c, d, x[9], S21, 0x21e1cde6L); /* 25 */
d = GG (d, a, b, c, x[14], S22, 0xc33707d6L); /* 26 */
c = GG (c, d, a, b, x[3], S23, 0xf4d50d87L); /* 27 */
b = GG (b, c, d, a, x[8], S24, 0x455a14edL); /* 28 */
a = GG (a, b, c, d, x[13], S21, 0xa9e3e905L); /* 29 */
d = GG (d, a, b, c, x[2], S22, 0xfcefa3f8L); /* 30 */
c = GG (c, d, a, b, x[7], S23, 0x676f02d9L); /* 31 */
b = GG (b, c, d, a, x[12], S24, 0x8d2a4c8aL); /* 32 */

/* Round 3 */
a = HH (a, b, c, d, x[5], S31, 0xfffa3942L); /* 33 */
d = HH (d, a, b, c, x[8], S32, 0x8771f681L); /* 34 */
c = HH (c, d, a, b, x[11], S33, 0x6d9d6122L); /* 35 */
b = HH (b, c, d, a, x[14], S34, 0xfde5380cL); /* 36 */
a = HH (a, b, c, d, x[1], S31, 0xa4beea44L); /* 37 */
d = HH (d, a, b, c, x[4], S32, 0x4bdecfa9L); /* 38 */
c = HH (c, d, a, b, x[7], S33, 0xf6bb4b60L); /* 39 */
b = HH (b, c, d, a, x[10], S34, 0xbebfbc70L); /* 40 */
a = HH (a, b, c, d, x[13], S31, 0x289b7ec6L); /* 41 */
d = HH (d, a, b, c, x[0], S32, 0xeaa127faL); /* 42 */
c = HH (c, d, a, b, x[3], S33, 0xd4ef3085L); /* 43 */
b = HH (b, c, d, a, x[6], S34, 0x4881d05L); /* 44 */
a = HH (a, b, c, d, x[9], S31, 0xd9d4d039L); /* 45 */
d = HH (d, a, b, c, x[12], S32, 0xe6db99e5L); /* 46 */
```

```
    c = HH (c, d, a, b, x[15], S33, 0x1fa27cf8L); /* 47 */
    b = HH (b, c, d, a, x[2], S34, 0xc4ac5665L); /* 48 */

    /* Round 4 */
    a = II (a, b, c, d, x[0], S41, 0xf4292244L); /* 49 */
    d = II (d, a, b, c, x[7], S42, 0x432aff97L); /* 50 */
    c = II (c, d, a, b, x[14], S43, 0xab9423a7L); /* 51 */
    b = II (b, c, d, a, x[5], S44, 0xfc93a039L); /* 52 */
    a = II (a, b, c, d, x[12], S41, 0x655b59c3L); /* 53 */
    d = II (d, a, b, c, x[3], S42, 0x8f0ccc92L); /* 54 */
    c = II (c, d, a, b, x[10], S43, 0xffeff47dL); /* 55 */
    b = II (b, c, d, a, x[1], S44, 0x85845dd1L); /* 56 */
    a = II (a, b, c, d, x[8], S41, 0x6fa87e4fL); /* 57 */
    d = II (d, a, b, c, x[15], S42, 0xfe2ce6e0L); /* 58 */
    c = II (c, d, a, b, x[6], S43, 0xa3014314L); /* 59 */
    b = II (b, c, d, a, x[13], S44, 0x4e0811a1L); /* 60 */
    a = II (a, b, c, d, x[4], S41, 0xf7537e82L); /* 61 */
    d = II (d, a, b, c, x[11], S42, 0xbd3af235L); /* 62 */
    c = II (c, d, a, b, x[2], S43, 0x2ad7d2bbL); /* 63 */
    b = II (b, c, d, a, x[9], S44, 0xeb86d391L); /* 64 */

    state[0] += a;
    state[1] += b;
    state[2] += c;
    state[3] += d;

}

/*Encode 把 long 数组按顺序拆成 byte 数组，因为 java 的 long 类型是 64bit 的，
  只拆低 32bit，以适应原始 C 实现的用途
*/
private void Encode (byte[] output, long[] input, int len) {
    int i, j;

    for (i = 0, j = 0; j < len; i++, j += 4) {
        output[j] = (byte)(input[i] & 0xffL);
        output[j + 1] = (byte)((input[i] >>> 8) & 0xffL);
        output[j + 2] = (byte)((input[i] >>> 16) & 0xffL);
        output[j + 3] = (byte)((input[i] >>> 24) & 0xffL);
    }
}

/*Decode 把 byte 数组按顺序合成成 long 数组，因为 java 的 long 类型是 64bit 的，
  只合成低 32bit，高 32bit 清零，以适应原始 C 实现的用途
```

```
        */
        private void Decode (long[] output, byte[] input, int len) {
                int i, j;


                for (i = 0, j = 0; j < len; i++, j += 4)
                        output[i] = b2iu(input[j]) |
                                (b2iu(input[j + 1]) << 8) |
                                (b2iu(input[j + 2]) << 16) |
                                (b2iu(input[j + 3]) << 24);

                return;
        }

        /*
          b2iu 是我写的一个把 byte 按照不考虑正负号的原则的 " 升位 " 程序，因为 java
没有 unsigned 运算
        */
        public static long b2iu(byte b) {
                return b < 0 ? b & 0x7F + 128 : b;
        }

    /*byteHEX()，用来把一个 byte 类型的数转换成十六进制的 ASCII 表示，
      因为 java 中的 byte 的 toString 无法实现这一点，我们又没有 C 语言中的
      sprintf(outbuf,"%02X",ib)
    */
        public static String byteHEX(byte ib) {
                char[] Digit = { '0','1','2','3','4','5','6','7','8','9',
                'A','B','C','D','E','F' };
                char [] ob = new char[2];
                ob[0] = Digit[(ib >>> 4) & 0X0F];
                ob[1] = Digit[ib & 0X0F];
                String s = new String(ob);
                return s;
        }

        public String getMD5String(String md5){
                return getMD5ofStr(md5).toLowerCase();
        }

        public static void main(String args[]) {


                MD5 m = new MD5();
                if (Array.getLength(args) == 0) {     //如果没有参数，执行标准的 Test Suite

                        System.out.println("MD5 Test suite:");
                        System.out.println("MD5(\"\"):"+m.getMD5ofStr(""));
                        System.out.println("MD5(\"a\"):"+m.getMD5ofStr("a"));
                        System.out.println("MD5(\"abc\"):"+m.getMD5ofStr("abc"));
                        System.out.println("MD5(\"message
digest\"):"+m.getMD5ofStr("message digest"));
                        System.out.println("MD5(\"abcdefghijklmnopqrstuvwxyz\"):"+
                                m.getMD5ofStr("abcdefghijklmnopqrstuvwxyz"));

        System.out.println("MD5(\"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrst
uvwxyz0123456789\"):"+

        m.getMD5ofStr("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01
23456789"));
                }
                else
                                System.out.println("MD5(" + args[0] + ")=" +
m.getMD5ofStr(args[0]));


        }

}
```

10.5 Openldap 的常建问题

10.5.1使用组织单元

建议您使用组织单元，规划 LDAP。所有条目全放于 dn 下，太乱，不易管理、维护。

例子：

```
        LDAPBaseDN      ou=pureftpd,dc=9812,dc=net
        LDAPBindDN      cn=Admin,ou=pureftpd,dc=9812,dc=net
        LDAPBindPW      your-passwd
```

10.5.2安全方面

对于 userPassword: { }建议使用 userPassword: { md5 } or userPassword: {crypt}

设置 ACL 权限

```
# database access control definitions
```

```
access to attr=userPassword
        by self write
        by anonymous auth
        by dn.base="cn=Admin,ou=pureftpd,dc=example,dc=com" write
        by * none
```

11 参考资料

OpenLDAP: http://www.openldap.org
LDAP Schema: http://ldap.akbkhome.com/
PostgreSQL: http://www.pgsqldb.org
http://pureftpd.sourceforge.net/documentation.shtml
Pure-ftpd on FreeBSD 之攻略（中文简体版）
http://www.openldap.org/
jldap: http://www.openldap.org/jldap/ Novell 开发 LDAP Classes for Java
个人认为 Novell JLDAP 比 SUN JNDI (Java Naming and Directory Interface)好用。
Pure-ftpd 安裝說明 for RedHat 7.3 (RPM 安裝版)

12 声明

转载请保持此文档完整
主页地址：
http://www.kdeopen.com
http://home.9812.net/linux

OICQ:13721218
ICQ:101888222
AIM:xnetkiller
Yahoo:snetkiller
MSN:netkiller@msn.com

作者：Netkiller(陈景峰)

《Pure-FTPd + LDAP + MySQL + PGSQL + Virtual-Users + Quota How To》
2003 年 6 月 27 日星期五 第一版
2003 年 7 月 23 日星期三 第二版

如有问题 E-Mail: netkiller@9812.net