FTP 文件服务器项目需求(百度网盘)

功能概述：

编写服务器端，服务器端启动，然后启动客户端，通过客户端可以输入以下命令进行服务器上的文件查看：

1.cd 进入对应目录
2.ls 列出相应目录文件
3.puts 将本地文件上传至服务器
4.gets 文件名 下载服务器文件到本地
5.remove 删除服务器上文件
6.pwd 显示目前所在路径
7.其他命令不响应

项目启动方法

./ftpserver ../conf/server.conf
客户端
./client ip port

二期功能

1、密码验证

对于 linux 的登陆验证我想大家可能都知道是将输入的密码进行相应编码加密后与/etc/shadow 文件中对应的密码部分进行比较，如果相同则验证通过，如果不同则表明密码错误，但是问题是我们要如何将用户输入的密码加密然后进行比较。

为了提高安全性，Linux 引入了 salt，所谓 salt 即为一个随机数，引入的时候为一个 12bit 的数值，当用户设置密码时，会随机生成一个 salt 与用户的密码一起加密，得到一个加密的字符串(salt 以明文形式包含在该字符中），存储到密码文件中。crypt 函数可以将用户输入的密码和 salt 一起适应某种算法进行加密，该加密后的字符串就是我们需要的与密码文件中密码对比的加密字符串。
crypt 为支持不同的方式将 salt 格式化为
idsalt$encode
其中 id 代表不同的算法

1 | MD5
2a | Blowfish (not in mainline glibc; added in some
| Linux distributions)
5 | SHA-256 (since glibc 2.7)
6 | SHA-512 (since glibc 2.7) **当前我们 Linux 采用的加密算法**

这样我们就可以利用 crypt 函数将用户输入的字符加密，然后与密码文件中的密码进行对比了，有一个函数 getspnam 可以根据用户名从/etc/shadow 中得到密码，函数返回的数据结构

为

```
struct spwd {
    char *sp_namp;      /* Login name */
    char *sp_pwdp;      /* Encrypted password */
    long  sp_lstchg;    /* Date of last change (measured
                           in days since 1970-01-01 00:00:00 +0000 (UTC)) */
    long  sp_min;       /* Min # of days between changes */
    long  sp_max;       /* Max # of days between changes */
    long  sp_warn;      /* # of days before password expires
                           to warn user to change it */
    long  sp_inact;     /* # of days after password expires
                           until account is disabled */
    long  sp_expire;    /* Date when account expires (measured
                           in days since 1970-01-01 00:00:00 +0000 (UTC)) */
    unsigned long sp_flag;  /* Reserved */
};
```

我们现在创建一个 test 用户，并将密码设置为 1234 来做个测试看一下
这是创建之后从/etc/shadow 中取出来的密码部分
6qOrvsN41$gGlv8z7P1sAKuyaTAY03AvCn9/Z6Ygc4DJ9Uwe0RVzNAI6TQsTfsdihPnIh3lSZV2C02HjW.9bvJNdep3k.ER.
可以看到这里的 salt 为6qOrvsN41

最后写个程序做下验证

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <shadow.h>
#include <errno.h>

void help(void)
{
    printf("用户密码验证程序\n 第一个参数为用户名!\n");
    exit(-1);
}

void error_quit(char *msg)
{
    perror(msg);
    exit(-2);
}
```

```
void get_salt(char *salt,char *passwd)
{
	int i,j;

	//取出 salt,i 记录密码字符下标,j 记录$出现次数
	for(i=0,j=0;passwd[i] && j != 3;++i)
	{
		if(passwd[i] == '$')
			++j;
	}

	strncpy(salt,passwd,i-1);
}

int main(int argc,char **argv)
{
	struct spwd *sp;
	char *passwd;
	char salt[512]={0};

	if(argc != 2)
		help();

	//输入用户名密码
	passwd=getpass("请输入密码:");
	//得到用户名以及密码,命令行参数的第一个参数为用户名
	if((sp=getspnam(argv[1])) == NULL)
		error_quit("获取用户名和密码");
	//得到 salt,用得到的密码作参数
	get_salt(salt,sp->sp_pwdp);

	//进行密码验证
	if(strcmp(sp->sp_pwdp,crypt(passwd,salt)) == 0)
		printf("验证通过!\n");
	else
		printf("验证失败!\n");

	return 0;
}
```

注意：gcc 编译该程序时需要加入 -lcrypt

注：进行登录密码验证后，客户端可以删除的文件及自己有权限删除的，而非 root 的。

通过 seteuid 进行业务进程的权限设置操作（适配的是多进程）。
针对多线程，自行存储用户 ID，通过检查文件权限，来自行判断是否有查看、删除、新增功能。

2.日志记录

1、日志记录客户端请求信息，及客户端连接时间
2、日志记录客户端操作记录，及操作时间

3.断点续传

1、进行 gets hello.avi 时候，判断本地是否存在 hello.avi，如果存在，stat 获取 hello.avi 的大小，传递给服务器的是 gets hello.avi 1000

4.mmap 将大文件映射入内存，进行网络传递

1、当你发现文件大于 100M，就进行 mmap 该文件，然后再传输

三期功能：

1、数据库连接，通过数据库存储用户名，salt 值(采用随机字符串生成)，密码(密文形式存储)，通过数据库存储日志
如何生成随机字符串，这里有一个帖子供参考
http://liuzhigong.blog.163.com/blog/static/178272375201121781915 23/
2、密码验证的方式是服务器先传输 salt 值给客户端，客户端 crypt 加密后，传输密码密文给服务器，服务器匹配后，判断是否成功
3、由数据库记录每一个用户的路径情况，每一个用户的文件及目录采用数据库的表进行存储，每个文件的文件内容以其 MD5 码存在磁盘上，并以 md5 码进行命名，所有用户的文件都存在一个目录里，各自用户只能看到自己的文件，不能看到其他人的文件，通过 C 接口得到每一个文件的 MD5，查看 http://bbs.chinaunix.net/thread-2125735-1-1.html，或者直接 man MD5

四期功能：

多点下载，多点下载是指客户端通过多个 sfd 连接不同的服务器，将一份大文件拆分为几段，不同段从不同的服务器进行下载
分为三级难度：
A:固定从某几台服务器下载，而且文件多点下载一定一次性下载完毕
B:从某台 IP 服务器上得到具有数据源的服务器地址，然后再从对应的地址进行下载，而且文件一次性下载完毕
C:如果多点下载过程中，客户端断开，那么下次重新下载时，需要再次进行多点下载，所以客户端本地需要使用文件，或者数据库存储多点下载，每个位置下载到什么地方

想研究实际 FTP 协议，详见
http://blog.csdn.net/yxyhack/article/details/1826256