

同济大学计算机系

计算机网络课程实验报告



学 号 1552239

姓 名 岳昊玮

专 业 计算机科学与技术

授课老师 沈坚

§ 0. 开篇

1、按给出的文档安装好 RHEL7.4 的基本系统

- 虚拟机软件 VMware 可以选用其它版本，也可以使用 VirtualBOX
- 无论使用何种虚拟机软件，RHEL7.4 的安装过程及选项按文档要求，不准更改!!!
- 在安装配置过程中，使用 yum 时，不允许从网络上下载软件包，所有需要的资源 iso 中均有

2、修改配置文件，使每次启动后网卡生效

系统装好之后，FlashFXP 和 SecureCRT 均无法连接，在虚拟机中输入 ifconfig 命令查看，（ifconfig 是 linux 中用于显示或配置网络设备(网络接口卡)的命令，英文全称是 network interfaces configuring。）不带任何参数即表示显示当前系统网络配置情况。发现我们在装系统时设置的 IP 和子网掩码并没有显示。后来才知道是因为默认开机网卡不被激活的。

```
[root@Raindrops ~]# ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 00:0c:29:1d:86:1b txqueuelen 1000 (Ethernet)
    RX packets 12 bytes 934 (934.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

首先输入一下命令，编辑模式进入 ifcfg-ens32 文件，

```
[root@Raindrops ~]# vi /etc/sysconfig/network-scripts/ifcfg-ens32
```

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV5_FAILURE_FATAL=yes
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens32
UUID=6bda65d2-ac47-478a-af45-5fd4f2dc21b8
DEVICE=ens32
ONBOOT=yes
IPADDR=192.168.70.230
PREFIX=24
IPV6_PRIVACY=no
:wq
```

将 onboot=yes，设置网卡为开机启动，然后按“Esc”退出编辑模式，输入:wq 保存并退出，然后重启虚拟机即可。开机后，输入以下命令，可以看到装系统时设置 IP 地址，说明网卡激活成功。

```

[root@Raindrops ~]# ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.70.230 netmask 255.255.255.0 broadcast 192.168.70.255
    inet6 fe80::3c61:a9ac:3f29:7114 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:1d:86:1b txqueuelen 1000 (Ethernet)
    RX packets 356 bytes 25210 (24.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 42 bytes 6539 (6.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

- 3、设置宿主机的 IP 地址与虚拟机在同一网段，使主机和虚拟机能双向 ping 通。

宿主机 ping 虚拟机

```

C:\Users\15801>ping 192.168.70.230

正在 Ping 192.168.70.230 具有 32 字节的数据:
来自 192.168.70.230 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.70.230 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.70.230 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.70.230 的回复: 字节=32 时间<1ms TTL=64

192.168.70.230 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 0ms, 最长 = 0ms, 平均 = 0ms

```

虚拟机 ping 主机

```

[root@Raindrops ~]# ping 192.168.70.1
PING 192.168.70.1 (192.168.70.1) 56(84) bytes of data.
64 bytes from 192.168.70.1: icmp_seq=1 ttl=128 time=0.609 ms
64 bytes from 192.168.70.1: icmp_seq=2 ttl=128 time=0.289 ms
64 bytes from 192.168.70.1: icmp_seq=3 ttl=128 time=0.325 ms
64 bytes from 192.168.70.1: icmp_seq=4 ttl=128 time=0.325 ms
^C
--- 192.168.70.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.289/0.387/0.609/0.129 ms

```

- 4、在 ping 通的基础上，安装 SecureCRT 并使能登录 Linux（root 用户登录后出现#）

发现 SecureCRT 可以支持多用户，（非 root 用户登录的话命令提示符是\$，只有 root 用户提示符是#）

```

192.168.70.230 192.168.70.230 (1) x 192.168.70.230 (2)
Last login: Thu Sep 21 03:34:36 2017 from 192.168.70.1
[root@Raindrops ~]#
[root@Raindrops ~]#
[root@Raindrops ~]#
[root@Raindrops ~]#

```

- 5、修改 Linux 配置文件，使中文显示正常(不允许通过修改 SecureCRT 中字符编码方式实现)

在虚拟机中修改/etc/sysconfig/i18n 文件，设置 LANG="zh_CN.gbk"。保存退出，输入 locale，查看支持的编码

```
LANG="zh_CN.gbk"
SYSFONT="latarcyrheb-sun16"
SUPPORTED="zh_CN.UTF-8.zh_CN:zh"
~
~

[root@Raindrops home]# locale
LANG=zh_CN.gbk
LC_CTYPE="zh_CN.gbk"
LC_NUMERIC="zh_CN.gbk"
LC_TIME="zh_CN.gbk"
LC_COLLATE="zh_CN.gbk"
LC_MONETARY="zh_CN.gbk"
LC_MESSAGES="zh_CN.gbk"
LC_PAPER="zh_CN.gbk"
LC_NAME="zh_CN.gbk"
LC_ADDRESS="zh_CN.gbk"
LC_TELEPHONE="zh_CN.gbk"
LC_MEASUREMENT="zh_CN.gbk"
LC_IDENTIFICATION="zh_CN.gbk"
LC_ALL=
```

这道题老师挖了一个不小的坑，重点在中文文件名的显示上面。如果结合第 10 题的话这道题或许会简单很多，关键在于使用 FlashFXP 向虚拟机传输文件时文件的编码格式的选择一定要和虚拟机设置的语言相一致，否则就会像下图一样，只有文件名无法显示中文

```
[root@Raindrops home]# ls -l
总用量 3304
-rw-r--r--. 1 root root 3383296 5月 15 21:50 棕?.esnw
```

这时要回到 FlashFXP 的文件字符编码的设置为 GBK，将具体操作见第 10 题。设置完之后，再回来重新利用 FlashFXP 上载中文文件到虚拟机，（这里要注意的是，修改完之后要重新进行连接才会生效，在这之前一定要删除原来按照先前字符编码上传的文件，否则这些文件将无法进行删除，修改重启之后，按照之前字符编码上传的文件在 FlashFXP 中也会显示成乱码，这也就是第 10 题的原理吧）

G:\桌面便笺			
名称	大小	修改时间	
上级目录			
飞.esnw	3.23 MB	2017/5/15 21:50:23	

名称	大小	修改时间	属性
上级目录			
棕?.esnw	3.23 MB	2017/5/15 21:50:22	-rw-r--r--

好了，下面修改上传文件的字符编码为 gbk 之后，重新上传了一个文件，再次查看

```
[root@Raindrops home]# ls -l
总用量 3304
-rw-r--r--. 1 root root 3383296 5月 15 21:50 飞.esnw
[root@Raindrops home]#
```

6、使数据库服务器（RHEL7.4 中使用了 MariaDB 数据库，与 MySQL 保持兼容）随开机启动，配置 MySQL 的 root 用户密码

输入 `ntsysv` 命令，然后可以查看并设置 linux 的所有开机启动项

```
[root@Raindrops home]# ntsysv
```



通过上下键移动到 `mariadb.service` 空格键设置开机启动，然后确定保存，reboot,其实也可以 `systemctl start mariadb` 的。



奥对，忘了配置数据库的密码：

顺便我还在/etc/my.cnf.d/server.cnf 中加了如下两句

```
[server]
character_set_server=gbk
# this is only for the mysqld standalone daemon
[mysqld]
character_set_server=gbk
# this is only for embedded server
[embedded]
```

在/etc/my.cnf.d/client.cnf 中

```
[client]
default-character-set=gbk
# This group is not read by mysql client library,
# If you use the same .cnf file for MySQL and MariaDB,
# use it for MariaDB-only client options
[client-mariadb]
```

下面是结果

```
MariaDB [(none)]> source /home/demo.sql;
Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

如果字符集设置不正确，当sql语句中遇到中文将无法正确执行

```
ERROR 1067 (42000) at line 7 in file: '/home/demo.sql': Invalid default value for 'sex'
ERROR 1146 (42S02) at line 15 in file: '/home/demo.sql': Table 'demo.student' doesn't exist
ERROR 1146 (42S02) at line 16 in file: '/home/demo.sql': Table 'demo.student' doesn't exist
ERROR 1146 (42S02) at line 17 in file: '/home/demo.sql': Table 'demo.student' doesn't exist
ERROR 1146 (42S02) at line 18 in file: '/home/demo.sql': Table 'demo.student' doesn't exist
MariaDB [(demo)]\ exit
```

(注意设置完之后一定要 reboot，只是 systemctl restart/start mariadb 是不会更新配置的)

```
MariaDB [demo]> show variables like 'character%';
```

variable_name	value
character_set_client	gbk
character_set_connection	gbk
character_set_database	gbk
character_set_filesystem	binary
character_set_results	gbk
character_set_server	gbk
character_set_system	utf8
character_sets_dir	/usr/share/mysql/charsets/

设置完成后的字符集编码

```
8 rows in set (0.01 sec)
```



```

Last login: Mon Sep 25 01:56:40 2017
[root@Anokoro ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> source /home/demo.sql; 导入数据库文件
Query OK, 0 rows affected (0.03 sec)

Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

MariaDB [demo]> select * from student 命令输完了一定要加分号!
-> ;
+-----+-----+-----+-----+-----+
| sno   | sname | ssex | sage | sdept |
+-----+-----+-----+-----+-----+
| 200215121 | 李勇  | 男   | 20   | CS    |
| 200215122 | 刘晨  | 女   | 19   | CS    |
| 200215123 | 王敏  | 女   | 18   | MA    |
| 200215125 | 张立  | 男   | 19   | IS    |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MariaDB [demo]> █

```

8、 安装 PHP 服务，使 Apache 服务随开机启动，配置 PHP 与 Apache 服务，建立 demo.php 文件（将附件的 demo.php 直接复制过来即可），使在浏览器中输入 http://192.168.80.230(虚拟机地址，若不同则对应修改)时，能执行 demo.php 并在浏览器中显示 student 表的 4 条记录。

【提示:】要在 php/httpd 的配置中使 demo.php 成为缺省启动页面

首先配置 yum 源，Yum 源的配置具体步骤如下：

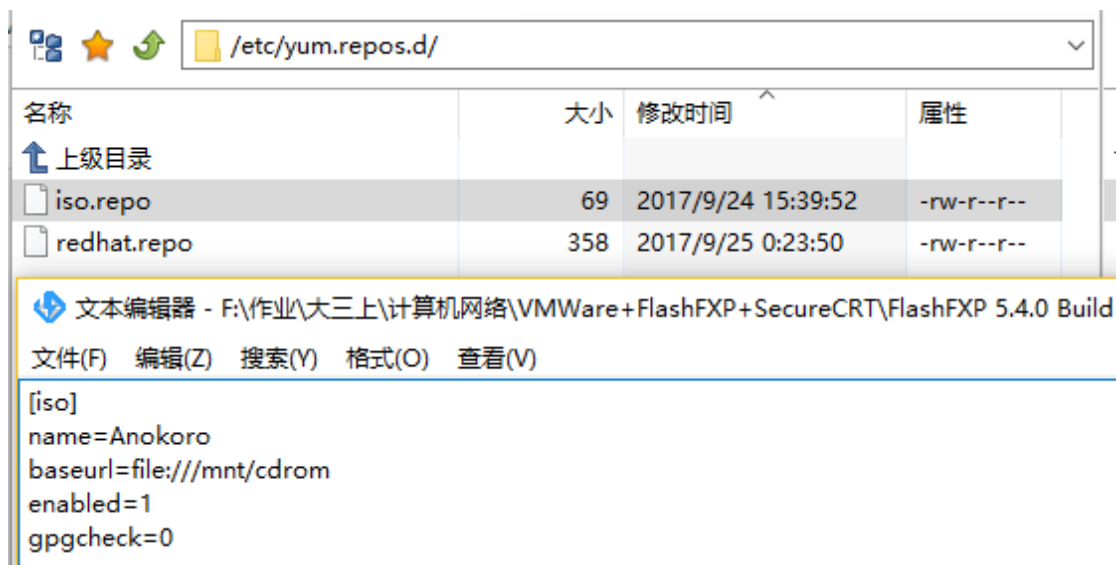
1. 新建/mnt/cdrom 目录 :mkdir /mnt/cdrom, 镜像文件通过 FlashFXP 直接上传到/mnt/cdrom。
2. 挂载光盘: “mount -o loop /dev/sr0 /mnt/cdrom”，然后进入挂载目录，可以查看光盘的内容。参数“-o loop”是挂载特殊设备的。
3. 新建 yum 源配置文件: vim /etc/yum.repos.d/dvd.repo .然后输入内容。注意: baseurl 后面跟的是挂载光盘所在的目录, 也就是 yum 的仓库。当然也可以使用其他的 yum 源, 比如 ftp, http, [iso]

name=Anokoro

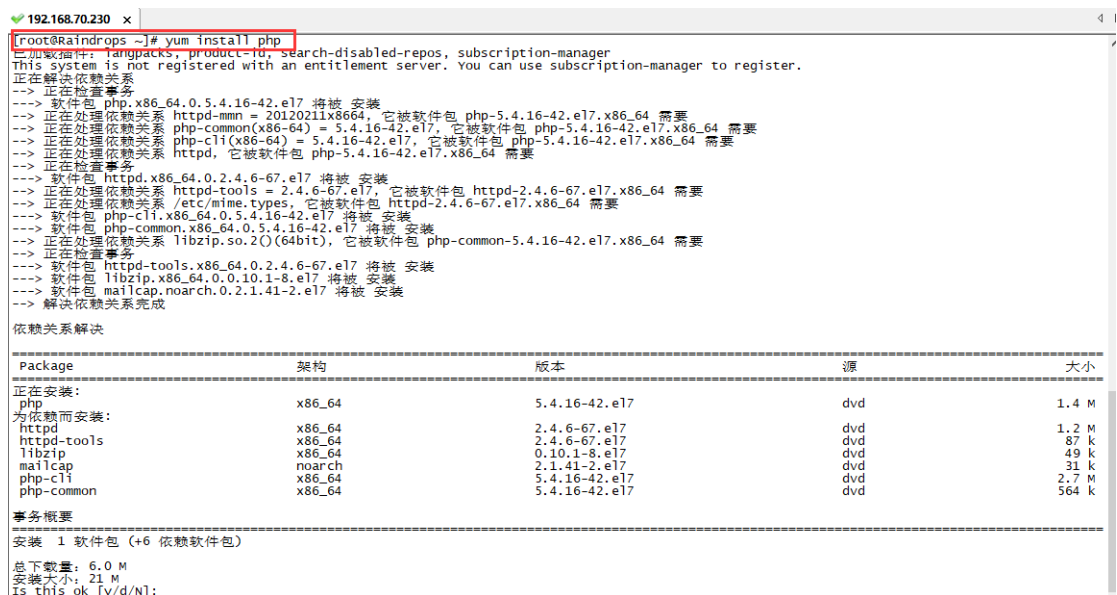
baseurl=file:///mnt/cdrom

enabled=1

gpgcheck=0



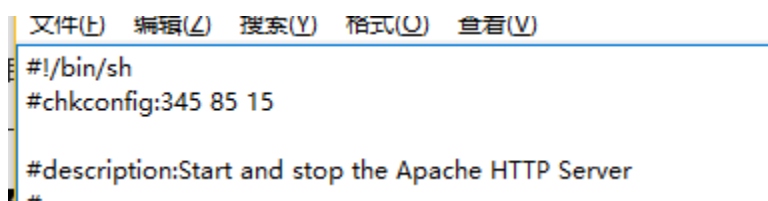
使用配置的 yum 源安装 php 服务，yum 源的仓库是安装系统时使用的 ISO。



增添 apache 为系统服务并设置开机自启动

```
[root@Anokoro ~]# find / -name apachectl
/usr/sbin/apachectl
[root@Anokoro ~]# cp /usr/sbin/apachectl /etc/init.d/httpd
[root@Anokoro ~]# chkconfig --add httpd
服务 httpd 不支持 chkconfig
[root@Anokoro ~]#
```

这里说的是 httpd 服务不支持 chkconfig，添加支持：vi /etc/init.d/httpd 在 #!/bin/sh 下添加这两句：



说明: chkconfig 后面是服务识别参数, 在级别 3、4、5 中启动, 启动和关闭顺序

```
[root@Anokoro ~]# chkconfig --add httpd
服务 httpd 不支持 chkconfig
[root@Anokoro ~]# chkconfig --add httpd
[root@Anokoro ~]# chkconfig --list httpd
```

注: 该输出结果只显示 SysV 服务, 并不包含原生 systemd 服务。SysV 配置数据可能被原生 systemd 配置覆盖。

添加对chkconfig的支持
后重新添加httpd服务

要列出 systemd 服务, 请执行 'systemctl list-unit-files'。
查看在具体 target 启用的服务请执行
'systemctl list-dependencies [target]'。

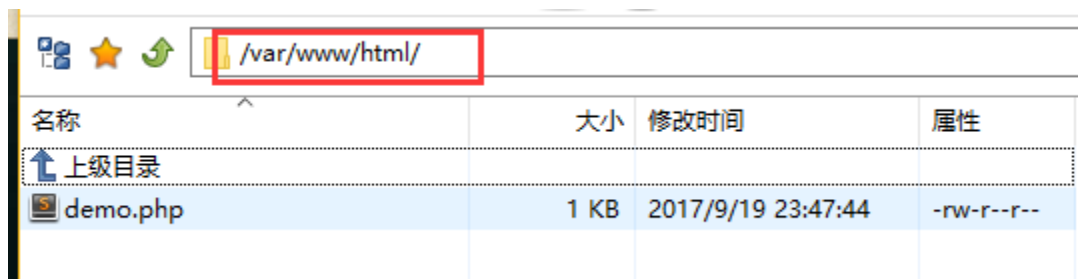
```
httpd          0:关   1:关   2:关   3:开   4:开   5:开   6:关
[root@Anokoro ~]#
```

将 demo.php 放在 Apache 的 Web 站点目录/var/www/html (默认)

(设置网站根目录, 即网页文档在系统中的存放路径)

```
#
DocumentRoot "/var/www/html"
```

这从配置文件/etc/httpd/conf/httpd.conf 中也可看出



设置网站的默认首页, 可以设置多个首页文件, 用空格隔开。默认字符集为 gbk

```
<IfModule dir_module>
  DirectoryIndex index.html demo.php
</IfModule>

#
AddDefaultCharset gbk

AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/x-httpd-php-source .phps
AddType application/x-httpd-php .php
#
```

刚开始一直连接不上, 后来才知道要关闭 linux 的防火墙, 或者打开 80 端口 (因为处于安全考虑到防火墙不能一直关闭, 所以我还是选择了临时关闭的方式, 但是是因为是自己电脑嘛, 所以, 嘿嘿, 还是设置为开机关闭好了)

```
[root@Anokoro ~]# systemctl stop firewalld
[root@Anokoro ~]#
```

开始运行成功之后, 打开浏览器, 发现一片空白, 我还踌躇了好久, 以为是我 demo.php 文件放错了位置, 排错许久后想起看看 error_log, 这下才恍然大悟, 原来是 mysqli 语句的问题, 也就是说所有的该配置的都配置好了, 文件位置也对了。log 的内容如下

```
[Wed Sep 27 02:41:49.545499 2017] [error] [pid 2059] [client 192.168.70.1:13622] PHP Fatal error: Class 'mysqli' not found in /var/www/html/demo.php on line 4
[Wed Sep 27 02:41:50.424873 2017] [error] [pid 2060] [client 192.168.70.1:13623] PHP Fatal error: Class 'mysqli' not found in /var/www/html/demo.php on line 4
[Wed Sep 27 02:41:57.441645 2017] [error] [pid 7135] [client 192.168.70.1:13631] PHP Fatal error: Class 'mysqli' not found in /var/www/html/demo.php on line 4
[Wed Sep 27 02:47:42.097069 2017] [error] [pid 2061] [client 192.168.70.1:13692] PHP Fatal error: Class 'mysqli' not found in /var/www/html/demo.php on line 4
[Wed Sep 27 02:47:50.658004 2017] [error] [pid 2062] [client 192.168.70.1:13693] PHP Fatal error: Class 'mysqli' not found in /var/www/html/demo.php on line 4
```

还有一个查错方法，当 httpd 服务无法开启时，由于配置文件 httpd.conf 的配置错误而产生的错误（因为之前在修改配置的时候不太确定，可能会改了很多不相关的东西而导致错误）可以通过 `systemctl status httpd` 来查看，如下

```
[root@Anokoro ~]# systemctl status httpd
* httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since 三 2017-09-27 03:25:55 CST; 3s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Process: 2464 ExecStop=/bin/kill -WINCH ${MAINPID} (code=exited, status=1/FAILURE)
  Process: 2463 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)
 Main PID: 2463 (code=exited, status=1/FAILURE)

9月 27 03:25:55 Anokoro systemd[1]: Starting The Apache HTTP Server...
9月 27 03:25:55 Anokoro httpd[2463]: AH00526: Syntax error on line 128 of /etc/httpd/conf/httpd.conf:
9月 27 03:25:55 Anokoro httpd[2463]: allow and deny must be followed by 'from'
9月 27 03:25:55 Anokoro systemd[1]: httpd.service: main process exited, code=exited, status=1/FAILURE
9月 27 03:25:55 Anokoro kill[2464]: kill: cannot find process ""
9月 27 03:25:55 Anokoro systemd[1]: httpd.service: control process exited, code=exited status=1
9月 27 03:25:55 Anokoro systemd[1]: Failed to start The Apache HTTP Server.
9月 27 03:25:55 Anokoro systemd[1]: Unit httpd.service entered failed state.
9月 27 03:25:55 Anokoro systemd[1]: httpd.service failed.
[root@Anokoro ~]# systemctl start httpd
[root@Anokoro ~]#
```

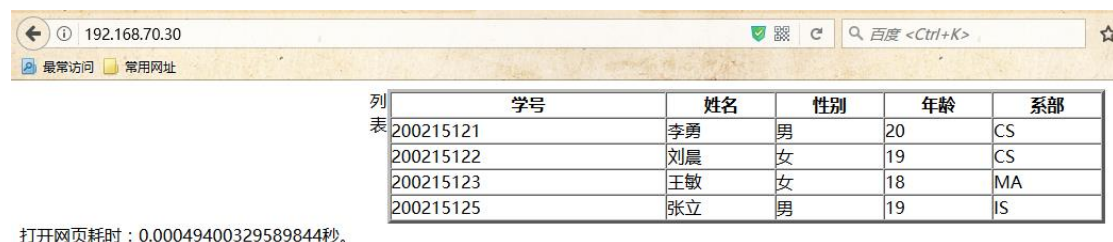
成功连接

```
[root@Anokoro ~]# netstat -anpt | grep httpd
tcp6      0      0 :::80          :::*           LISTEN     2480/httpd
```

好了，下面进入下一个坑，数据库的问题，根据查看 `error_log` 得出的 php 对 `mysqli` 的不支持，根据以前在 windows 下面解决这个问题的操作：取消 `php.ini` 中的 `extension=php_mysqli.dll` 的注释即可，我以为 linux 下也是如此方便结果一打开 `php.ini` 就傻眼了，完全没有分行，通过 `Ctrl+F` 也找不到，那只能说明 linux 不是这种操作了，所以就另寻他路。

```
[root@Anokoro ~]# yum install php-mysqli
```

刷新刚才什么都没有的页面，我的天，终于出来了!!!



打开网页耗时：0.00049400329589844秒。

学号	姓名	性别	年龄	系部
200215121	李勇	男	20	CS
200215122	刘晨	女	19	CS
200215123	王敏	女	18	MA
200215125	张立	男	19	IS

9、安装缺少的 MySQL 组件，使 C/C++ 能编译 MySQL 测试程序(测试程序为附件的 `mysql_demo.cpp`)并通过，执行后显示 `student` 表的 4 条记录

【提示:】编译 MySQL 需要加编译选项，可以直接命令行输入，也可以写 `makefile` 文件
编译生成可执行文件，运行后显示 `student` 表的四条记录，且中文显示正常
(这道题：真·错误一个接一个)

因为缺少头文件 `<mysql>` 等等，首先安装 `mariadb-devel`，提供编译环境，

```
[root@Anokoro ~]# yum install mariadb-devel
已加载插件: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered with an entitlement server. You can use subscription-manager to register.
file:///mnt/cdrom/repodata/repomd.xml: [Errno 14] curl#37 - "couldn't open file /mnt/cdrom/repodata/repomd.xml"
正在尝试其它镜像。
软件包 1:mariadb-devel-5.5.56-2.el7.x86_64 已安装并且是最新版本
无须任何处理
[root@Anokoro ~]#
```

```
[root@Anokoro ~]# rpm -qa | grep maria
mariadb-5.5.56-2.el7.x86_64
mariadb-libs-5.5.56-2.el7.x86_64
mariadb-server-5.5.56-2.el7.x86_64
mariadb-devel-5.5.56-2.el7.x86_64
```

安装完之后，gcc 编译在自己的库中并找不知道，就相当于没有安装，在编译的时候，加上下面的语句就可以了，这样 gcc 编译时就会去对应的目录下面找需要的头文件（不过我为了方便直接把 mysql 的 include 目录下的所有头文件拷贝到了 gcc 的 include 目录下，这样每次编译的时候就不需要加一大长串的参数，gcc 在自己的 include 下就能找到自己所需要的所有文件）如下所示目录

/usr/include/mysql/		/usr/lib/gcc/x86_64-redhat-linux/4.8.2/include/		
你	大小		大小	修改时间
上级目录				
private	8 KB			

```
[root@Anokoro ~]# gcc -o main /home/mysql_demo.cpp -lmysqlpp -I/usr/include/mysql -I/usr/local/include/
```

解决了一堆问题之后，运行

```
[root@Anokoro ~]# gcc /home/mysql_demo.cpp
/tmp/ccrQvEre.o: 在函数'main'中:
mysql_demo.cpp:(.text+0x18): 对'mysql_init'未定义的引用
mysql_demo.cpp:(.text+0x32): 对'std::cout'未定义的引用
mysql_demo.cpp:(.text+0x37): 对'std::basic_ostream<char, std::char_traits<char, std::char_traits<char> >&, char const*>'未定义的引用
mysql_demo.cpp:(.text+0x3c): 对'std::basic_ostream<char, std::char_traits<char, std::char_traits<char> >&)'未定义的引用
mysql_demo.cpp:(.text+0x44): 对'std::ostream::operator<<(std::ostream& (&mysql_demo.cpp:(.text+0x86): 对'mysql_real_connect'未定义的引用
```

出了好长一个错误列表，其中 Mysql_init 等等和数据库有关的操作出问题还能理解，可能是数据库相关还没搞好嘛，但是 cout 等等只和 C++ 有关的编译不通过就不对了，查了一下才发现，原来 gcc 是 c 语言的编译器，而我们写的代码是 C++ 代码，所以改为输入 g++。

```
[root@Anokoro ~]# g++ /home/mysql_demo.cpp
/tmp/cc6bYPA1.o: 在函数'main'中:
mysql_demo.cpp:(.text+0x18): 对'mysql_init'未定义的引用
mysql_demo.cpp:(.text+0x86): 对'mysql_real_connect'未定义的引用
mysql_demo.cpp:(.text+0x9c): 对'mysql_error'未定义的引用
mysql_demo.cpp:(.text+0xee): 对'mysql_set_character_set'未定义的引用
mysql_demo.cpp:(.text+0xff): 对'mysql_query'未定义的引用
mysql_demo.cpp:(.text+0x114): 对'mysql_error'未定义的引用
mysql_demo.cpp:(.text+0x161): 对'mysql_store_result'未定义的引用
mysql_demo.cpp:(.text+0x1a3): 对'mysql_num_rows'未定义的引用
mysql_demo.cpp:(.text+0x33b): 对'mysql_fetch_row'未定义的引用
mysql_demo.cpp:(.text+0x35b): 对'mysql_free_result'未定义的引用
mysql_demo.cpp:(.text+0x367): 对'mysql_close'未定义的引用
collect2: 错误: ld 返回 1
[root@Anokoro ~]#
```

好了，剩下的都是和数据库有关的操作函数有问题了。

在后面加入 -L/usr/lib64/mysql -lmysqlclient 指令，运行，成功!! 其实 ./a.out 也行，我只是 find 了一下，发现是默认在那个目录下面的，因为 linux 下生成的可执行文件都无脑命名为 a，格式为 out，不像 windows 下的是 exe，但是可以在编译阶段设置其名字，另外我就把 libmysqlclient.so 拷贝到可被系统搜索的目录下 /usr/lib 目录下面，这样就不用加它的路径了，如下

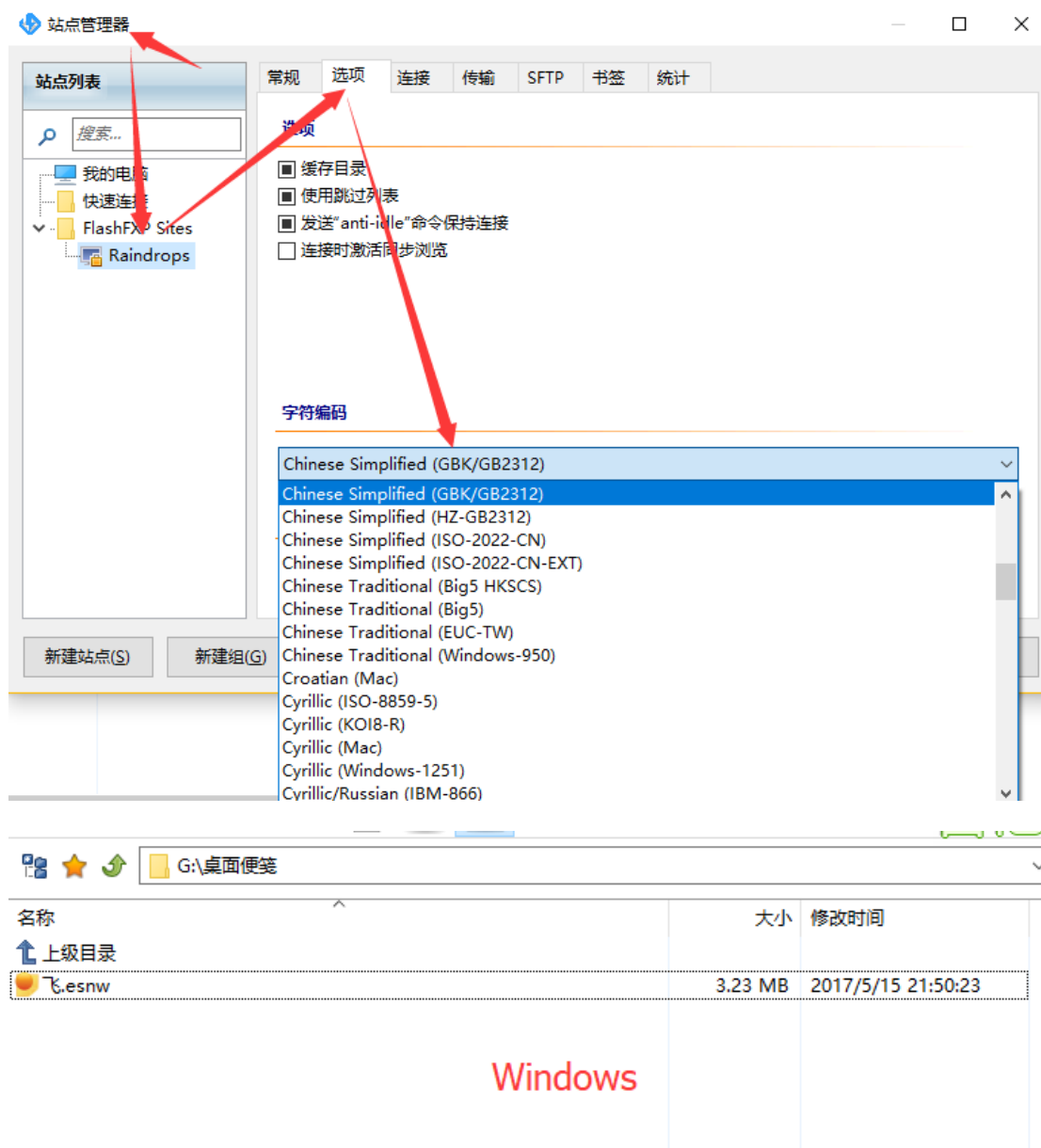
```

[root@Anokoro ~]# g++ mysql_demo.cpp -o mysql_demo -lmysqlclient
[root@Anokoro ~]# ./a.out
-bash: ./a.out: 没有那个文件或目录
[root@Anokoro ~]# ./mysql_demo
select return 4 records
学号: 200215121  姓名: 李勇  性别: 男  年龄: 20  系部: CS
学号: 200215122  姓名: 刘晨  性别: 女  年龄: 19  系部: CS
学号: 200215123  姓名: 王敏  性别: 女  年龄: 18  系部: MA
学号: 200215125  姓名: 张立  性别: 男  年龄: 19  系部: IS

```

10、具体的原理上面在第 5 题已经介绍过了，这里只说下具体的操作步骤:

在 FlashFXP 的站点管理器中修改上传文件的字符编码为 GBK，应用之后，断开重连即可更新配置，然后上传中文文件名的文件。



<div> <div> <div></div> <div>☆</div> <div>↶</div> </div> <div>/home/</div> <div>▼</div> </div>			
名称	大小	修改时间	属性
<div>↑ 上级目录</div>			
<div>🔴 飞.esnw</div>	3.23 MB	2017/5/15 21:50:22	-rw-...

Linux目录

```

[root@Raindrops home]# ls -l
总用量 3304
-rw-r--r--. 1 root root 3383296 5月 15 21:50 飞.esnw
[root@Raindrops home]# vi /etc/sysconfig/i18n

```

SecureCRT