

计算机网络与应用 实验二 实验报告

姓名：李宇飏

学号：2020011645

班级：自 05

实验目的

学习网络服务的配置和应用、加深对各类网络服务和网络机制的理解，锻炼配置网络能力。

实验内容

FTP 服务

实验配置：两台计算机（Windows 7）通过 USB 网卡直连，一台 IP 设置为 192.168.1.1/24，另一台配置为 192.168.1.2/24，默认网关、DNS 服务器指向 Server。

使用 Serv-U 在 Server 端搭建 FTP 服务器，配置不同权限的用户，并在 Client 端使用 FTP 软件测试该服务。基本步骤如下：

1. Server 端启动 Serv-U，新建域，配置其绑定 IP、端口（默认 21）、域名、本地目录及其他权限设置；
2. Server 端为该域新建 2 个用户，分别赋予目录端读写和只读权限；
3. Client 端启动任意 FTP 软件，连接 \\192.168.1.1:21，并分别使用两个账号登录，测试文件的上传、下载。当使用只读权限账号上传文件时，上传被拒绝，提示 **permission denied**；其他情况下的上传、下载均成功。

HTTP 服务

使用 Flask 在本地提供 HTTP 服务。基本步骤如下：

1. 配置环境，部署 Python 3.8，带有 Flask、SQLite 等依赖；
2. 编写 `template/template.py`，在需要显示具体数据数据的区域设置绑定逻辑；
3. 编写 `app.py` 中的数据库交互逻辑与渲染逻辑：
 - 新用户登录后，先从数据库中取出上一次访问的数据；若返回 `None`，则上一次访问 IP、时间字符串设置为“上一次访问”，否则从返回的 `Visitor`（好像叫这个名字）实例中获得上次访问的 IP 与时间戳，后者通过提供的 API 转换为 IEEE8601 时间字符串；
 - 在数据库中插入本次登录数据：新建 `Visitor` 实例，填写此次访问的来源 IP 与时间戳，调用提供的接口写入数据库；
 - 调用提供的接口获得访客总数（包含此次访问）；
 - 根据访客总数奇偶，设置图片 URL（奇偶分别显示不同图片）
 - 将所需数据按照绑定逻辑传入 `render_template(...)` 方法。
4. 启动服务器：在站点文件目录下运行 `[PATH_TO_PYTHON_EXECUTABLE] -m flask run`，其中 `[PATH_TO_PYTHON_EXECUTABLE]` 替换为 `python.exe` 路径。
5. 访问 `http://127.0.0.1:5000` 查看网页效果，刷新可观察到访客数量增加、上次访问信息更新、图片切换。

HTTP 服务部署

使用 Apache 在本地部署 HTTP 服务器。基本步骤如下：

1. 修改 `httpd.conf`，配置 `SRVROOT` 指向 `htdocs` 所在目录；
2. 运行 `httpd.exe`，启动 Apache，登陆 `http://localhost` 即可看到 `htdocs/index.html`，显示 `"It works!"`；
3. 关闭 Apache，配置路由 alias： `/static/ -> xxx/server/static/`，并配置 `xxx/server` 的访问权限：

```
Alias /static/ "xxx/server/static/"
<Directory "xxx/server">
    AllowOverride None
    Options None
    Require all granted
</Directory>
```

4. 启动 Apache, 可通过 **http://localhost/static/** 访问 **xxx/server/static/** 下的静态文件;
5. 关闭 Apache, 继续配置 WSGI: 在 httpd.conf 中配置 WSGIScriptAlias 等变量, 其中 WSGIScriptAlias 需要配置在 DocumentRoot 之后, 以覆盖对 / 的路由;
6. 调整上一任务中的网页代码, 将图片的 URL 前缀改为 http://localhost/static/, 配置从 HTTP 服务获取静态文件;
7. 启动 Apache, 可通过 **http://localhost** 看到上一任务中配置的网页; 其中, 图片文件也从 HTTP 服务获取。

DNS

实验配置: 两台计算机通过 USB 网卡直连, 一台 (Windows Server 2003) IP 设置为 192.168.1.1/24, 充当 Server; 另一台 (Windows 7) 配置为 192.168.1.2/24, 默认网关、DNS 服务器指向 Server, 充当 Client。

使用 Windows Server 提供 DNS 服务, 基本步骤如下:

1. 在 Windows Server 上启动 DNS 服务器, 新建区域 test.local, 添加 A 记录与对应的反向搜索 PTR:
 - server.test.local -> 192.168.1.1, 1.1.168.192.in-addr.arpa -> server.test.local
 - client.test.local -> 192.168.1.2, 2.1.168.192.in-addr.arpa -> client.test.local

2. 启动 DNS 服务器，在 Client 端查询：

- **nslookup server.test.local**，查询到 A 记录为 **192.168.1.1**，响应服务器为 **server.test.local**；
- **nslookup -qt=ptr 192.168.1.1**，查询到对应记录为 **server.test.local**，响应服务器为 **server.test.local**；

3. Client 端修改上一任务中第 6 步的图片 URL，前缀的 **localhost** 改为 **client.test.local**；

4. Client 启动上一任务中的 Apache，并访问 **http://client.test.local**，可以查看上一任务中的网页。

DHCP 服务

实验配置：两台计算机通过 USB 网卡直连，一台（Windows Server 2003）IP 设置为 192.168.1.1/24，充当 Server；另一台（Windows 7）配置使用自动获取的 IP、DNS 服务器，充当 Client。

使用 Windows Server 提供 DHCP 服务，基本步骤如下：

1. 在 Server 上启动 DHCP 服务，新建作用域 **test**，设置：

- 地址范围 192.168.1.1 - 192.168.1.254
- 子网掩码 24 (255.255.255.0)
- 租期期限 8 天
- 排除地址 192.168.1.1 - 192.168.1.5
- 默认 DNS 服务器、路由器为 192.168.1.1

2. 可在 **Server** 的统计数据中查看到可用地址池中有 **249** 个；

3. 在 Client 上获取 IP：

- **ipconfig /all** 可以查看到 USB 网卡没有得到 IP 地址等；
- **ipconfig /renew**，网卡得到地址 **192.168.1.6**，同时查看到下发的 DNS 服务器、路由器为 **192.168.1.1**；
- **/ipconfig /release, ipconfig /renew**，网卡得到地址 **192.168.1.6**，这是因为 DHCP 服务器端根据 MAC 地址，默认下发上次使用的 IP（若可用）；可以配置总是下发新 IP，这时获取到 **192.168.1.7**；

4. 在 Server 上配置 Client 该网卡的 MAC 地址与 192.168.1.100 绑定（保留），此时 **Client** 端重新获取 IP，得到 **192.168.1.100**。

实验思考

HTTP 服务

Apache 服务器的配置文件

可以在 Apache 中修改其他配置：

- ServerName [SERVER_NAME] 配置站点名称（域名/IP）
- Listen [PORT] 配置监听端口
- DirectoryIndex [INDEX_1], [INDEX_2], ... 配置（多个不同优先级的）默认首页文件名
- Alias [VIRTUAL_PATH] [ACTUAL_PATH] 配置 VIRTUAL_PATH -> ACTUAL_PATH 的虚拟目录，一般需要配合 <Directory> 标签指定权限；
-

Apache v.s. Nginx

- Nginx 相比 Apache 较为轻量，资源需求相对较少；
- Nginx 处理请求异步非阻塞，而 Apache 为阻塞，前者能提供更好的高并发性能；
- Apache 对 PHP 及其他动态请求的支持比 Nginx 更好；
- Apache 的稳定性较好，可用模块相对 Nginx 较多。

本地数据库

数据储存在本地目录下的 .db 文件中，表中包涵每次访问的来源 IP 和访问时间（时间戳形式）。

其他Web 编程框架

除了 Flask 外，还有：

- ASP.NET Core (C#，前端部分可配合 Razor)：微软出品，跨平台，高度专业化；
- Spring (Java)：可拓展性较强，应用广泛；
- Django：可拓展性较强，面向无代码，利于创建数据库驱动的网站。

DNS

DNS 数据库

数据库中的数据是动态更新的，主要有如下形式：

1. 手动更新：在 DNS 服务器中可手动更新记录，常在**权威 DNS 服务器**中进行；
2. DNS 服务器缓存：**本地 DNS 服务器**在收到本地没有缓存的解析时，递归查询并将结果缓存在本地；
3. DNS 传播：某域名的**权威服务器**向其**二级 DNS 服务器**更新该域名的 DNS。

DNS 反向搜索记录

因为正向搜索记录只能将主机名映射到 IP 地址，无法反向查找。反向搜索记录用于反向查找。

没有生效的 DNS 解析

修改解析后，客户端测试中没有生效，可能是本地 DNS 缓存还没有刷新。

添加下级子域

在创建的域中选择“新建域”，继续添加子域，配置子域名即可。

该子域中的主机名为 `xx.yy.zz`，其中 `yy.zz` 是创建的域名，`xx` 是子域名。

DHCP

服务器自动获取 IP

DHCP 服务器不可以选择自动获取 IP。

DHCP 服务的保留 IP 功能

设置保留 IP 功能，可为网络中的部分设备（如网管交换机、日志服务器、NAS 等）固定 IP 地址，方便在维护时访问，而不需要先查看其地址。

保留 IP 需要知道目标机器的 MAC 地址，因为 DHCP 下发的 IP 与 MAC 绑定；特别地，在 DHCP 服务器向客户机发送 DHCP Offer 时，客户机还没有拿到 IP 地址，此时需要通过 MAC 地址通讯。

获得租约

DHCP 获取 IP 租约需要 5 个步骤：

1. 发现：客户机并不知道 DHCP 服务器地址，因此广播 DHCP Discovery 报文；
2. 提供：（所有的）DHCP 服务器收到到该报文，从自己的地址池中选择1个可用地址，向该客户机发送 DHCP Offer 报文，其中包含了该地址、租期；
3. 选择：该客户机收到该报文（若有多个 DHCP 服务器，则选择收到的首个），广播 DHCP Request 报文，其中包含选择地址来源的 DHCP 服务器标识；
4. 确认：DHCP 服务器收到该消息，确认租约无误，向客户端发送 DHCP ACK 报文；
5. 查重：客户端广播 ARP 确认该地址是否已被占用；等待响应超时，认为该地址没被占用，使用该地址；否则向 DHCP 服务器发送 DHCP Decline 报文，重新申请 IP。

IP 异常

169.254.0.0/16 网段中的地址是 Windows 在没能从 DHCP 获取地址后为自己分配的地址。可能有如下问题：

1. 网络中的 DHCP 服务器不可用；
2. 网络中的 DHCP 服务器地址池穷尽；

3. 本机网卡故障，无法完成 DHCP 的租赁过程。

作用域的地址

C 类网 192.168.0.1/24 中，IP 地址 192.168.0.0 是该网络的网络号，192.168.0.255 是广播地址，这两个地址被保留不能使用。