

A Simple DNS System

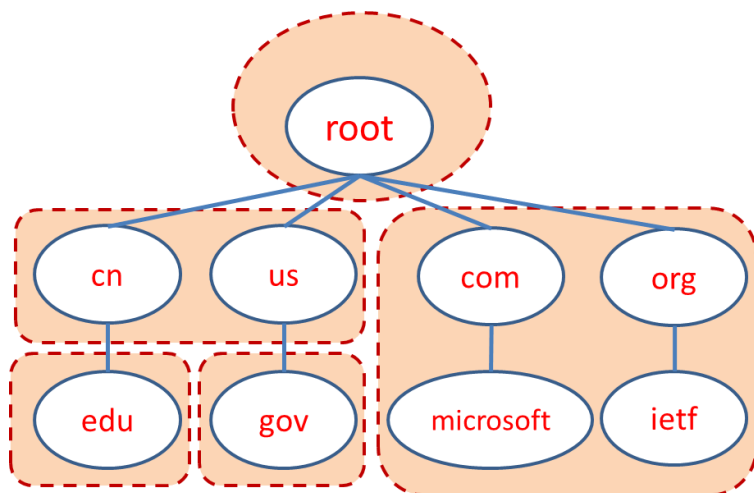
Internet Applications 课程设计任务书

实验目标

深入理解 DNS (Domain Name System) 协议的有关知识, 结合历次实验课的编程实践, 完成一个基于 Linux 命令行终端的 DNS 系统原型 (包括客户端和服务端), 实现英文域名解析。

实验要求一：基本功能

1. 实现英文域名的解析, 例如: `www.bupt.edu.cn`;
数据库记录示例:
`www.bupt.edu.cn, 86400, IN, A, 192.168.1.25`
`bupt.edu.cn, 86400, IN, MX, mx.bupt.edu.cn`
`mx.bupt.edu.cn, 86400, IN, A, 192.168.1.37`
2. 至少支持 4 个顶级域, 至少实现三级域名的解析。程序需要实现的实体有: `client`、至少 6 个 DNS server (含 local DNS server)。
4 个顶级域名: `.cn`、`.org`、`.com`、`.us`
二-三级域名: 自定义 (例如: `edu.cn`, `bupt.edu.cn` 等等)
DNS server 的部署架构可参考下图的示例:



3. 支持的 Resource Record 类型: A、MX、CNAME; 对于 MX 类型的查询, 要求在 Additional Section 中携带对应 IP 地址;
4. 支持的解析方法: 迭代解析;
5. 传输层协议:

client 与 local DNS server 之间：UDP；

DNS server 之间：TCP；

6. 应用层协议：DNS

要求通信过程中使用的所有 DNS 报文必须能够用 Wireshark 正确解析；

7. server 的数据维护方式可采用文件；

8. 书写完整的设计文档，参考 *Sample-Project-Report.doc*；

9. 程序中应包含详细的代码注释，使用良好的编程风格；

10. 程序运行稳定，支持错误处理，如：命令无效、参数缺失、参数错误、查询失败等；

11. 每两人一组，分工完成，将设计文档和源代码文件经打包、压缩后按以下规则命名：

dns_2020211501_2020211234&2020215678_李宁&张安.tar.gz

实验要求二：扩展功能

1. 支持 PTR 类型的 Resource Record；

2. 支持 cache；

3. 打印查询的 trace 记录（查询路径、服务器响应时间）；

实验环境

1. C 语言以及相关函数库；

2. gcc 编译器、gdb 调试工具、Wireshark 抓包分析工具；

3. Linux 操作系统、Linux 下的 nslookup 命令及其 help。

实验方法

1. 利用 Wireshark 抓包工具，分析 DNS 协议的工作原理；

2. 分析需求，提出解决方案，划分功能模块；

3. 实现基于 UDP 和 TCP 的 socket 程序框架，实现英文域名解析服务。