

UC Day19

预习课

预习
内容

项目概述

WEB服务器

开发流程

- 1) 需求分析：做啥？解决什么问题？实现什么功能？具体要求
- 2) 概要设计：采用什么框架？大体结构？技术路线？
- 3) 制定计划：评估每个模块的工作投入，编写开发计划和测试计划
- 4) 详细设计：设计每个模块的具体细节（库，文件，函数，数据类型，类）
- 5) 编写代码：根据详细设计实现代码
- 6) 系统测试：测试，提交测试报告
- 7) 产品发布
- 8) 总结



需求分析

- web服务器，提供静态页面（html）资源的下载，传输层基于TCP协议，应用层基于HTTP协议
- 浏览器发送http请求，服务器接收请求后进行解析，从中解析出浏览器所请求的资源路径，验证该资源是否合法并确定类型，在本地获取资源内容，组织成http响应传回web浏览器，web服务器要支持并发处理（线程）



概要设计



概要设计

- 主模块负责启动服务器模块，服务器模块借助通信模块等待并接收web浏览器的连接请求
- 服务器模块在接收到web浏览器的连接请求后，创建一个客户机线程
- 在客户机线程模块中借助通信模块接收浏览器的http请求
- 在客户机线程模块中借助http模块完成http请求的解析，提取资源路径
- 在客户机线程模块中借助资源管理模块，对资源进行验证并确定类型
- 在客户机线程模块中借助http模块完成http的响应，在通过通信模块发送响应



详细设计

http模块

http.h

http.c

通信模块

socket.h

socket.c

资源模块

resource.h

resource.c

mime.h

线程模块

client.h

client.c

信号模块

signals.h

signals.c

服务器模块

server.h

server.c

主模块

main.c



直播课见

UC

C/C++教学体系

目录

http模块

网络模块

http模块

http模块

- http模块主要负责完成针对对http协议的处理工作，具体包括
 - 1) 对所接收到的http请求内容的解析。从客户端发来的http请求，被服务器接收到后，要从请求中将后续要用到的重要数据抽离出来，并进行存储。
 - 2) 构造http响应头。服务器在接收到客户端的http请求后，要给客户端进行回传响应，响应分为相应头和响应体两部分，相应头的构造在该模块中完成。



网络模块

网络模块

- 所有和网络通信相关的功能都在本模块中实现，具体包括
 - 1) 初始化套接字。服务器传输层采用的是TCP协议，在服务器搭建时，需实现创建套接字、组织地址结构、绑定、监听等功能。
 - 2) 等待并接收客户端的连接请求。当客户端需要访问服务器时，需要完成三次握手，服务器需要等待连接请求的到来，并完成三次握手，建立和客户端的通信连接。
 - 3) 接收客户端的http请求。在服务器和客户端建立通信连接后，客户端像服务器发送http请求，服务器需接收请求并完成存储，用以后续操作使用。



网络模块

- 4) 发送响应头。相应头为http模块中所构建，在网络模块中，负责将构造好的相应头内容发送给客户端。
- 5) 发送响应体。响应体为对方所请求的具体文件内容，服务器需在本地获取该文件，读取文件的内容，并发送给客户端。
- 6) 关闭套接字。



谢谢

UC Day19

复习课

信号模块

信号模块

- 在信号模块中，要忽略大部分的信号，提高服务器的生存能力
- 注意，是忽略，不是屏蔽。
- 头文件 `signals.h`
- 源文件 `signals.c`



信号模块

- signals.h

```
1 //信号模块头文件
2 #ifndef __SIGNALS_H_
3 #define __SIGNALS_H_
4 //初始化信号 忽略大部分信号
5 int initSignals(void);
6
7 #endif //__SIGNALS_H_
```



信号模块

- signals.c

```
1 //信号模块实现
2 #include<unistd.h>
3 #include<sys/syscall.h>
4 #include<signal.h>
5 #include<stdio.h>
6 #include"signals.h"
7
8 //初始化信号
9 int initSignals(void){
10     printf("%d.%ld > 忽略大部分信号\n",getpid(),
11           syscall(SYS_gettid));
12     for(int signum = SIGHUP;signum <= SIGRTMAX;signum++){
13         if(signum != SIGINT && signum != SIGTERM){
14             signal(signum,SIG_IGN);
15         }
16     }
17     return 0;
18 }
```



下节课见