天津大学

《计算机网络》课程设计
周进度报告

题目：第一周 实现简单的 Echo Web Server

学　号：	3020202184 3020244344
姓　名：	刘锦帆 李镇州
学　院：	智能与计算学部
专　业：	计算机科学与技术
年　级：	2020 级
任课教师：	石高涛

2022 年 3 月 17 日

目　录

一 协议设计

1.1 源文件架构

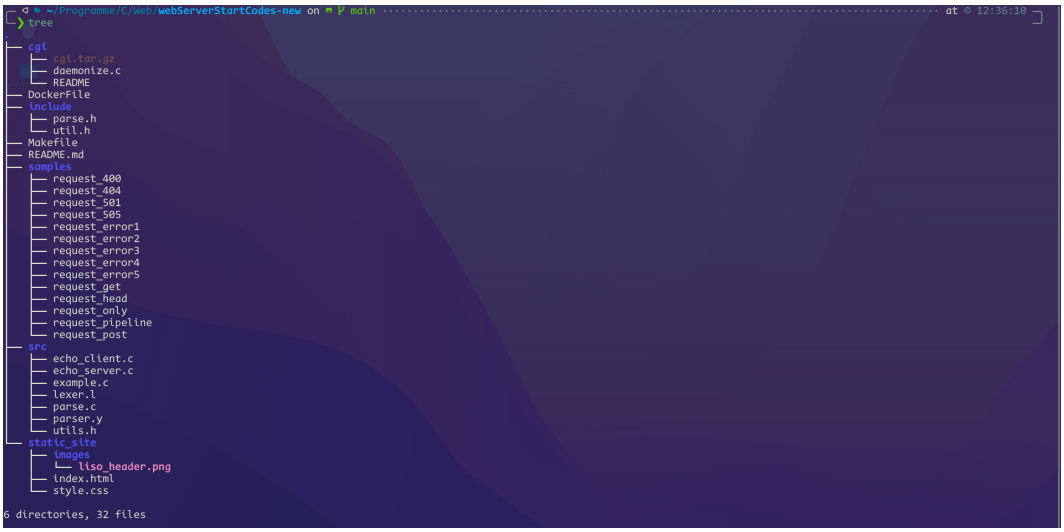


图 1-1 Structure of the Source Code

如图1-1所示，源文件由 *cgi*, *include*, *samples*, *src* 和 *static_sites* 五个文件夹和 *Makefile* 组成，其中 *include*，*src*，*samples* 和 *Makefile* 是第一周实验需要用到的文件。

- include 文件夹中包含一个 parse.h 文件，是用于解析报文的 parse.c 文件的头文件，其中定义了重要的结构体 Request_header 和 Request，并给出了共 echo_server 和 echo_client 调用的解析接口 Request* parse(char *buffer, int size,int socketFd)。（截图中的文件 util.h 是方便调试写进去的工具包：包括一些方便输出的宏定义）；
- src 文件夹中包含三组代码：负责消息解析的 parser.y，parse.c，和 lexer.l；负责测试消息解析的 example.c 以及一个网络应用程序 echo_client.c 和 echo_server.c；
- samples 文件夹中包含各类消息，用于对消息解析的测试；
- Makefile 负责管理项目的编辑。

1.2 功能模块

基础代码的功能主要包括两大部分：

1. 测试 parcer 解析的情况：由 parse.c, parser.y 等文件组成，用于调试
2. 网络应用程序，即使用 socket 进行通信的 echo_server 和 echo_client 两个

文件

1.3 消息解析

消息解析中，采用的是 yacc 和 lex 相配合的方式，二者的关系可以由图1-2指出，即 Lex 用于分词，Yacc 用于语义的翻译。

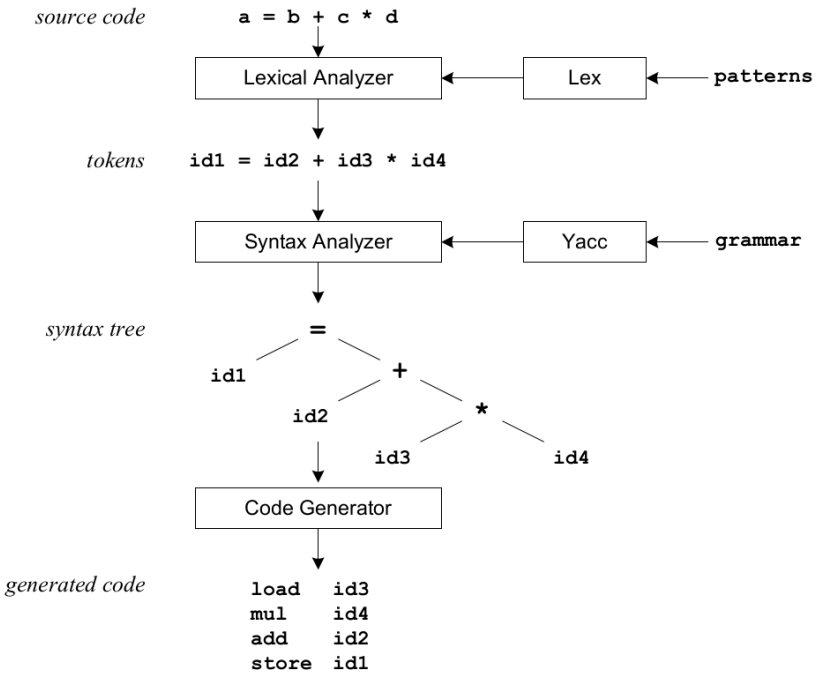


图 1-2 Yacc and Lex

具体到代码层面，我们主要关注 parser.y，即 Yacc 部分。我们需要通过添加更多的 grammar 和相关的操作，来解析更多消息的类型。

二 协议实现

第一周的协议实现主要分为“消息解析”和“消息反馈”两个部分。

2.1 消息解析

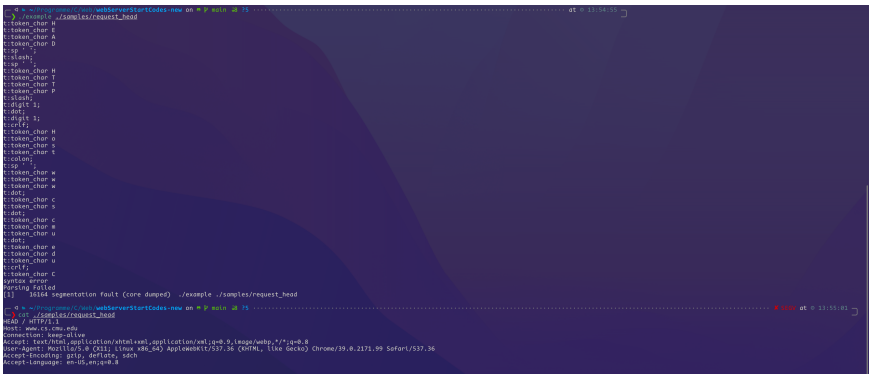


图 2-1 Segment Fault

如图2-1，通过 example.c 的测试，我们发现了第一个错误：Segment fault，且该错误总是在解析第三行消息时出现。

通过对 parser.y 和 parse.h 文件的理解，我们认为此问题应该出自其 Request 中 Request_header，它在最初定义时是一个指针 *header，所以在解析第二行时没有错误，而在进行第三行即以后的解释时，如果不进行人为的空间分配，则当然会出现上述的 Segment Fault。

所以对于这个问题，我们只需在 parser.y 中进行修改，增加 realloc 的函数，对 header 进行扩容，并且相应的 grammar 与之对应即可。

当然，还可以在 example.c 中处理一下返回错误的情况，避免直接访问 NULL 的空间导致 Segment Fault。使用批量脚本测试则效果如图2-2。

2.2 消息反馈

在完成消息解析后，我们对 echo_server.c 进行编程，完成消息的解析、处理与返回。研读源代码，梳理出如"echo_server" 代码段的伪代码。

我们只需要在第 7 行"TODO: DEAL WITH MESSAGE" 处增加如代码段“处理数据”的“解析、分类、封装返回值”即可。

- 具体而言，先通过 example.c 一样的方法解析 buf 中收到的数据。在该步骤中，可能会出现依赖问题，对此我们需要在 Makefile 中增加相关的依赖。让 echo_server 在连接时加上 parse.o 的文件即可。
- 得到 request 后，我们对 request 的两种情况进行分析

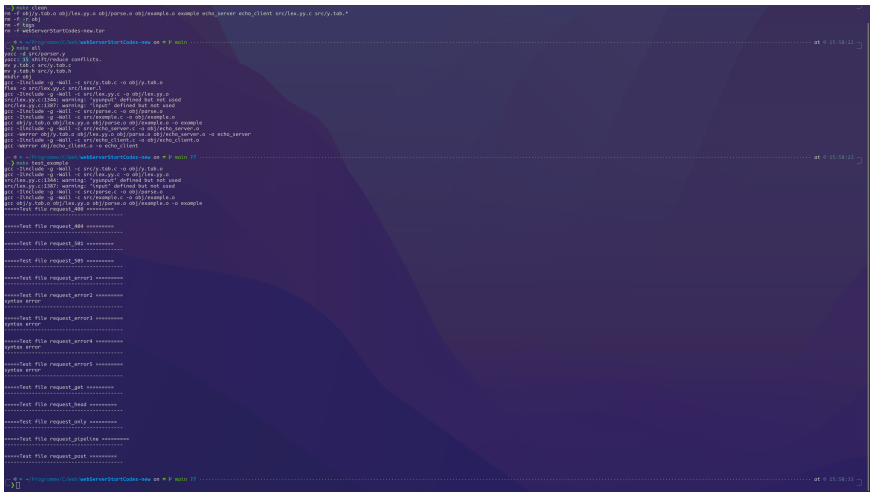


图 2-2 Example Test

- 如果 request 为 NULL，则解析不成功，即返回 400 错误
- 如果 reqeust 成功，但是发现其中 method 暂不支持，则返回 500 错误
- 否则直接返回受到的数据

echo_server

```
create socket
bind socket
listen on socket
while(1):
    accept connection
    while(recieve message):
        TODO: DEAL WITH MESSAGE
        send back
    close client socket
close socket
```

处理数据

```
def deal_with_request(buf):
    Request *request = parse(buf, BUF_SIZE, 8192)
    if request is NULL:
        return _400msg
    if method_not_support:
        return _500msg
    return buf
```

三　实验结果及分析

3.1　对消息的正确解析

在 Makefile 中增加一项 test_example 来批量测试 example，结果如图2-2所示。其中 Makefile 增加的脚本如下:

Makefile: test example

```
1 test_example: example
2 @for f in $(shell ls samples); do \
3     echo "=====Test file" $$f "=========="; \
4     ./example samples/$$f | grep Segmentation --color ; \
5     echo "--------------------------------------\n"; \
6 done
```

3.1.1　结果分析

可以看到结果只在 error2, error3, error4 和 error5 四个情况下出现了 syntax error 的返回值，因为我们在 parser.y 增加了对测试样例中的 400 做了特殊处理，所以不会出现解析错误。且在 Makefile 中将 parse.c 原本的输出给屏蔽了，所以也不会有多余输出。

3.2　实现简单的 echo web server

实验操作如下:

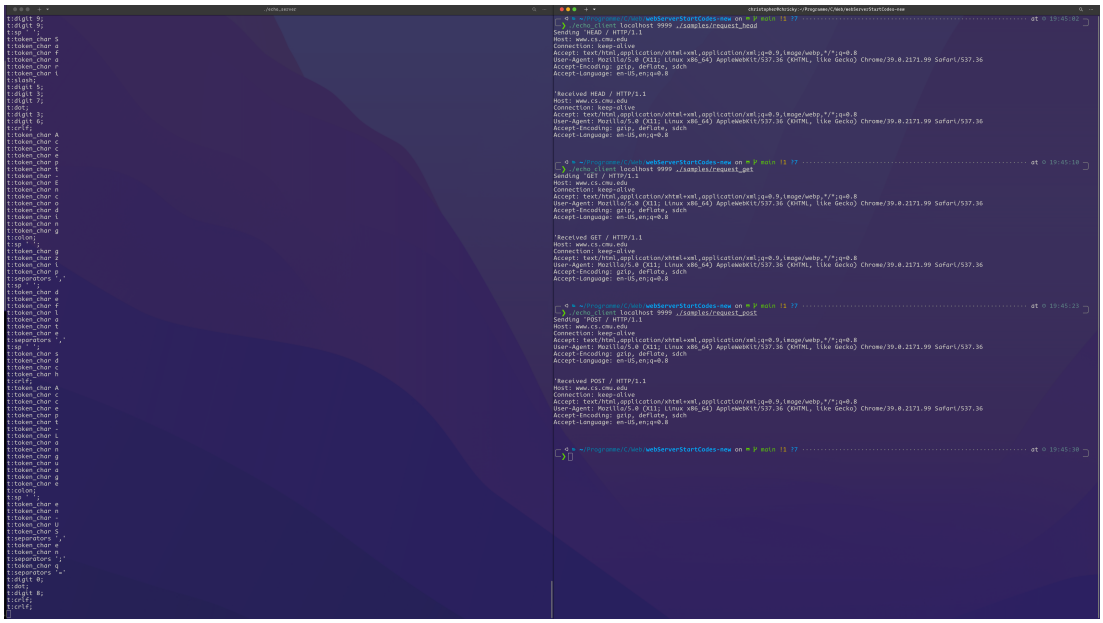
1. 在 echo_client 中增加直接从文件中读取的支持，以便直接使用提供的 samples；
2. 先打开 echo_server；
3. 然后通过 echo_client 发送一系列文件，查看 echo_server 的结果。

3.2.1　实验结果截图及分析

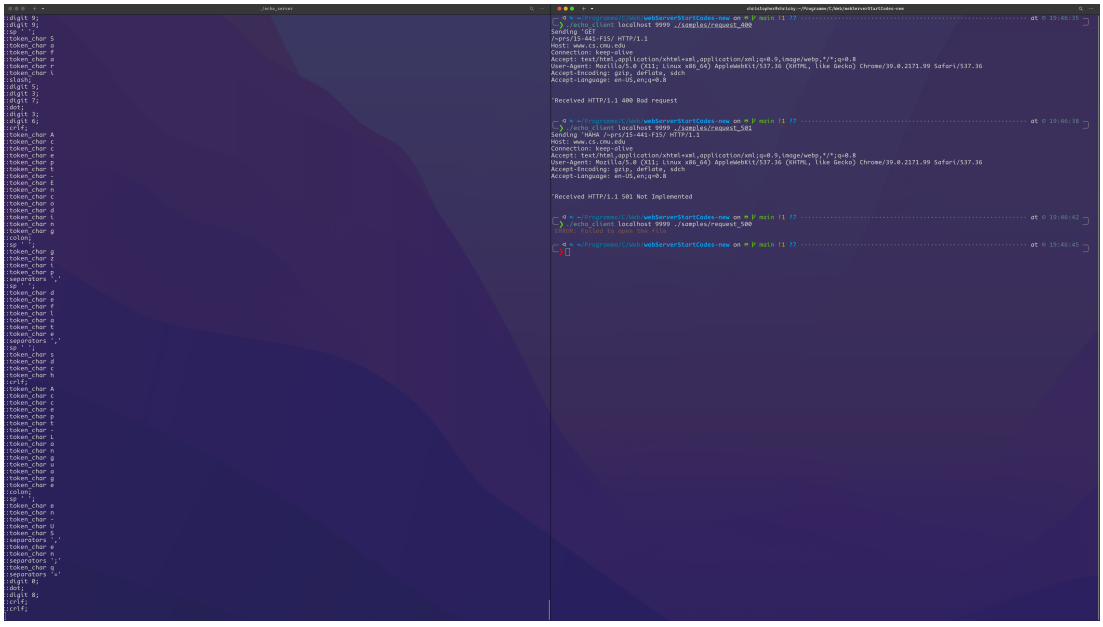
实验结果如图3-1所示。可以看到，在 head, get 和 post 方法的实验中，我们实现了 echo 的效果。在 400 和 501 的输入中，我们实现了相应的返回消息。在输入文件不存在时，我们也得到了相应的正确返回，体现了我们程序的鲁棒性。

3.2.2　上交实验平台

如图3-2，我们的程序在第一次提交及获得了满分的成绩，且在次日稍作修改（删除部分多余代码）后依然能满分通过第一周的测试点。



a) Head get and post actions



b) 400 and 501

图 3-1 echo web server

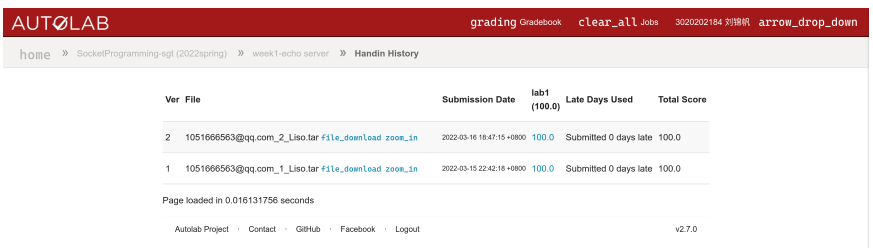


图 3-2 Auto Lab Confirm

四　进度总结及项目分工

4.1 本周进度情况

本周主要完成了对 yacc 和 lex 文件的一定研究与实践，对消息解析的原理进行了了解。同时，复习了 socket 编程的框架，实现了 socket 编程的实践部分。具体完成情况如表4-1所示。

本周任务要求	完成	没完成	备注
1、阅读 HTTP/1.1 的标准文档 RFC2616	✓		无
2、搭建编程环境	✓		无
3、熟悉 Socket 编程方法	✓		无
4、掌握 lex 和 yacc 正确解析消息（message）的方法	✓		无
5.1、实现简单的 echo web server Echo GET, HEAD, POST	✓		无
5.2、响应没有实现的方法	✓		无
5.3、响应错误的方法	✓		无
6、功能测试	✓		无

表 4-1　本周进度完成表

4.2 人员分工

人员分工如表4-2所示。

人员	项目分工
刘锦帆	完成消息解析及分发，整体项目的测试与 Debug。同时，完成实验报告相应部分的撰写
李镇州	完成 echo_server 返回消息的封装与实验报告的框架

表 4-2　人员分工表