

Homework 17

郭天魁
信息科学技术学院
1300012790

December 22, 2014

1 Homework 17

1.1 11.6

A. Tiny已经输出了header，只需要输出status line即可。

```
1 void doit(int fd)
2 {
3     // ...
4
5     /* Read request line and headers */
6     Rio_readinitb(&rio, fd);
7     Rio_readlineb(&rio, buf, MAXLINE);
8     printf("%s", buf);
9     sscanf(buf, "%s_%s_%s", method, uri, version);
10
11     // ...
12 }
```

B. 使用Chromium访问了自带的home.html (<http://localhost:6568/home.html>)，Tiny产生如下输出：

```
GET /home.html HTTP/1.1
Host: localhost:6568
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/39.0.2171.65 Chrome/39.0.2171.65 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6,zh-TW;q=0.4
```

```

GET /godzilla.gif HTTP/1.1
Host: localhost:6568
Connection: keep-alive
Cache-Control: max-age=0
Accept: image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
    like Gecko) Ubuntu Chromium/39.0.2171.65 Chrome/39.0.2171.65 Safari
    /537.36
Referer: http://localhost:6568/home.html
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6,zh-TW;q=0.4

```

C. HTTP/1.1。

D.

Host: 请求资源所在的域名/IP以及端口号。例如在Apache中会利用Host来连接对应的VirtualHost。本例中为Tiny运行的IP及端口。

Connection: 指定完成本次连接后是否断开连接。本例中为保持连接。

Cache-Control: 指定当前对象的一系列cacheable信息，如不缓存，Age阈值，freshness阈值等等。本例中为可缓存本次获取的对象，但不使用之前的缓存内容。

Accept: 指定接受的介质类型。本例中为首选网页文件和图片文件。

User-Agent: 浏览器信息。本例中为Ubuntu Chromium。

Referer: 来源网页信息。本例中图片文件来源为home.html。

Accept-Encoding: 指定接受的编码方法，通常为压缩方法。本例中为首选gzip压缩格式。

Accept-Language: 指定接受的语言。本例中为首选中文。

1.2 11.7

```

1 int endswith(char *filename, char *ext) {
2     char *dot = strrchr(filename, '.');
3     return (dot && !strcmp(dot, ext));
4 }
5
6 int endswithany(char *filename, char *exts[], int n) {
7     int i;
8     for (i = 0; i < n; i++)
9         if (endswith(filename, exts[i]))
10             return 1;
11     return 0;
12 }
13
14 /*
15  * get_filetype - derive file type from file name
16  */
17 void get_filetype(char *filename, char *filetype)
18 {
19     const char *mpg[] = {".mpg", ".mpeg", ".mp1", ".mp2", ".mp3", "
        .m1v", ".m1a", ".m2a", ".mpa", ".mpv"};

```

```

20     const char *mp4[] = {".mp4", ".m4a", ".m4p", ".m4b", ".m4r", ".
        m4v"};
21     if (endswith(filename, ".html"))
22         strcpy(filetype, "text/html");
23     else if (endswith(filename, ".gif"))
24         strcpy(filetype, "image/gif");
25     else if (endswith(filename, ".jpg"))
26         strcpy(filetype, "image/jpeg");
27     else if (endswithany(filename, mpg, sizeof(mpg) / sizeof(char
        *)))
28         strcpy(filetype, "video/mpeg");
29     else if (endswithany(filename, mp4, sizeof(mp4) / sizeof(char
        *)))
30         strcpy(filetype, "video/mp4");
31     else
32         strcpy(filetype, "text/plain");
33 }

```

另外，你需要将所有的Rio_前缀替换为rio_。