C语言构建WEB管理系统（五）：CGI实现上传文件 - 程序猿的挨踢人生 - CSDN博客

在很多网站中经常会遇到有向服务器上传文件的情况，比如在博客或空间中上传自己的头像。这一节我们来看一下在后台如何使用C语言实现文件上传这一功能。

首先创建一个html文档来上传文件，然后使用wireshark抓取数据包来分析一下上传文件的文件内容如何解析。html文档如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />

        <title>CGI 实现文件上传</title>

    </head>
    <body>
        <form action="/cgi-bin/upload.cgi" method="post" enctype="multipart/form-data">
            <input type="file" name="upfilename" value="fname" /> <br>
            <input type="file" name="upfilename_sec" value="fname_sec" />

            <p><input type="submit" value="提交" /></p>
        </form>
    </body>
</html>
```

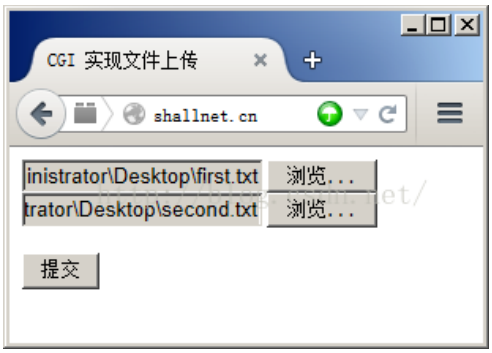
下面为wireshark抓包获取的数据：

所以要解析出文件内容需要首先从content_type中获取boundry，以下为在解析出boundry基础上将文件内容写入到一个临时文件中的处理函数：

```
int sln_cgi_input_multi_content_parse(const char *content_input,
                                      int content_length, const char *boundary)
{
    char        attr[SLN_ATTR_MAX], value[SLN_MAX_VALUE_LEN],
                filename[SLN_MAX_FILENAME_LEN], tfilename[SLN_MAX_FILENAME_LEN],
                tmpname[SLN_MAX_FILENAME_LEN];
    char        *start, *end, *pname, *quote;
    int         boundary_len, attr_len, value_len, filename_len, file_size = 0;
    FILE        *fp = NULL;

    boundary_len = strlen(boundary);

    start = (char *)content_input;

    if ((end = strstr(start, boundary))) {
        start = end + boundary_len;
    }

    while (start) {
        attr[0] = '\0', filename[0] = '\0', filename[0] = '\0', tfilename[0] = '\0';
        if (0 == memcmp(start, "--\r\n", 4)) {
            break;
        } else {
            start += 2;
        }


        end = strstr(start, "\r\n\r\n");
        if (NULL == end) {
            break;
        }
        *end = '\0';


        pname = strstr(start, "name=\"");
        if (NULL == pname) {
            start = end + 1;
            continue;
        }
        quote = strchr(pname + 6, '\"');
        if (NULL == quote) {
            start = end + 1;
            continue;
        }

        attr_len = quote - (pname + 6);
        strncpy(attr, pname + 6, attr_len);
        attr[attr_len] = '\0';
        printf("<p>attr(len=%d): %s</p>\n", attr_len, attr);


        pname = strstr(start, "filename=\"");
        if (NULL != pname) {
            quote = strchr(pname + 10, '\"');
            if (NULL == quote) {
                start = end + 1;
                continue;
            }
            filename_len = quote - (pname + 10);
            strncpy(filename, pname + 10, filename_len);
            filename[filename_len] = '\0';
            printf("<p>filename(len=%d): %s</p>\n", filename_len, filename);
        }

        start = end + 4;
        if ('\0' == filename[0]) {
            end = strstr(start, boundary);
            if (NULL != end) {
                value_len = (end-4) - start;
                if (value_len > sizeof(value)) {
                    value_len = sizeof(value) - 1;
                }
                strncpy(value, start, value_len);
                value[value_len - 1] = '\0';
                printf("<p>value(len=%d): %s</p>\n", value_len, value);
```

```
                start = end + strlen(boundary);
            }
        } else {
            end = sln_memsearch(start, content_length - (start-content_input), boundary, boundary_len);
            if (NULL != end) {
                file_size = (end-4) - start;



                if (NULL == tmpnam(tmpname)) {
                    snprintf(tfilename, SLN_MAX_FILENAME_LEN, "%s", filename);
                } else {
                    snprintf(tfilename, SLN_MAX_FILENAME_LEN, "%s", tmpname);
                }


                file_size =
                    file_size > SLN_MAX_CONTENT_LEN ? SLN_MAX_CONTENT_LEN : file_size;
                fp = fopen(tfilename, "w+");
                if (NULL != fp) {
                    fwrite(start, 1, file_size, fp);
                    fclose(fp);
                } else {

                }
                printf("<p>tfilename: %s, file_size: %d</p>\n", tfilename, file_size);

                start = end + strlen(boundary);
            }
        }
    }

    return 0;
}
```

最后将代码编译成可执行文件upload.cgi，在页面上传两个文件，如下：



提交后页面跳转到如下页面：

我们将所示文件cat出来看一下文件内容，如下：

```
[root@shallnet 4_upload]# cat /tmp/fileke5tvT
The strings in the environment list are of the form name=value.[root@shallnet 4_upload]#
[root@shallnet 4_upload]# cat /tmp/file7dP091
This page is part of release 3.22 of the Linux man-pages project.[root@shallnet 4_upload]#
```

可以看到文件上传成功！