# 模块功能需求分析：

TFTP分为TFTP客户端（client）和TFTP服务器（Server）两部分；

**基本功能：**

是实现TFTP客户端能够往TFTP服务器上 ,上传一个几十/几百MB的文件，也能从TFTP服务器上,下载一个几十/几百MB的文件；

**扩展功能：**

①客户端可以指定传输使用的Blksize（512/1024/1468/2048B/4096Byte等）；

②客户端下载有进度条显示；

③客户端可以在下载时重命名/选择覆盖；

④可支持超时重传；

⑤可支持断点续传（自定义扩展选项为bpid，暂未实现）；

⑥可支持BlkSize重复利用（实现TFTP的超大文件传输）；

⑦服务器支持多客户端同时上传/下载；

⑧服务器有客户端的操作记录日志；

⑨服务器有debug开关进行有关数据、任务的查看；

等。

# 基本功能描述：

## 2.1、协议格式流程介绍：

协议支持六种报文类型，分别为：读请求、写请求、ACK回复、DATA数据、ERROR报文、OPTIONAL选项请求、OACK选项回复ACK报文。如下：

opcode operation

1 Read request (RRQ)

2 Write request (WRQ)

3 Data (DATA)

4 Acknowledgment (ACK)

5 Error (ERROR)

6 Acknowledge of Request and options (OACK)

对于报文的格式如下：

## 2.2、基础模块实现描述：

### 2.2.1、shell功能模块

以链表结构实现命令的注册、命令管理、命令动态注册、命令提示、命令格式化输入、命令解析、命令匹配执行等功能，比如debug命令用来实现debug信息的输出，以及debug信息内容的可选择性输出（debug信息所打印的当前taskName等）。

命令注册的接口为：

tftp\_shell\_cmd\_register

(

tftp\_cmd\_deal\_fun function,

tftpCmdAbil\_t ability,

CHAR \* cmd\_str

);

function为注册命令的回调函数，ability为注册命令的特性与能力，如：是否隐藏、是否动态注册等，cmd\_str为格式化的命令注册字符串，为命令的子命令与子命令描述信息的格式化字符串，由tftp\_shell\_cmd\_register负责解析并挂载到命令注册管理的链表中去。

以如下为例：

**tftp\_shell\_cmd\_register((tftp\_cmd\_deal\_fun)tftp\_shell\_cmd\_display,**

**\_\_TFTP\_CMD\_NORMAL\_ | \_\_TFTP\_CMD\_DYN\_,**

**"shellcmd{Display command with format for special command}"**

**"display{display command format}"**

**"\_\_STRING\_\_{command string}");**

注册命令”shellcmd display \_\_STRING\_\_”，命令用于显示某一条命令的详细格式以及信息。\_\_TFTP\_CMD\_NORMAL\_ 和 \_\_TFTP\_CMD\_DYN\_表示该条命令可以正常显示，并且默认注册，\_\_STRING\_\_表示输入的信息为字符串（该字符串如果匹配到注册的某条命令就会把该命令的命令格式以及命令描述输出），实际的功能即：遍历命令管理的链表，匹配字符串，并将匹配到的命令格式化输出，如tftpserver命令为启动tftp服务器，命令格式如下：

tftp>**shellcmd display tftpserver**

command list detail information: -------------------------------------<<<-command:tftpserver->>>------------------------------------

tftpserver: tftp server enable/disable

\_\_STRING\_\_: enable or disable

\_\_IPADDR\_\_: server ip address

---------------------------------------------------------------------------------------------------

此外有一条隐藏的命令：dynamic command \_\_STRING\_\_ enable/disable，用于使能/隐藏可以动态注册的命令（默认都是可以动态注册的，除了dynamic命令外，由于dynamic不对外呈现，且不能被删除，因此除了调试外，基本别无他用）：

tftp>d**ynamic command all enable**

sem is already dynamic register success!

tftptask is already dynamic register success!

taskpool is already dynamic register success!

tftp>**shellcmd display tftptask**

command list detail information:

--------------------------------------<<<-command:tftptask->>>-------------------------------------

tftptask: all childs task information for tftp process display

display: display some information

taskId: display with tid

\_\_INT32\_\_: task tid(-1 is all)

---------------------------------------------------------------------------------------------------

tftp>

### 2.2.2、log功能模块

Log功能即日志功能，用来实现一些正常/异常/debug的信息打印与记录保存，分为读个级别：normal正常打印、note提示打印、debug调试打印、warn警告打印、error错误打印等，其中记录日志到日志文件中（当前所有日志同时都会默认打印到shell），日志文件以级别命名：tftpDebug.log、tftpError.log、tftpNormal.log、tftpNote.log、tftpWarn.log等。debug日志需要debug命令开启对应debug开关才可以打印出来，默认都是关闭debug信息的。

Debug开关命令格式为：

tftp>**shellcmd display tftplogDbg**

command list detail information:

-------------------------------------<<<-command:tftplogDbg->>>------------------------------------

tftplogDbg: tftp log debug switch

debug: debug switch open or close

\_\_UINT32\_\_: switch choose[eg:

->(0)task

->(1)server

->(2)client

->(3)shell

->(4)sem

->(5)send

->(6)recv

->(7)pack

->(8)other]

\_\_STRING\_\_: open/close

---------------------------------------------------------------------------------------------------

此外tftplogInfo用来读取日志文件并显示，和tftplogTask用来打开/关闭日志记录中的taskName选项。

tftp>**shellcmd display tftplogTask**

command list detail information:

------------------------------------<<<-command:tftplogTask->>>------------------------------------

tftplogTask: tftp log task name optional

taskname: task name for logging display

\_\_STRING\_\_: open/close

---------------------------------------------------------------------------------------------------

如当taskName选项打开时，log信息打印如下：

tftp>**tftplogTask taskname open**

open task name logging switch

tftp>**tftpserver enable 192.168.10.100**

2019-10-02 21:33:41(UTC)[Wednesday] (tftpShellTask) %% TFTP-ERROR :(65)bind socket fd fail, sockfd = 8, port = 69, errno = 99

### 2.2.3、task功能模块

### 2.2.4、sem功能模块

### 2.2.5、server功能模块

### 2.2.6、client功能模块

### 2.2.7、cfg功能模块

## 2.3、扩展功能模块与创新点描述：

### 2.3.1、md5功能（已实现）：

计算文件的MD5值，可以用Linux自带的md5sum命令实现，system(“md5sum filePath/fileName”);而此处之前有自己实现过，因此拷贝过来即可，比起md5sum的效率慢多了（1G得算几十秒），可以优化：一次性多读取一些字节，如2048Byte而不是64Byte。此处不再优化，tftp一般传输的文件也就几十兆，最大百十来兆，计算MD5的耗时也就几秒，暂时不考虑优化。

### 2.3.2、超时重传功能（已实现）：

采用select函数实现timeout重传的的功能，在网络波动时既可以等待一定时间重发送，防止报文未发送成功或者对端未接收到，但是有重传时间和次数限制，保证让线程池中线程或者客户端不会永久性阻塞下去，并提示信息，以达到故障排查的目的。

### 2.3.3、断点续传功能（未实现）：

断点续传指的是，网络中断，长时间无法恢复，在此文件已经传输部分会被临时保存，并且创建一个断点信息的断点文件（如客户端/服务器地址、文件名、文件大小、已经传输大小、当时采用blksize、断点blkid、断点blkid被复用的轮数等信息），当数小时、数天、数月之后再次传输该文件时，会检查断点文件，如果确认传输的文件与之前的断点文件为同一个文件，则启用断点续传机制：blkid从非0开始，该功能应该是私有协议，因为标准字段并无bpid的定义。

# 3、TFTP实现架构：

## 3.1、客户端流程：

## 3.2、服务器流程：

# 4、TFTP测试注意点：

# 5、参考资料：

RFC1350（RFC783已作废）：THE TFTP PROTOCOL (REVISION 2)

RFC2347（RFC1782已作废）： TFTP Option Extension

RFC2348（RFC1783已作废）：TFTP Blocksize Option

RFC2349（RFC1784已作废）：TFTP Timeout Interval and Transfer Size Options

RFC1785：TFTP Option Negotiation Analysis