FTP服务器项目传输数据协议手册

目录

1. FTP服务器客户端发送LS命令数据包协议 ---------------1
2. FTP服务器客户端发送GET命令数据包协议---------------2
3. FTP服务器客户端发送PUT命令数据包协议---------------4
4. FTP服务器客户端发送BYE命令数据包协议---------------6

1.FTP服务器客户端发送LS命令数据包协议

1. 客户端发送形式，每帧包含 10个字节（十六进制）：

数据下标 数据名 数值范围 数据名含义

Byte0: 0xC0 0x00~0xFF 包头标志

Byte1: pkg\_len 0x00~0xFF 将pkg\_len的低8位存入数组

Byte2: pkg\_len 0x00~0xFF 将pkg\_len右移8位后的低8位存入数组

Byte3: pkg\_len 0x00~0xFF 将pkg\_len右移16位后的低8位存入数组

Byte4: pkg\_len 0x00~0xFF 将pkg\_len右移24位后的低8位存入数组

Byte5: cmdnum 0x00~0xFF 将cmdnum的低8位存入数组

Byte6: cmdnum 0x00~0xFF 将cmdnum右移8位后的低8位存入数组

Byte7: cmdnum 0x00~0xFF 将cmdnum右移16位后的低8位存入数组

Byte8: cmdnum 0x00~0xFF 将cmdnum右移24位后的低8位存入数组

Byte9: 0xF0 0x00~0xFF 包尾标志

具体表示如下表所示:

|  |  |  |  |
| --- | --- | --- | --- |
| 0xC0(1)包头 | pkg\_len(4) 数据包长度 | cmdnum(4)命令名 | 0xF0 |

(注:括号里面的数字表示该数据占用的字节数)

1. 服务器发送形式，每帧至少包含 13个字节（十六进制）:

数据下标 数据名 数值范围 数据名含义

Byte0: 0xC0 0x00~0xFF 包头标志

Byte1: pkg\_len 0x00~0xFF 将pkg\_len的低8位存入数组

Byte2: pkg\_len 0x00~0xFF 将pkg\_len右移8位后的低8位存入数组

Byte3: pkg\_len 0x00~0xFF 将pkg\_len右移16位后的低8位存入数组

Byte4: pkg\_len 0x00~0xFF 将pkg\_len右移24位后的低8位存入数组

Byte5: cmdnum 0x00~0xFF 将cmdnum的低8位存入数组

Byte6: cmdnum 0x00~0xFF 将cmdnum右移8位后的低8位存入数组

Byte7: cmdnum 0x00~0xFF 将cmdnum右移16位后的低8位存入数组

Byte8: cmdnum 0x00~0xFF 将cmdnum右移24位后的低8位存入数组

Byte9: resp\_len 0x00~0xFF 将resp\_len的低8位存入数组

Byte10: resp\_len 0x00~0xFF 将resp\_len右移24位后的低8位存入数组

Byte11: resp\_len 0x00~0xFF 将resp\_len右移24位后的低8位存入数组

Byte12: resp\_len 0x00~0xFF 将resp\_len右移24位后的低8位存入数组

Byte13~x: filenames 0x00~0xFF 需要的数据长度

Bytex+1: result 0x00~0xFF 执行是否成功的标志

Bytex+2: 0xF0 0x00~0xFF 包尾标志

具体表示如下表所示:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0xC0(1)包头 | pkg\_len(4) 数据包长度 | cmdnum(4)命令名 | resp\_len(4)文件名 | Filenames(x)需要的数据 | result(1)  执行标志 | 0xF0(1)包尾 |

2.FTP服务器客户端发送GET命令数据包协议

1. 客户端发送形式，每帧至少包含 22个字节（十六进制）：

数据下标 数据名 数值范围 数据名含义

Byte0: 0xC0 0x00~0xFF 包头标志

Byte1: pkg\_len 0x00~0xFF 将pkg\_len的低8位存入数组

Byte2: pkg\_len 0x00~0xFF 将pkg\_len右移8位后的低8位存入数组

Byte3: pkg\_len 0x00~0xFF 将pkg\_len右移16位后的低8位存入数组

Byte4: pkg\_len 0x00~0xFF 将pkg\_len右移24位后的低8位存入数组

Byte5: cmdnum 0x00~0xFF 将cmdnum的低8位存入数组

Byte6: cmdnum 0x00~0xFF 将cmdnum右移8位后的低8位存入数组

Byte7: cmdnum 0x00~0xFF 将cmdnum右移16位后的低8位存入数组

Byte8: cmdnum 0x00~0xFF 将cmdnum右移24位后的低8位存入数组

Byte9: filecount 0x00~0xFF 将filecount的低8位存入数组

Byte10: filecount 0x00~0xFF 将filecount右移8位后的低8位存入数组

Byte11: filecount 0x00~0xFF 将filecount右移16位后的低8位存入数组

Byte12: filecount 0x00~0xFF 将filecount右移24位后的低8位存入数组

Byte13: arg\_len 0x00~0xFF 将arg\_len的低8位存入数组

Byte14: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte15: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte16: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte16~20: arg 0x00~0xFF 需要下载的文件名

Bytex21+x: result 0x00~0xFF 执行是否成功的标志

Bytex22+x: 0xF0 0x00~0xFF 包尾标志

具体表示如下表所示:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0xC0(1)包头 | pkg\_len(4) 数据包长度 | cmdnum(4)命令名 | filecount(4)传输文件数 | arg\_len(4)  要传输的文件名长度 | arg(4)  要传输的文件名 | 0xF0(1)包尾 |

(注:括号里面的数字表示该数据占用的字节数)

1. 服务器接收回复形式，每帧至少包含 23个字节（十六进制）:

数据下标 数据名 数值范围 数据名含义

Byte0: 0xC0 0x00~0xFF 包头标志

Byte1: pkg\_len 0x00~0xFF 将pkg\_len的低8位存入数组

Byte2: pkg\_len 0x00~0xFF 将pkg\_len右移8位后的低8位存入数组

Byte3: pkg\_len 0x00~0xFF 将pkg\_len右移16位后的低8位存入数组

Byte4: pkg\_len 0x00~0xFF 将pkg\_len右移24位后的低8位存入数组

Byte5: cmdnum 0x00~0xFF 将cmdnum的低8位存入数组

Byte6: cmdnum 0x00~0xFF 将cmdnum右移8位后的低8位存入数组

Byte7: cmdnum 0x00~0xFF 将cmdnum右移16位后的低8位存入数组

Byte8: cmdnum 0x00~0xFF 将cmdnum右移24位后的低8位存入数组

Byte9: filecount 0x00~0xFF 将filecount的低8位存入数组

Byte10: filecount 0x00~0xFF 将filecount右移8位后的低8位存入数组

Byte11: filecount 0x00~0xFF 将filecount右移16位后的低8位存入数组

Byte12: filecount 0x00~0xFF 将filecount右移24位后的低8位存入数组

Byte13: arg\_len 0x00~0xFF 将arg\_len的低8位存入数组

Byte14: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte15: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte16: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte16~20: arg 0x00~0xFF 需要下载的文件名

Byte21+x: filedata 0x00~0xFF 需要下载的文件名中的数据

Bytex22+x: result 0x00~0xFF 执行是否成功的标志

Bytex23+x: 0xF0 0x00~0xFF 包尾标志

具体表示如下表所示:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0xC0(1)包头 | pkg\_len(4) 数据包长度 | cmdnum(4)命令名 | filecount(4)传输文件数 | arg\_len(4)  要传输的文件名长度 | arg(4)  要传输的文件名 | Filenames(x)需要的数据 | result(1)  执行标志 | 0xF0(1)包尾 |

3.FTP服务器客户端发送PUT命令数据包协议

1. 客户端发送形式，每帧至少包含 22个字节（十六进制）：

数据下标 数据名 数值范围 数据名含义

Byte0: 0xC0 0x00~0xFF 包头标志

Byte1: pkg\_len 0x00~0xFF 将pkg\_len的低8位存入数组

Byte2: pkg\_len 0x00~0xFF 将pkg\_len右移8位后的低8位存入数组

Byte3: pkg\_len 0x00~0xFF 将pkg\_len右移16位后的低8位存入数组

Byte4: pkg\_len 0x00~0xFF 将pkg\_len右移24位后的低8位存入数组

Byte5: cmdnum 0x00~0xFF 将cmdnum的低8位存入数组

Byte6: cmdnum 0x00~0xFF 将cmdnum右移8位后的低8位存入数组

Byte7: cmdnum 0x00~0xFF 将cmdnum右移16位后的低8位存入数组

Byte8: cmdnum 0x00~0xFF 将cmdnum右移24位后的低8位存入数组

Byte9: filecount 0x00~0xFF 将filecount的低8位存入数组

Byte10: filecount 0x00~0xFF 将filecount右移8位后的低8位存入数组

Byte11: filecount 0x00~0xFF 将filecount右移16位后的低8位存入数组

Byte12: filecount 0x00~0xFF 将filecount右移24位后的低8位存入数组

Byte13: arg\_len 0x00~0xFF 将arg\_len的低8位存入数组

Byte14: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte15: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte16: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte16~20: arg 0x00~0xFF 需要上传的文件名

Byte21+x: filedata 0x00~0xFF 需要上传的文件名中的数据

Bytex22+x: 0xF0 0x00~0xFF 包尾标志

具体表示如下表所示:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0xC0(1)包头 | pkg\_len(4) 数据包长度 | cmdnum(4)命令名 | filecount(4)传输文件数 | arg\_len(4)  要传输的文件名长度 | arg(4)  要传输的文件名 | Filenames(x)需要的数据 | 0xF0(1)包尾 |

1. 服务器接收回复形式，每帧至少包含 23个字节（十六进制）:

数据下标 数据名 数值范围 数据名含义

Byte0: 0xC0 0x00~0xFF 包头标志

Byte1: pkg\_len 0x00~0xFF 将pkg\_len的低8位存入数组

Byte2: pkg\_len 0x00~0xFF 将pkg\_len右移8位后的低8位存入数组

Byte3: pkg\_len 0x00~0xFF 将pkg\_len右移16位后的低8位存入数组

Byte4: pkg\_len 0x00~0xFF 将pkg\_len右移24位后的低8位存入数组

Byte5: cmdnum 0x00~0xFF 将cmdnum的低8位存入数组

Byte6: cmdnum 0x00~0xFF 将cmdnum右移8位后的低8位存入数组

Byte7: cmdnum 0x00~0xFF 将cmdnum右移16位后的低8位存入数组

Byte8: cmdnum 0x00~0xFF 将cmdnum右移24位后的低8位存入数组

Byte9: filecount 0x00~0xFF 将filecount的低8位存入数组

Byte10: filecount 0x00~0xFF 将filecount右移8位后的低8位存入数组

Byte11: filecount 0x00~0xFF 将filecount右移16位后的低8位存入数组

Byte12: filecount 0x00~0xFF 将filecount右移24位后的低8位存入数组

Byte13: arg\_len 0x00~0xFF 将arg\_len的低8位存入数组

Byte14: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte15: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte16: arg\_len 0x00~0xFF 将arg\_len右移24位后的低8位存入数组

Byte16~20: arg 0x00~0xFF 需要上传的文件名

Byte21+x: filedata 0x00~0xFF 需要上传的文件名中的数据

Bytex22+x: result 0x00~0xFF 执行是否成功的标志

Bytex23+x: 0xF0 0x00~0xFF 包尾标志

具体表示如下表所示:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0xC0(1)包头 | pkg\_len(4) 数据包长度 | cmdnum(4)命令名 | filecount(4)传输文件数 | arg\_len(4)  要的文件名长度 | arg(4)  要传输的文件名 | result(1)  执行标志 | 0xF0(1)包尾 |

(注:括号里面的数字表示该数据占用的字节数)

1. FTP服务器客户端发送BYE命令数据包协议
2. 客户端发送形式，每帧包含 10个字节（十六进制）：

数据下标 数据名 数值范围 数据名含义

Byte0: 0xC0 0x00~0xFF 包头标志

Byte1: pkg\_len 0x00~0xFF 将pkg\_len的低8位存入数组

Byte2: pkg\_len 0x00~0xFF 将pkg\_len右移8位后的低8位存入数组

Byte3: pkg\_len 0x00~0xFF 将pkg\_len右移16位后的低8位存入数组

Byte4: pkg\_len 0x00~0xFF 将pkg\_len右移24位后的低8位存入数组

Byte5: cmdnum 0x00~0xFF 将cmdnum的低8位存入数组

Byte6: cmdnum 0x00~0xFF 将cmdnum右移8位后的低8位存入数组

Byte7: cmdnum 0x00~0xFF 将cmdnum右移16位后的低8位存入数组

Byte8: cmdnum 0x00~0xFF 将cmdnum右移24位后的低8位存入数组

Byte9: 0xF0 0x00~0xFF 包尾标志

具体表示如下表所示:

|  |  |  |  |
| --- | --- | --- | --- |
| 0xC0(1)包头 | pkg\_len(4) 数据包长度 | cmdnum(4)命令名 | 0xF0 |

(注:括号里面的数字表示该数据占用的字节数)

1. 服务器发送形式，每帧至少包含 13个字节（十六进制）:

数据下标 数据名 数值范围 数据名含义

Byte0: 0xC0 0x00~0xFF 包头标志

Byte1: pkg\_len 0x00~0xFF 将pkg\_len的低8位存入数组

Byte2: pkg\_len 0x00~0xFF 将pkg\_len右移8位后的低8位存入数组

Byte3: pkg\_len 0x00~0xFF 将pkg\_len右移16位后的低8位存入数组

Byte4: pkg\_len 0x00~0xFF 将pkg\_len右移24位后的低8位存入数组

Byte5: cmdnum 0x00~0xFF 将cmdnum的低8位存入数组

Byte6: cmdnum 0x00~0xFF 将cmdnum右移8位后的低8位存入数组

Byte7: cmdnum 0x00~0xFF 将cmdnum右移16位后的低8位存入数组

Byte8: cmdnum 0x00~0xFF 将cmdnum右移24位后的低8位存入数组

Byte9: resp\_len 0x00~0xFF 将resp\_len的低8位存入数组

Byte10: resp\_len 0x00~0xFF 将resp\_len右移24位后的低8位存入数组

Byte11: resp\_len 0x00~0xFF 将resp\_len右移24位后的低8位存入数组

Byte12: resp\_len 0x00~0xFF 将resp\_len右移24位后的低8位存入数组

Byte13~x: filenames 0x00~0xFF 需要的数据长度

Bytex+1: result 0x00~0xFF 执行是否成功的标志

Bytex+2: 0xF0 0x00~0xFF 包尾标志

具体表示如下表所示:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0xC0(1)包头 | pkg\_len(4) 数据包长度 | cmdnum(4)命令名 | resp\_len(4)文件名 | Filenames(x)需要的数据 | result(1)  执行标志 | 0xF0(1)包尾 |