



(12) 发明专利

(10) 授权公告号 CN 113965561 B

(45) 授权公告日 2023. 08. 25

(21) 申请号 202111219482.6

G06F 9/50 (2006.01)

(22) 申请日 2021.10.20

G06F 9/54 (2006.01)

(65) 同一申请的已公布的文献号

申请公布号 CN 113965561 A

(56) 对比文件

CN 107743137 A, 2018.02.27

CN 108076106 A, 2018.05.25

CN 111835786 A, 2020.10.27

CN 112019623 A, 2020.12.01

CN 112035255 A, 2020.12.04

CN 112637343 A, 2021.04.09

CN 112685148 A, 2021.04.20

CN 113422842 A, 2021.09.21

US 2008133654 A1, 2008.06.05

WO 2010148882 A1, 2010.12.29

WO 2017112326 A1, 2017.06.29

鲁先志. 基于SEDA的服务器端应用程序设计与实现. 中国优秀硕士学位论文全文数据库. 2009, (第06期), 全文.

(43) 申请公布日 2022.01.21

(73) 专利权人 中电科航空电子有限公司

地址 610000 四川省成都市高新区百川街9号

(72) 发明人 宋光璠 周银辉 冉进刚 陈柯龙 王远

(74) 专利代理机构 成都行之专利代理有限公司 51220

专利代理师 张杨

审查员 谢思敏

(51) Int. Cl.

H04L 67/06 (2022.01)

H04L 69/16 (2022.01)

G06F 3/06 (2006.01)

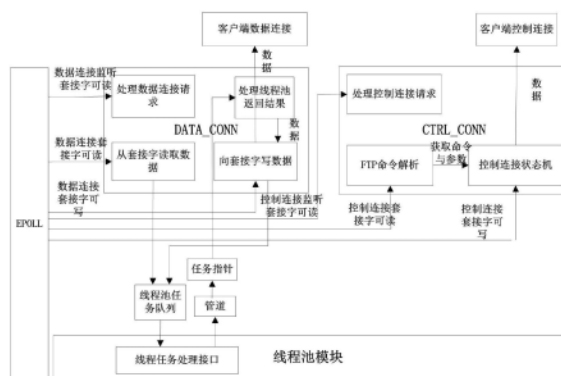
权利要求书3页 说明书9页 附图5页

(54) 发明名称

一种基于异步事件驱动的机载文件传输系统

(57) 摘要

本发明涉及文件传输领域,公开了一种基于异步事件驱动的机载文件传输系统,包括控制连接模块、数据连接模块、线程池模块和连接资源清理模块,数据连接模块:数据通过所述数据连接模块在网络、套接字数据缓冲区以及磁盘之间进行传输;控制连接模块:控制数据在所述数据连接模块之间的传输,并根据所述数据连接模块的状态选择是否调用连接资源清理模块;线程池模块:通过线程池模块实现套接字数据缓冲区与磁盘之间的数据读写;连接资源清理模块:对数据连接模块和控制连接模块的数据进行销毁,释放数据连接模块和控制连接模块的数据占用的内存空间。本发明中,当读/写任务的操作不能满足时,进程不再因某个套接字的读写而被阻塞。



CN 113965561 B

1. 一种基于异步事件驱动的机载文件传输系统,其特征在于,包括控制连接模块、数据连接模块、线程池模块和连接资源清理模块,其中,

数据连接模块:数据通过所述数据连接模块在网络、套接字数据缓冲区以及磁盘之间进行传输;

控制连接模块:控制数据在所述数据连接模块之间的传输,并根据所述数据连接模块的状态选择是否调用连接资源清理模块;

线程池模块:通过线程池模块实现套接字数据缓冲区与磁盘之间的数据读写;

连接资源清理模块:对数据连接模块和控制连接模块的数据进行销毁,释放数据连接模块和控制连接模块的数据占用的内存空间;

所述数据连接模块包括处理数据连接请求模块、读取数据模块、发送数据模块以及线程池返回结果模块,当客户端发出数据连接请求时,触发EPOLL产生数据读/写事件,EPOLL用于监听数据连接套接字的读/写状态;所述数据连接套接字存放在EPOLL的套接字数据缓冲区中;

所述处理数据连接请求模块接收并处理客户端发送的数据连接请求,并与所述客户端建立数据连接;

所述读取数据模块用于读取EPOLL中的数据连接套接字的数据,并将所述数据放入线程池的任务队列模块中,线程池模块从任务队列读取数据并写磁盘,任务完成后,线程池模块通过管道向数据连接模块的线程池返回结果模块写回任务完成结果;

当EPOLL监听到的数据连接套接字的状态为可写时,所述发送数据模块通过所述线程池模块读取的磁盘数据,并将读取到的磁盘数据存入套接字数据缓冲区中,然后通过网络发送给客户端,同时,将读取到的磁盘数据传输至线程池的任务队列模块中,以便反馈至对应的工作线程中;

所述线程池返回结果模块根据线程池模块反馈的返回结果判断上传任务是否完成或者下载过程是否出错,若返回结果为成功,则读取数据模块和/或发送数据模块继续前操作,否则,所述读取数据模块和/或所述发送数据模块断开与EPOLL的连接并注销EPOLL事件;

所述线程池返回结果模块的处理流程包括:

S4.1、获取线程池返回任务的返回结果;

S4.2、通过所述返回结果判断读/写磁盘是否出错,若是,则跳转到S4.3,否则,跳转到S4.5;

S4.3、判断任务队伍模块中的任务类型是否为写磁盘,若是,取消任务队伍模块中上传文件的其他任务,关闭该上传文件的数据连接套接字并删除该上传文件,然后通过控制连接模块进行报错,接着关闭读取数据模块与EPOLL之间的数据连接,然后跳转至S4.8,若任务类型不为写磁盘,跳转至S4.4;

S4.4、关闭下载文件的数据连接套接字,然后通过控制连接模块进行报错,接着关闭发送数据模块与EPOLL之间的数据连接,然后跳转至S4.8;

S4.5、判断任务队伍模块中的任务类型是否为写磁盘,若是,跳转至S4.6,否则,跳转至S4.7;

S4.6、判断上传文件是否传输结束,若是,改变控制连接状态机的连接状态,然后跳转

至S4.8,否则,直接跳转至S4.8;

S4.7、判断是否读到下载文件的结束符,若是,改变控制连接状态机的连接状态,然后跳转至S4.8,否则,将从磁盘读出的数据写入数据连接缓冲区中,然后跳转至S4.8;

S4.8、结束。

2.根据权利要求1所述的一种基于异步事件驱动的机载文件传输系统,其特征在于,所述控制连接模块包括处理控制连接请求模块、命令解析模块、控制连接状态机以及命令操作接口,当客户端发出控制连接请求时,触发EPOLL产生控制读/写事件,EPOLL用于监听控制连接套接字的读/写状态,所述控制连接套接字存放在EPOLL的套接字数据缓冲区中;

所述处理控制连接请求模块用于接收客户端发出的控制连接请求;

所述命令解析模块用于读取并解析EPOLL发出的控制连接套接字数据,并将解析后的数据发送至控制连接状态机;

所述控制连接状态机判断解析后的数据在当前状态下是否符合规则,同时,EPOLL对所述控制连接状态机的状态进行监听,若符合规则,则调用相应的命令操作接口,当EPOLL监测到的控制连接套接字的状态为写状态时,修改所述控制连接状态机的状态,然后将解析后的数据发送给客户端。

3.根据权利要求2所述的一种基于异步事件驱动的机载文件传输系统,其特征在于,所述线程池模块包括任务指针、任务队列模块、线程任务处理接口;其中,

所述任务队列模块存放所述读取数据模块的写磁盘任务、发送数据模块的读磁盘任务;

所述线程任务处理接口用于连接任务队列和线程池模块中的工作线程;

所述任务指针为磁盘数据,所述任务指针通过所述线程任务处理接口到达管道中,通过所述管道传输至所述线程池返回结果模块中。

4.根据权利要求1所述的一种基于异步事件驱动的机载文件传输系统,其特征在于,所述线程池返回结果模块根据线程池模块反馈的返回结果判断任务是否完成具体为:

判断从数据连接套接字读取数据并写入磁盘的上传任务是否完成,若上传任务完成,读取数据失败,返回结果为失败,否则,返回结果为成功,读取数据成功,继续从数据连接套接字中读取数据并放入线程池的任务队列模块中;

判断从磁盘读取磁盘数据的下载过程是否发生错误,若下载过程未出错,返回结果为成功,将磁盘数据发送至所述发送数据模块,并通过所述发送数据模块将磁盘数据存储在EPOLL的套接字数据缓冲区,并通过网络发送给客户端,然后继续从磁盘读取磁盘数据,否则,返回结果为失败,同时,通过所述发送数据模块将磁盘数据发送至线程池的任务队列模块中,以便反馈至对应的工作线程中。

5.根据权利要求3所述的一种基于异步事件驱动的机载文件传输系统,其特征在于,在所述线程池模块包括多个工作线程,每个所述工作线程通过所述线程任务处理接口分别与所述管道和任务队列模块连接。

6.根据权利要求2所述的一种基于异步事件驱动的机载文件传输系统,其特征在于,所述控制连接模块的工作流程包括:

S1.1、客户端发出控制连接请求;

S1.2、所述处理控制连接请求模块接收客户端发出的控制连接请求并与客户端建立联

系；

S1.3、触发EPOLL产生控制读/写事件，产生控制连接套接字数据；

S1.4、所述命令解析模块读取并解析EPOLL发出的控制连接套接字数据，并将解析后的数据发送至控制连接状态机；

S1.5、所述控制连接状态机判断解析后的数据在当前状态下是否符合规则，同时，EPOLL对所述控制连接状态机的状态进行监听，若符合规则，则调用相应的命令操作接口，否则，系统报错；

S1.6、根据EPOLL对控制连接套接字的状态监测的结果，将所述控制连接状态机的状态修改为相应的状态。

7. 根据权利要求1所述的一种基于异步事件驱动的机载文件传输系统，其特征在于，所述读取数据模块从客户端读取数据的流程包括：

S2.1、客户端发出上传文件请求，触发EPOLL读事件的产生；

S2.2、调用EPOLL\_wait等待上传文件的数据连接套接字就绪；

S2.3、所述读取数据模块从所述数据连接套接字中读取数据；

S2.4、判断是否读到文件结束符，若是，则到S2.5，否则，添加写磁盘任务至线程池的任务队列模块中，然后返回到S2.2；

S2.5、注销产生的EPOLL读事件；

S2.6、结束。

8. 根据权利要求7所述的一种基于异步事件驱动的机载文件传输系统，其特征在于，所述发送数据模块从磁盘中读取磁盘数据的流程包括：

S3.1、客户端发出下载文件请求，触发EPOLL写事件的产生；

S3.2、调用EPOLL\_wait等待下载文件的数据连接套接字就绪；

S3.3、判断所述发送数据模块是否已读到磁盘数据，若是，则向网络发送数据；否则，跳转至S3.5；

S3.4、判断磁盘中的文件数据是否传输结束，若否，跳转至S3.6；否则，注销EPOLL写事件，然后跳转至S3.7；

S3.5、判断读磁盘任务是否已添加至线程池的任务队伍模块中，若是，跳转至S3.2；

S3.6、向任务队伍模块中添加读磁盘任务；

S3.7、结束。

## 一种基于异步事件驱动的机载文件传输系统

### 技术领域

[0001] 本发明涉及文件传输领域,具体涉及一种基于异步事件驱动的机载文件传输系统。

### 背景技术

[0002] 目前机载系统大多数采用的文件传输系统是基于FTP服务的,通过FTP服务,其它设备可以从授权的机载服务器上获取所需的文件,也可以把本地文件传输到其它设备上。这就为机载设备之间双向传输文件提供了一种有效的手段。FTP可运载任何数据,从原始的ASCII文本,SGML,到可执行代码、图像、声音、视频动画等。目前的机载文件传输所使用的FTP服务器大多采用Linux环境下的FTP,像VSTP、ProFTP等,大多均采用多进程同步I/O机制。当连接请求上升到上百甚至上千时,服务器就需要创建连接个数匹配的多个进程来服务客户请求。多个进程间频繁的上下文切换和调度就需要消耗极大的操作系统开销(比如CPU、内存等)。这样机载服务器自身所能获得的系统资源就很少,导致处理客户端服务请求的能力急剧降低。而且随着智能客舱的不断发展,机载服务器需要支持的连接数将会是上千甚至更多,十分影响服务器在网络上的响应速度。

### 发明内容

[0003] 本发明提供一种基于异步事件驱动的文件传输系统,摒弃了传统的多进程同步阻塞I/O机制,引入了非阻塞I/O与事件驱动机制EPOLL,解决了机载系统的阻塞问题,加快服务器在网络上的响应速度。

[0004] 本发明通过下述技术方案实现:

[0005] 一种基于异步事件驱动的机载文件传输系统,包括控制连接模块、数据连接模块、线程池模块和连接资源清理模块,其中,

[0006] 数据连接模块:数据通过所述数据连接模块在网络、套接字数据缓冲区以及磁盘之间进行传输;

[0007] 控制连接模块:控制数据在所述数据连接模块之间的传输,并根据所述数据连接模块的状态选择是否调用连接资源清理模块;

[0008] 线程池模块:通过线程池模块实现套接字数据缓冲区与磁盘之间的数据读写;

[0009] 连接资源清理模块:对数据连接模块和控制连接模块的数据进行销毁,释放数据连接模块和控制连接模块的数据占用的内存空间。

[0010] 作为优化,所述数据连接模块包括处理数据连接请求模块、读取数据模块、发送数据模块以及线程池返回结果模块,当客户端发出数据连接请求时,触发EPOLL产生数据读/写事件,EPOLL用于监听数据连接套接字的读/写状态;所述数据连接套接字存放在EPOLL的套接字数据缓冲区中;

[0011] 所述处理数据连接请求模块接收并处理客户端发送的数据连接请求,并与所述客户端建立数据连接;

[0012] 所述读取数据模块用于读取EPOLL中的数据连接套接字的数据,并将所述数据放入线程池的任务队列模块中;

[0013] 当EPOLL监听到的数据连接套接字的状态为可写时,所述发送数据模块通过所述线程池模块读取的磁盘数据,并将读取到的磁盘数据存入套接字数据缓冲区中,然后通过网络发送给客户端,同时,将读取到的磁盘数据传输至线程池的任务队列模块中,以便反馈至对应的工作线程中;

[0014] 所述线程池返回结果模块根据线程池模块反馈的返回结果判断上传任务是否完成或者下载过程是否出错,若返回结果为成功,则读取数据模块和/或发送数据模块继续前操作,否则,所述读取数据模块和/或所述发送数据模块断开与EPOLL的连接并注销EPOLL事件。

[0015] 作为优化,所述控制连接模块包括处理控制连接请求模块、命令解析模块、控制连接状态机以及命令操作接口,当客户端发出控制连接请求时,触发EPOLL产生控制读/写事件,EPOLL用于监听控制连接套接字的读/写状态,所述控制连接套接字存放在EPOLL的套接字数据缓冲区中;

[0016] 所述处理控制连接请求模块用于接收客户端发出的控制连接请求;

[0017] 所述命令解析模块用于读取并解析EPOLL发出的控制连接套接字数据,并将解析后的数据发送至控制连接状态机;

[0018] 所述控制连接状态机判断解析后的数据在当前状态下是否符合规则,同时,EPOLL对所述控制连接状态机的状态进行监听,若符合规则,则调用相应的命令操作接口,当EPOLL监测到的控制连接套接字的状态为写状态时,修改所述控制连接状态机的状态,然后将解析后的数据发送给客户端。

[0019] 作为优化,所述线程池模块包括任务指针、任务队列模块、线程任务处理接口;其中,

[0020] 所述任务队列模块存放所述读取数据模块的写磁盘任务、发送数据模块的读磁盘任务;

[0021] 所述线程任务处理接口用于连接任务队列和线程池模块中的工作线程;

[0022] 所述任务指针为磁盘数据,所述任务指针通过所述线程任务处理接口到达管道中,通过所述管道传输至所述线程池返回结果模块中。

[0023] 作为优化,所述线程池返回结果模块根据线程池模块反馈的返回结果判断任务是否完成具体为:

[0024] 判断从数据连接套接字读取数据并写入磁盘的上传任务是否完成,若上传任务完成,读取数据失败,返回结果为失败,否则,返回结果为成功,读取数据成功,继续从数据连接套接字中读取数据并放入线程池的任务队列模块中;

[0025] 判断从磁盘读取磁盘数据的下载过程是否发生错误,若下载过程未出错,返回结果为成功,将磁盘数据发送至所述发送数据模块,并通过所述发送数据模块将磁盘数据存储在EPOLL的套接字数据缓冲区,并通过网络发送给客户端,然后继续从磁盘读取磁盘数据,否则,返回结果为失败,同时,通过所述发送数据模块将磁盘数据发送至线程池的任务队列模块中,以便反馈至对应的工作线程中。

[0026] 作为优化,在所述线程池模块包括多个工作线程,每个所述工作线程通过所述线

程任务处理接口分别与所述管道和任务队列模块连接。

[0027] 作为优化,所述控制连接模块的工作流程包括:

[0028] S1.1、客户端发出控制连接请求;

[0029] S1.2、所述处理控制连接请求模块接收客户端发出的控制连接请求并与客户端建立联系;

[0030] S1.3、触发EPOLL产生控制读/写事件,产生控制连接套接字数据;

[0031] S1.4、所述命令解析模块读取并解析EPOLL发出的控制连接套接字数据,并将解析后的数据发送至控制连接状态机;

[0032] S1.5、所述控制连接状态机判断解析后的数据在当前状态下是否符合规则,同时,EPOLL对所述控制连接状态机的状态进行监听,若符合规则,则调用相应的命令操作接口,否则,系统报错;

[0033] S1.6、根据EPOLL对控制连接套接字的状态监测的结果,将所述控制连接状态机的状态修改为相应的状态。

[0034] 作为优化,所述读取数据模块从客户端读取数据的流程包括:

[0035] S2.1、客户端发出上传文件请求,触发EPOLL读事件的产生;

[0036] S2.2、调用EPOLL\_wait等待上传文件的数据连接套接字就绪;

[0037] S2.3、所述读取数据模块从所述数据连接套接字中读取数据;

[0038] S2.4、判断是否读到文件结束符,若是,则到S2.5,否则,添加写磁盘任务至线程池的任务队列模块中,然后返回到S2.2;

[0039] S2.5、注销产生的EPOLL读事件;

[0040] S2.6、结束。

[0041] 作为优化,所述发送数据模块从磁盘中读取磁盘数据的流程包括:

[0042] S3.1、客户端发出下载文件请求,触发EPOLL写事件的产生;

[0043] S3.2、调用EPOLL\_wait等待下载文件的数据连接套接字就绪;

[0044] S3.3、判断所述发送数据模块是否已读到磁盘数据,若是,则向网络发送数据;否则,跳转至S3.5;

[0045] S3.4、判断磁盘中的文件数据是否传输结束,若否,跳转至S3.6;否则,注销EPOLL写事件,然后跳转至S3.7;

[0046] S3.5、判断读磁盘任务是否已添加至线程池的任务队伍模块中,若是,跳转至S3.2;

[0047] S3.6、向任务队伍模块中添加读磁盘任务;

[0048] S3.7、结束。

[0049] 作为优化,所述线程池返回结果模块的处理流程包括:

[0050] S4.1、获取线程池返回任务的返回结果;

[0051] S4.2、通过所述返回结果判断读/写磁盘是否出错,若是,则跳转到S4.3,否则,跳转到S4.5;

[0052] S4.3、判断任务队伍模块中的任务类型是否为写磁盘,若是,取消任务队伍模块中该上传文件的其他任务,关闭该上传文件的数据连接套接字并删除该上传文件,然后通过控制连接模块进行报错,接着关闭读取数据模块与EPOLL之间的数据连接,然后跳转至

S4.8,若任务类型不为写磁盘,跳转至S4.4;

[0053] S4.4、关闭下载文件的数据连接套接字,然后通过控制连接模块进行报错,接着关闭发送数据模块与EPOLL之间的数据连接,然后跳转至S4.8;

[0054] S4.5、判断任务队伍模块中的任务类型是否为写磁盘,若是,跳转至S4.6,否则,跳转至S4.7;

[0055] S4.6、判断上传文件是否传输结束,若是,改变控制连接状态机的连接状态,然后跳转至S4.8,否则,直接跳转至S4.8;

[0056] S4.7、判断是否读到下载文件的结束符,若是,改变控制连接状态机的连接状态,然后跳转至S4.8,否则,将从磁盘读出的数据写入数据连接缓冲区中,然后跳转至S4.8;

[0057] S4.8、结束。

[0058] 本发明与现有技术相比,具有如下的优点和有益效果:

[0059] 本发明的套接字缺省是非阻塞的,当读/写任务的操作不能满足时,将立即返回EWOULDBLOCK故障报错,进程不再因某个套接字的读写而被阻塞。

## 附图说明

[0060] 为了更清楚地说明本发明示例性实施方式的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本发明的某些实施例,因此不应被看作是对范围的限定,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他相关的附图。在附图中:

[0061] 图1为本发明所述的一种基于异步事件驱动的机载文件传输系统的系统架构图;

[0062] 图2为图1中上传文件的过程流程图;

[0063] 图3为图1中下载文件的过程流程图;

[0064] 图4为图1中控制连接模块的过程流程图;

[0065] 图5为图1中线程池返回结果模块的处理流程。

## 实施方式

[0066] 为使本发明的目的、技术方案和优点更加清楚明白,下面结合实施例和附图,对本发明作进一步的详细说明,本发明的示意性实施方式及其说明仅用于解释本发明,并不作为对本发明的限定。

[0067] 一种基于异步事件驱动的机载文件传输系统,包括控制连接模块、数据连接模块、线程池模块和连接资源清理模块,其中,

[0068] 数据连接模块(DATA\_CONN):数据在网络与连接套接字数据缓冲区之间数据的传输,具体的,数据通过所述数据连接模块在网络、套接字数据缓冲区以及磁盘之间进行传输;

[0069] 控制连接模块(CTRL\_CONN):负责控制连接各项工作。包括命令接收并执行相应动作,发送应答等任务,具体的,控制数据在所述数据连接模块之间的传输,并根据所述数据连接模块的状态选择是否调用连接资源清理模块;

[0070] 线程池模块:通过线程池模块实现套接字数据缓冲区与磁盘之间的数据读写;

[0071] 连接资源清理模块:对数据连接模块和控制连接模块的数据进行销毁,释放数据



连接模块和控制连接模块的数据占用的内存空间,防止操作系统内存泄露。

[0072] 在数据传输过程中,控制连接模块与数据传输(连接)模块间存在着交互。数据连接模块需要将自己的状态信息通知控制连接模块。控制连接模块根据数据连接模块的状态发回相应的应答并改变控制连接当前的状态。当数据传输终止时,控制连接模块调用清理模块释放数据连接占用的资源。

[0073] 数据连接模块同线程池模块也存在交互,数据连接模块需要将通过网络读取到的数据交付给线程池,由线程池模块写入磁盘完成持久化。同时,也需要线程池模块读取磁盘数据。具体的实现方式是数据连接模块将读写磁盘任务放入任务队列,线程池模块监听任务队列发现队列中有任务到达时,线程池模块从任务队列获取任务并下发给线程池中的工作线程处理。处理完成后通过管道返回任务的处理信息。比如读取是否成功,读取到的数据等。

[0074] 本实施例中,所述数据连接模块包括处理数据连接请求模块、读取数据模块、发送数据模块以及线程池返回结果模块,文件传输开始前,文件传输系统在特定(指定)端口接收并处理客户端发送的数据连接请求并同客户端建立数据连接。

[0075] 具体的,客户端上传文件时触发EPOLL读事件产生。所述处理数据连接请求模块接收并处理客户端发送的数据连接请求,并与所述客户端建立数据连接;EPOLL用于监听数据连接套接字的读/写状态,所述数据连接套接字存放在EPOLL的套接字数据缓冲区中。

[0076] 当EPOLL监听到的数据连接套接字的状态为可写时,所述发送数据模块通过所述线程池模块读取的磁盘数据,并将读取到的磁盘数据存入套接字数据缓冲区中,然后发送给网络,同时,将读取到的磁盘数据传输至线程池的任务队列模块中,以便反馈至对应的工作线程中。

[0077] 所述线程池返回结果模块根据线程池模块反馈的返回结果判断任务是否完成,若返回结果为成功,则读取数据模块和/或发送数据模块继续前操作,否则,所述读取数据模块和/或所述发送数据模块断开与EPOLL的连接并注销EPOLL事件。

[0078] 上传文件时,文件传输系统中的读取数据模块从数据连接套接字中读取数据,读取到的数据放入线程池的任务队列。线程池模块从任务队列读取数据并写磁盘,任务完成后,线程池模块通过管道向数据连接模块的线程池返回结果模块写回任务完成结果。数据连接模块中的线程池返回结果模块根据返回结果判断数据读/写是否成功,即线程池返回结果模块判断从数据连接套接字读取数据并写入磁盘的任务是否完成,若任务完成,读取数据失败,返回结果为失败,否则,返回结果为成功,读取数据成功,继续从数据连接套接字中读取数据并放入线程池的任务队列模块中;如果发现返回结果为失败,则关闭数据连接并注销EPOLL读事件。上传文件的流程图如图2,具体步骤如下:

[0079] S2.1、客户端发出上传文件请求,触发EPOLL读事件的产生;

[0080] S2.2、调用EPOLL\_wait等待上传文件的数据连接套接字就绪;

[0081] S2.3、所述读取数据模块从所述数据连接套接字中读取数据;

[0082] S2.4、判断是否读到文件结束符,若是,则到S2.5,否则,添加写磁盘任务至线程池的任务队列模块中,然后返回到S2.2;

[0083] S2.5、注销产生的EPOLL读事件;

[0084] S2.6、结束。

[0085] 下载文件时,传输系统先把读磁盘任务放入任务队列模块中,线程池模块完成读磁盘任务后,通过管道向数据连接模块的线程池返回结果模块返回任务处理结果。数据连接模块的线程池返回结果模块根据返回结果判断是否发生错误(即是否成功),也就是,判断从磁盘读取磁盘数据的下载过程是否发生错误,若下载过程未出错,返回结果为成功,将磁盘数据发送至所述发送数据模块,并通过所述发送数据模块将磁盘数据存储于EPOLL的套接字数据缓冲区,并通过网络发送给客户端,然后继续从磁盘读取磁盘数据,否则,返回结果为失败。同时,通过所述发送数据模块将磁盘数据发送至线程池的任务队列模块中,以便反馈至对应的工作线程中。下载文件的流程图如图3,具体步骤如下:

[0086] S3.1、客户端发出下载文件请求,触发EPOLL写事件的产生;

[0087] S3.2、调用EPOLL\_wait等待下载文件的数据连接套接字就绪;

[0088] S3.3、判断所述发送数据模块是否已读到磁盘数据,若是,则向网络发送数据;否则,跳转至S3.5;

[0089] S3.4、判断磁盘中的文件数据是否传输结束,若否,跳转至S3.6;否则,注销EPOLL写事件,然后跳转至S3.7;

[0090] S3.5、判断读磁盘任务是否已添加至线程池的任务队伍模块中,若是,跳转至S3.2;

[0091] S3.6、向任务队伍模块中添加读磁盘任务;

[0092] S3.7、结束。

[0093] 控制连接模块包含以下几个部分:处理控制连接请求模块、命令解析模块、控制连接状态机以及命令操作接口。该模块在特定(指定)端口上等待客户端发起的控制连接请求,连接建立后,需要对用户发送的命令(控制连接请求)进行解析,得到命令部分和参数部分。将解析出的命令放入控制连接状态机,判断该命令在当前状态下是否符合规则,如果符合的话调用相应的命令操作接口执行命令,改变当前状态值。

[0094] 具体的,当客户端发出控制连接请求时,触发EPOLL产生控制读/写事件,EPOLL用于监听控制连接套接字的读/写状态,所述控制连接套接字存放在EPOLL的套接字数据缓冲区中;

[0095] 所述处理控制连接请求模块用于接收客户端发出的控制连接请求;

[0096] 所述命令解析模块用于读取并解析EPOLL发出的控制连接套接字数据,并将解析后的数据发送至控制连接状态机;

[0097] 所述控制连接状态机判断解析后的数据在当前状态下是否符合规则,同时,EPOLL对所述控制连接状态机的状态进行监听,若符合规则,则调用相应的命令操作接口,当EPOLL监测到的控制连接套接字的状态为写状态时,修改所述控制连接状态机的状态,然后将解析后的数据发送给客户端。控制连接模块流程图如图4所示,具体的流程步骤如下:

[0098] S1.1、客户端发出控制连接请求;

[0099] S1.2、所述处理控制连接请求模块接收客户端发出的控制连接请求并与客户端建立联系;

[0100] S1.3、触发EPOLL产生控制读/写事件,产生控制连接套接字数据;

[0101] S1.4、所述命令解析模块读取并解析EPOLL发出的控制连接套接字数据,并将解析后的数据发送至控制连接状态机;

[0102] S1.5、所述控制连接状态机判断解析后的数据在当前状态下是否符合规则,同时,EPOLL对所述控制连接状态机的状态进行监听,若符合规则,则调用相应的命令操作接口,否则,系统报错;

[0103] S1.6、根据EPOLL对控制连接套接字的状态监测的结果,将所述控制连接状态机的状态修改为相应的状态。

[0104] 所述线程池模块包括任务指针、任务队列模块、线程任务处理接口;其中,

[0105] 所述任务队列模块存放所述读取数据模块的写磁盘任务、发送数据模块的读磁盘任务;

[0106] 所述线程任务处理接口用于连接任务队列和线程池模块中的工作线程;

[0107] 所述任务指针为磁盘数据,所述任务指针通过所述线程任务处理接口到达管道中,通过所述管道传输至所述线程池返回结果模块中。在所述线程池模块包括多个工作线程,每个所述工作线程通过所述线程任务处理接口分别与所述管道和任务队列模块连接。

[0108] 具体的,线程池模块主要功能有线程池创建、线程池初始化、任务队列监控、任务处理、线程池销毁。线程池的创建主要完成线程池数据结构的初始化,线程池初始化主要完成对工作线程的创建。任务处理主要负责对线程池获取到的任务进行处理;任务队列监控主要负责监控任务队列,当有任务到达时唤醒工作线程对新任务进行执行;线程池销毁主要是负责进程退出时对线程池占用的系统资源进行释放。线程池模块预先创建多个工作线程,线程池模块的结构主要包括工作线程、任务队列、线程任务处理接口、任务指针部分。其中工作线程是一个可循环执行任务的线程,没有任务时处于等待状态,新任务(读/写磁盘)到达时被唤醒。任务队列提供了一种缓冲机制来临时存放待处理的任务。线程任务处理接口是需要每个任务必须实现的接口,工作线程通过该接口调度任务的执行。

[0109] 连接资源清理模块:由于系统运行中需要创建和销毁控制连接和数据连接,因此需要对这两种连接所占用的内存资源进行管理。因此该模块主要负责当数据连接和控制连接关闭后,释放这两种连接在使用过程中所申请的内存资源。

[0110] 线程池返回结果处理流程的流程图如图5,具体步骤如下:

[0111] S4.1、获取线程池返回任务的返回结果;

[0112] S4.2、通过所述返回结果判断读/写磁盘是否出错,若是,则跳转到S4.3,否则,跳转到S4.5;

[0113] S4.3、判断任务队伍模块中的任务类型是否为写磁盘,若是,取消任务队伍模块中该上传文件的其他任务,关闭该上传文件的数据连接套接字并删除该上传文件,然后通过控制连接模块进行报错,接着关闭读取数据模块与EPOLL之间的数据连接,然后跳转至S4.8,若任务类型不为写磁盘,跳转至S4.4;

[0114] S4.4、关闭下载文件的数据连接套接字,然后通过控制连接模块进行报错,接着关闭发送数据模块与EPOLL之间的数据连接,然后跳转至S4.8;

[0115] S4.5、判断任务队伍模块中的任务类型是否为写磁盘,若是,跳转至S4.6,否则,跳转至S4.7;

[0116] S4.6、判断上传文件是否传输结束,若是,改变控制连接状态机的连接状态,使得EPOLL读事件被注销,然后跳转至S4.8,否则,直接跳转至S4.8;

[0117] S4.7、判断是否读到下载文件的结束符,若是,改变控制连接状态机的连接状态,

使得EPOLL写事件被注销,然后跳转至S4.8,否则,将从磁盘读出的数据写入数据连接缓冲区中,然后跳转至S4.8;

[0118] S4.8、结束。

[0119] 上述的判断是否符合规则中的规则均为文件传输协议的规则,注销通过连接资源清理模块实现。

[0120] 本发明采用事件驱动机制和非阻塞I/O方式。上述的判断是否符合规则中的规则均为文件传输协议的规则,注销通过连接资源清理模块实现。

[0121] 采用事件驱动机制:

[0122] 传统的处理网络I/O的技术分为多进程、多线程同步阻塞I/O或者是SELECT.POLL的单线程事件驱动I/O等方式。其中多线程和多进程同步阻塞网络I/O技术具有模型直观,使用方便等优点,但是当处理高并发的网络连接时,因为存在fork(分叉)和上下文切换操作产生较大的系统开销,内存开销也较大不能满足服务器性能要求。SELECT和POLL的单线程驱动I/O技术虽然没有大量的上下文切换,但应用程序需要不断地向内核写入套接字集或POLLED结构,将会导致过多的用户控件和内核空间拷贝,另外内核必须扫描套接字集中所有套接字来探测就绪的套接字的就绪状态,这些操作也会增加系统开销。由于EPOLL在处理高并发、高负载数据通信中具有传统网络I/O不具有的优点,比如直接返回就绪的套接字字符集,不需要轮询所有的监听套接字。当有新的连接请求到达时,系统立即创建对应的网络套接字并将它们注册为EPOLL的读事件或者写事件,由EPOLL监听网络套接字的活跃情况。根据获取EPOLL的返回事件集中的事件处理网络套接字的数据收发。这样就避免了只要有新的客户端连接就需要创建一个新执行体的情况。

[0123] 采用非阻塞I/O:

[0124] 套接字缺省是阻塞的,当发生一个不能立即完成的套接字调用时,进程投入睡眠直到操作完成。例如,读套接字时,如果套接字相关的接收缓冲区为空,进程会被阻塞,等待数据到达接收缓冲区。写套接字时,如果套接字相关的发送缓冲区满了进程也会被阻塞,直到缓冲区数据清空。这些I/O操作引发的阻塞会使得服务进程进入睡眠,无法进行其它操作影响了服务器性能。

[0125] 对于非阻塞套接字,如果操作(比如read或者write)不能满足将立即返回EWOULDBLOCK,进程不再因某个套接字的读写阻塞。

[0126] 在本发明的传输系统的主框架下,采用EPOLL异步驱动监听各个模块的请求。EPOLL异步驱动的实现机制是把与通信实体交互的通信单元抽象成Agent类。每个通信实体对应于一个Agent实例。EPOLL异步事件通知机制可以将某套接字与epoll\_event结构体下的ptr指针成员进行绑定,当这个套接字或者文件描述符上有I/O事件到达时,EPOLL会返回这个套接字所绑定的地址指针,如果把这个地址指针指向这个套接字或者文件描述符对应的Agent实例,这样就可以让EPOLL返回Agent实例的地址,然后根据I/O事件的不同调用Agent里面的对应函数处理与通信实体间的交互。Agent类处理的交互一般包括读写事件处理,以及Agent的启动和停止。通过这种实现方式加快了服务器在网络上的响应速度。比如在该方案中,系统初始化时,在特定端口创建监听套接字,并为它创建Agent的派生类实例并注册读事件,循环调用EPOLL的接口epoll\_wait等待客户端控制连接的请求。

[0127] 其次,通过创建线程池以及调用线程池中的工作线程执行文件的磁盘I/O,避免了

某个客户端磁盘读写任务阻塞其它客户端网络读写请求,保证该文件传输系统的并发性和总体的数据传输速度。

[0128] 以上所述的具体实施方式,对本发明的目的、技术方案和有益效果进行了进一步详细说明,所应理解的是,以上所述仅为本发明的具体实施方式而已,并不用于限定本发明的保护范围,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

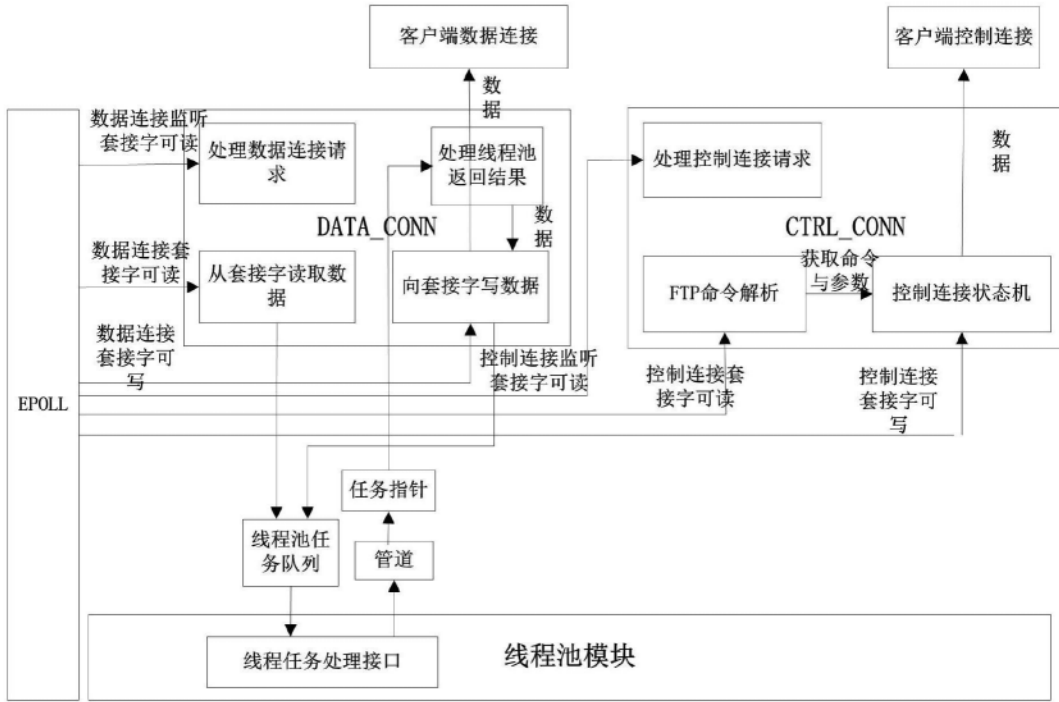


图1

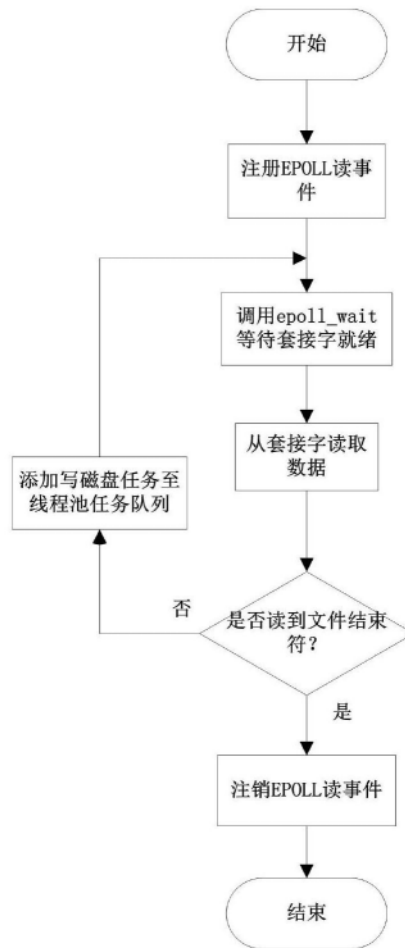


图2

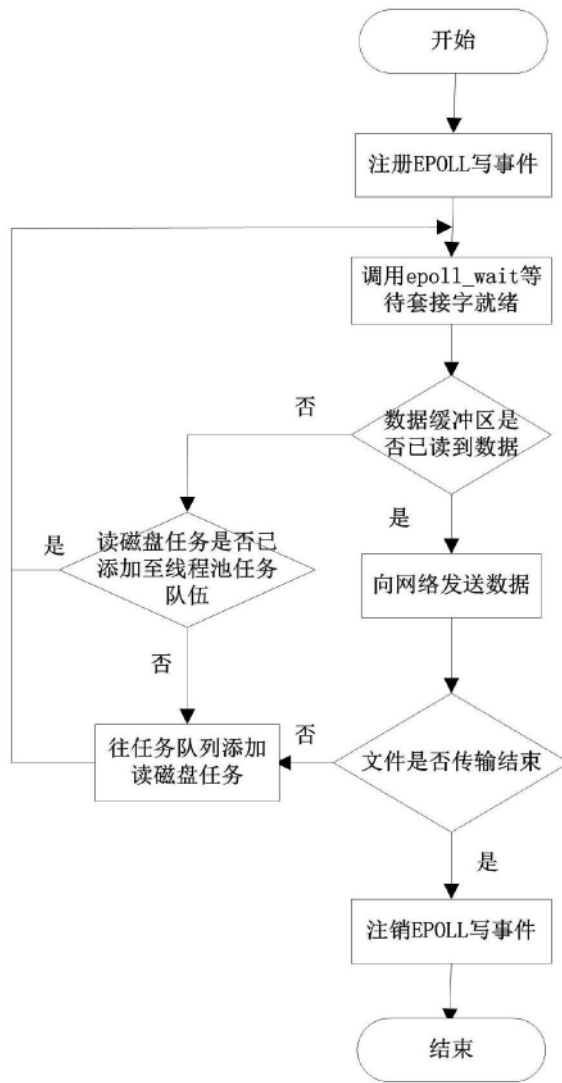


图3



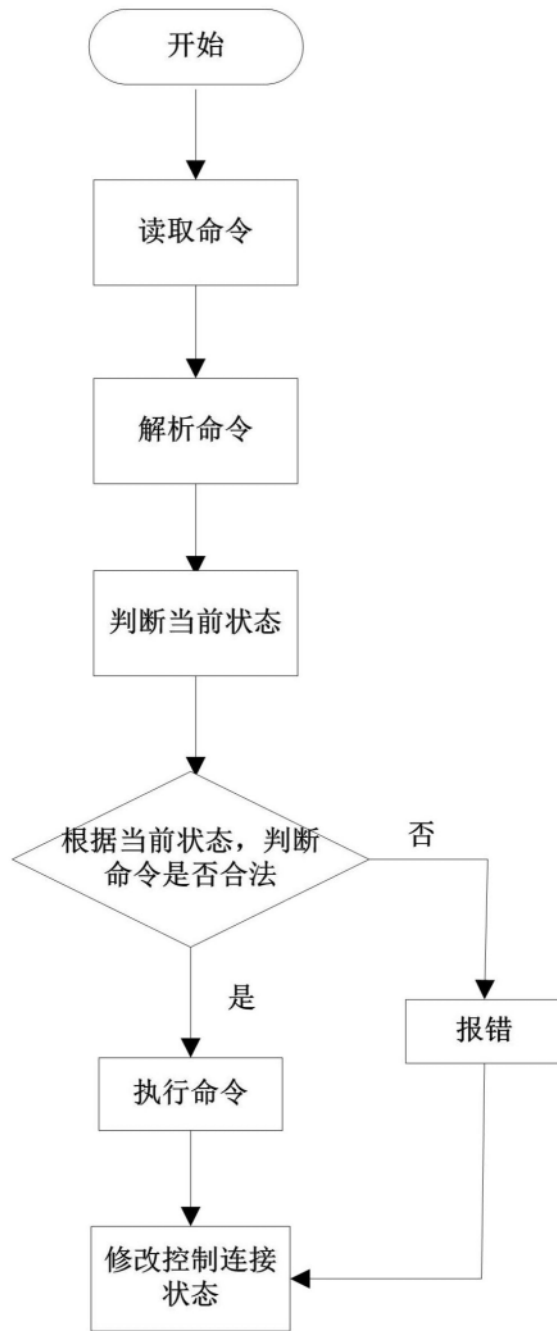


图4

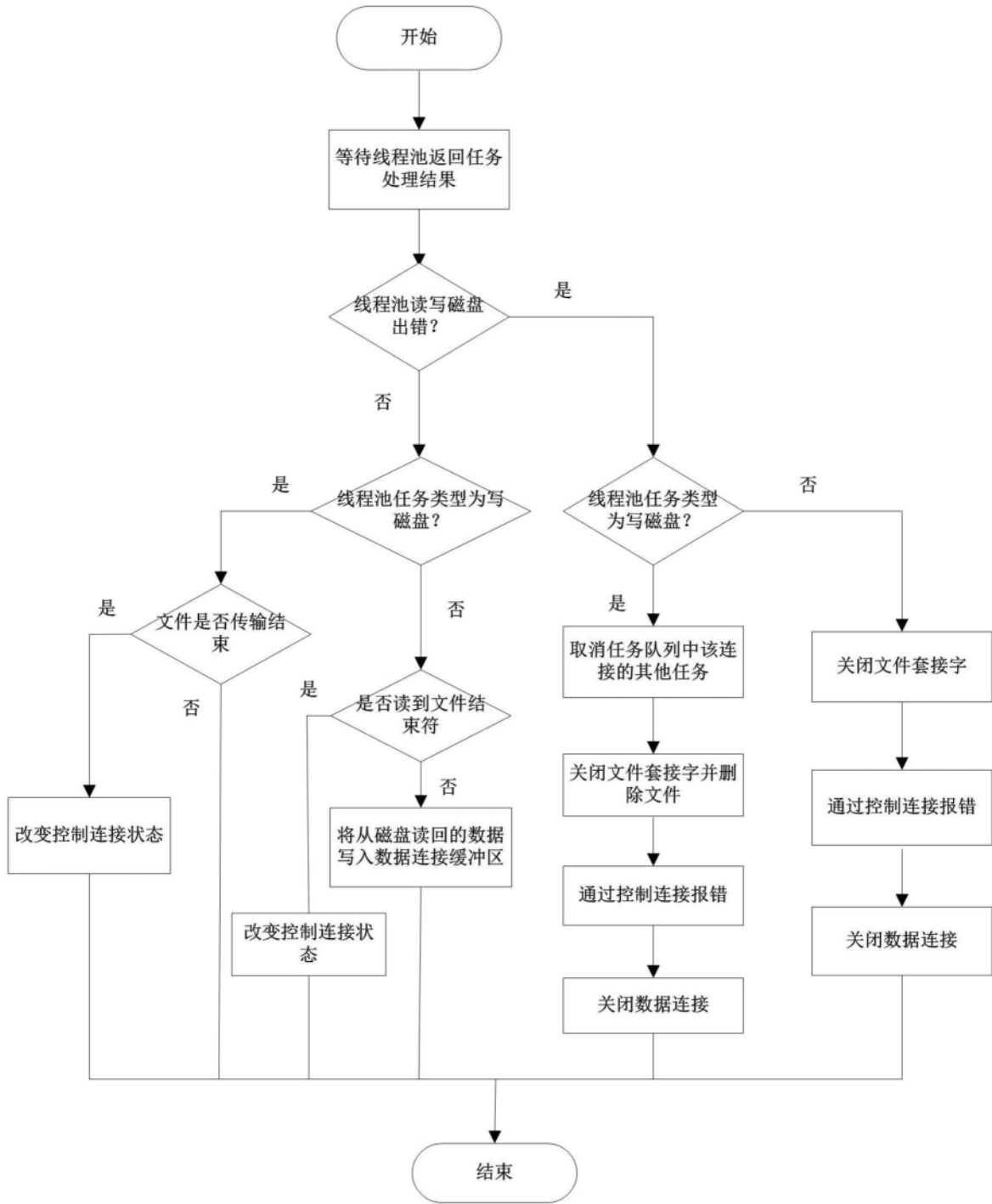


图5