

网盘-技术手册



小组成员 1 _____ 1953186 龚乙骁

小组成员 2 _____ 1951330 魏鹏远

专 业 _____ 信息安全

授课老师 _____ 沈坚

一、数据库设计

数据库一共 5 张表



1. dir: 用户的目录结构

did	name	f_did	username
880	/	-1	wpy4
900	/camera_video	880	wpy4
901	/camera_video/1951330	900	wpy4
902	/camera_video/1951330/test	901	wpy4
903	/camera_video/23	900	wpy4
904	/camera video/23/342	903	wpy4

Field	Type	Comment
did	int(11) NOT NULL	
name	varchar(500) NOT NULL	
f_did	int(11) NOT NULL	
username	varchar(50) NOT NULL	

各个字段的含义:

did: 目录 id

name: 目录名, 包含整个路径比如 “/etc/home”

f_did: 父目录的 id

username: 目录属于的用户名

2. user: 用户信息

uid	name	passwd	token	root_did
12	wpy4	eeac73e94c491334713e5c96bc6b028f	Pg3sluR0M54z4b8u45L7vPq725nAq431	840
*	(Auto)	(NULL)	(NULL)	(NULL)

Field	Type
uid	int(10) unsigned NOT NULL
name	varchar(50) NOT NULL
passwd	varchar(32) NOT NULL
token	varchar(32) NULL
root_did	int(11) NOT NULL

各个字段的含义:

uid: 用户 id

name: 用户名

passwd: 用户密码的 md5

token: 用户登录后获得的 32 位随机字符串令牌, 用于认证

root_did: 用户的初始目录 id

3. file: 服务器上存在的文件信息（物理文件）

fid	filesize	md5	link_num	name	type
796	8813591	d927004f618b168dd6496c14de4f9441	1	wpy4R627W7MH41jGbt72u4kh_1951330-cameraVideo-2022-5-1217h42m13s.web	0
797	17632133	779852d0b2e1c627c015978df856bed5	1	wpy49YOE594LS06S3t26r8RB_1951330-cameraVideo-2022-5-1217h48m30s.web	0
798	8573156	c02174a89410e27e334cb5fef0455b4c	1	wpy4bpXzWHF9RQ5BkUV6hoQ_1951330-cameraVideo-2022-5-1218h15m2s.webm	0
799	693651	bdcc5766cd95598d0a673b44c4edc45d	1	wpy4Z6vRaj52G5V10ia26sDN_1951330-cameraVideo-2022-5-1219h18m3s.webm	0
800	474537	a9d3c6c29058146b765bd1b66ade0796	1	wpy4sF2q110TMD6C1C6qtCW_1951330-cameraVideo-2022-5-1219h22m51s.web	0
801	228938	12ae1138549066d24713fd1d03888793	2	wpy40W3SIX5te824dde8fYvY_1951330-cameraVideo-2022-5-1219h24m15s.web	0
802	6988317	1c96839ed7ffe90eacf7000a729179e5	1	wpy4pmr4G317kt1BP9YMcHg_1951330-cameraVideo-2022-5-1220h49m15s.web	0
803	2009899	331c3ae95773badc5a0a7b1a8720f2ed	2	wpy4qE6hIfkcc4XSSWVaF0n2_1951330-cameraVideo-2022-5-1413h35m15s.web	0
804	403714	c20683e9f37e8f0729af530d0977b63e	1	wpy47IPQDPXa6bIyomNajuFw_1951330-cameraVideo-2022-5-1414h11m55s.web	0
805	0	d41d8cd98f00b204e9800998ecf8427e	2	wpy4K0677Tq835d3xcP9Dgq1_1.txt	0

Field	Type	Comment
fid	int(11) NOT NULL	
filesize	bigint(20) NOT NULL	
md5	varchar(32) NOT NULL	
link_num	int(11) NOT NULL	
name	varchar(500) NOT NULL	
type	varchar(20) NOT NULL	

各个字段的含义：

fid: 物理文件 id

filesize: 文件大小，单位字节

md5: 文件 md5

link_num: 物理文件对应的逻辑文件数，当此项为 0 时应该删除物理文件

name: 物理文件名，构造方法是 用户名+20 位随机字符串+逻辑文件名

type: 文件类型，暂时没有用到（之前打算将文件名和后缀分开）

4. file_map: 将服务器上文件映射到用户目录的逻辑文件信息（逻辑文件）

fmid	fid	username	did	name	type	md5
949	796 wpy4		901	1951330-cameraVideo-2022-5-1217h42m13s.webm	0	d927004f618b168dd6496c14de4f9441
950	797 wpy4		901	1951330-cameraVideo-2022-5-1217h48m30s.webm	0	779852d0b2e1c627c015978df856bed5
951	798 wpy4		901	1951330-cameraVideo-2022-5-1218h15m2s.webm	0	c02174a89410e27e334cb5fef0455b4c
952	799 wpy4		901	1951330-cameraVideo-2022-5-1219h18m3s.webm	0	bdcc5766cd95598d0a673b44c4edc45d
953	800 wpy4		901	1951330-cameraVideo-2022-5-1219h22m51s.webm	0	a9d3c6c29058146b765bd1b66ade0796
954	801 wpy4		901	1951330-cameraVideo-2022-5-1219h24m15s.webm	0	12ae1138549066d24713fd1d03888793
955	802 wpy4		901	1951330-cameraVideo-2022-5-1220h49m15s.webm	0	1c96839ed7ffe90eacf7000a729179e5
956	803 wpy4		901	1951330-cameraVideo-2022-5-1413h35m15s.webm	0	331c3ae95773badc5a0a7b1a8720f2ed
957	804 wpy4		901	1951330-cameraVideo-2022-5-1414h11m55s.webm	0	c20683e9f37e8f0729af530d0977b63e
958	805 wpy4		905	1.txt	0	d41d8cd98f00b204e9800998ecf8427e

Field	Type	Comment
fmid	int(11) NOT NULL	
fid	int(11) NOT NULL	
username	varchar(50) NOT NULL	
did	int(11) NOT NULL	
name	varchar(500) NOT NULL	
type	varchar(20) NOT NULL	
md5	varchar(32) NOT NULL	

各个字段的含义：

fmid: 逻辑文件 id

fid: 逻辑文件对应的物理文件 id

did: 逻辑文件所属的目录 id

name: 逻辑文件名

type: 文件类型，暂时没有用到

md5: 逻辑文件对应的物理文件 md5

5. task: 用户提交的上传、下载任务信息

tid	f_tid	type	offset	size	last_time	ftype	dst_dir	md5	username	filename
841	-1	file_down	8388608	17632133	1656573391	null	-1	779852d0b2elc627c015978df856bed5	wpy4	/camera_video/1951330-cameraVideo-2022-5-1217h48m30s.webm
1042	-1	file_down	0	49546189	1656675715	null	-1	87548983edd5d29e987db91fd4d98a26	wpy4	/test2
1043	-1	file_down	0	11975472	1656675749	null	-1	ef52734bca6ba2a72233884562f226aa	wpy4	/MathType.zip
1046	-1	file_down	0	11975472	1656676171	null	-1	ef52734bca6ba2a72233884562f226aa	wpy4	/MathType.zip
1047	-1	file_down	0	11975472	1656676253	null	-1	ef52734bca6ba2a72233884562f226aa	wpy4	/MathType.zip
1052	-1	file_down	0	8813591	1656676415	null	-1	d927004f618b168dd6496c14de4f9441	wpy4	/camera_video/1951330-cameraVideo-2022-5-1217h48m30s.webm

Field	Type	Comment
tid	int(11) NOT NULL	
f_tid	int(11) NOT NULL	
type	varchar(10) NOT NULL	
offset	bigint(20) NOT NULL	
size	bigint(20) NOT NULL	
last_time	bigint(20) unsigned NOT NULL	
ftype	varchar(20) NOT NULL	
dst_dir	int(11) NOT NULL	
md5	varchar(32) NOT NULL	
username	varchar(50) NOT NULL	
filename	varchar(500) NOT NULL	
realname	varchar(500) NOT NULL	
local_dir	varchar(500) NOT NULL	

各个字段的含义:

- tid: 任务 id
- f_tid: 任务的父任务 id, 暂时没用 (之前用于设计目录传输的进度)
- type: 任务类型, 表示上传或者下载任务
- offset: 文件传输的进度
- size: 传输文件的总大小
- last_time: 最后一次传输的时间 (用于计算传输速度)
- ftype: 文件类型, 暂时没用
- dst_dir: 文件上传到网盘的目录
- md5: 文件的 md5
- username: 任务所属用户名
- filename: 任务涉及的逻辑文件名
- realname: 任务涉及的物理文件名
- local_dir: 用户上传或者下载时选择的本地目录

二、服务端总体设计

服务端通过 8 个 cgi 文件实现, 各个 cgi 的接口详见 G00102/document 中 Server-API.xlsx

```
[root@VM-12-2-centos src]# ll
总用量 3392
-rwxrwxrwx 1 root root 433552 7月 1 21:28 windows-create-task.cgi
-rwxrwxrwx 1 root root 429320 7月 1 21:28 windows-download.cgi
-rwxrwxrwx 1 root root 433304 7月 1 21:28 windows-getfiles.cgi
-rwxrwxrwx 1 root root 433304 7月 1 21:28 windows-get-task.cgi
-rwxrwxrwx 1 root root 433304 7月 1 21:28 windows-login.cgi
-rwxrwxrwx 1 root root 437496 7月 1 21:28 windows-other.cgi
-rwxrwxrwx 1 root root 433304 7月 1 21:28 windows-register.cgi
-rwxrwxrwx 1 root root 433920 7月 1 21:28 windows-upload.cgi
```

功能	cgi	服务端接受字段	含义	服务端返回字段	含义
创建单文件下载任务	/cgi-bin/src/Windows-creat-task.cgi				
		token	认证令牌	result	0: 任务创建完成 1: 用户认证失败 2: 要下载的文件不存在
	如果继续下载, 需要的字段如下 type: 任务类型 "file_down"	type	任务类型 "file_down"	filesize	文件大小
	tid	filename	要下载的文件名 "/mysrc/eg.txt"	md5	文件md5
	username	username	用户名	tid	任务id
	token	local_dir	用户本地的下载目的路径		
下载文件	/cgi-bin/src/Windows-download.cgi				
		username	用户名	tid	任务名
		tid	任务id	offset	文件偏移量
		offset	文件偏移量	size	本次下载的字节数
		size	本次要下载的字节数	内容	
		finish	1: 完成 0: 未完成 所有字节下载完后传输一次 finish信号表示任务完成	出错则直接返回错误信息	
				收到finish返回:	finished
返回所有文件	/cgi-bin/src/Windows-getfiles.cgi	username	用户名	result	0: 成功 1: 用户认证失败
		token	用户令牌	files	文件名列表 (vector<string>) 目录列表

服务端和客户端除了文件传输, 都用 json 数据传输字段, 各个 cgi 简要介绍:

1. Windows-login.cgi: 用户登录接口

在这个 cgi 中主要完成用户的登录认证, 通过比对用户上传的密码 md5 和数据库中的密码 md5 来对用户进行认证, 根据认证结果返回不同的数字

2. Windows-register.cgi: 用户注册接口

在这个 cgi 中主要完成新用户的注册, 密码强度的检查在客户端, 此处主要负责检查有无同名, 并且在创建新用户成功时同时创建新用户的根目录

3. Windows-other.cgi: 文件操作接口

在这个 cgi 中主要完成客户端对文件的一些操作, 比如文件复制, 目录移动, 服务端通过客户端上传的任务类型来判断进行的操作。目录移动的递归、父子目录的判断都在此处进行, 客户端只需要提供相应的目录名即可完成操作

4. Windows-getfiles.cgi: 获取文件信息接口

在这个 cgi 中主要完成客户端获取用户拥有的文件和目录的请求, 将用户的文件和目录返回给客户端

5. Windows-creat-task.cgi: 任务创建接口

在这个 cgi 中主要完成客户端对上传或者任务的建立, 用户需要传给服务器足够的信息来创建一个任务, 之后再次获取任务信息只需要提供任务 id 等少量信息。在创建任务时就会进行秒传判断、续传判断(下载由客户端判断秒), 需要注意的是用户所有的上传和下载都需要从这个接口开始, 由服务端找到相应任务后再与客户端商量之后的传输

6. Windows-get-task.cgi: 任务信息获取接口

通过这个接口, 客户端可以获取到用户所有的任务信息, 用于断点续传和任务进度显示

7. Windows-upload.cgi: 文件上传接口

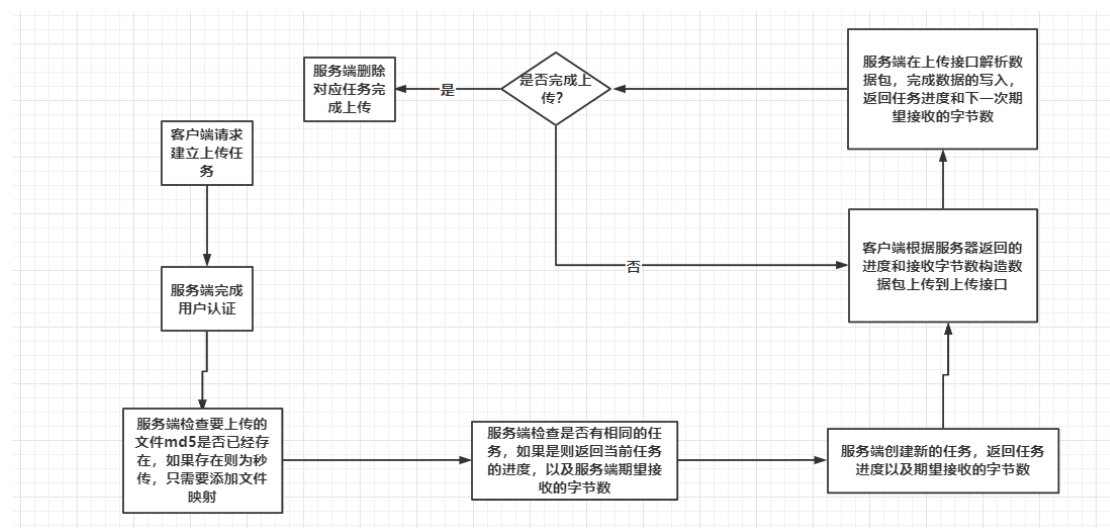
这个接口接收用户传输的文件数据并写入到对应物理文件中。每次传输用户需要提供传输的字节数以及偏移，服务端在写入后返回下一次传输的字节数和偏移，下载和上传每次传输的文件大小最大为 2MB

8. Windows-download.cgi: 文件下载接口

这个接口主要完成用户的下载数据请求，用户需要告诉服务端下载的文件路径、偏移量、字节数。服务端在读取相应的数据后传输数据给客户端

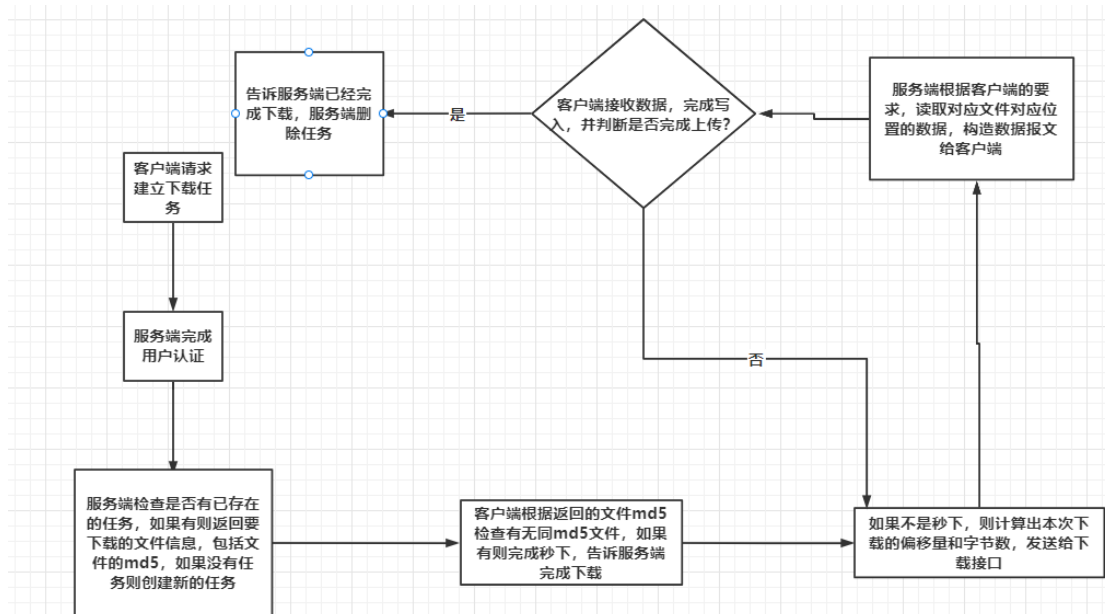
三、文件上传、下载功能实现

1. 文件上传流程图:



客户端首先提交上传任务，服务端完成用户认证后（认证失败则返回相应信息），检查上传的文件的 md5 是否已存在，如果存在则秒传，返回上传成功的消息；如果 md5 不同则检查是否已经有对应上传任务，如果有则根据任务的进度和单次传输大小限制返回下一次传输的偏移量以及字节数；如果没有对应任务服务端创建新的任务，返回下次传输的偏移量和字节数。客户端在接收到偏移量和字节数后，读取本地文件对应的数据，构造一定结构的数据报文给服务端，服务端解析报文，获取本次上传的字节数和偏移，将数据写入到物理文件中，如果文件完成上传则返回上传成功的消息，否则继续返回下次传输的偏移量和字节数。

2. 文件下载流程图:



客户端先向服务端提交下载任务建立请求，服务端在检查完是否已存在同样任务后将文件的 md5 等信息返回给客户端，客户端检查本地是否有同位置同 md5 的文件，如果有则告诉服务端完成秒下；如果不是秒下，客户端应该根据本地已存在的文件大小计算出本次下载的字节数和偏移量发送给服务端下载接口，之后下载接口根据客户端的请求读取文件相应的位置数据，构造数据报文发送给客户端；客户端收到数据后，根据报文中的字节数和偏移量完成写入，如果没有下载完则继续发送下载请求，如果已完成下载则发送完成消息给服务端，服务端收到完成消息后删除对应任务

四、Client 与 Server 端数据传输实现

客户端和服务端一般通过 json 数据传输请求和相应，通过 http 头中的 Content-Length 判断本次的传输数据大小。

json 数据示例如下（一次文件列表请求）：

```

POST http://121.4138.47:80/cgi-bin/src/Windows-getfiles.cgi
Body
[{"username": "wpy4", "token": "c5c0wz2m5fQw1ChsX5v84M7UGhHot76o"}]
Body
{"files": [{"/camera_video/1951338/1951338-cameraVideo-2022-5-1217h42m13s.webm", "/camera_video/1951338/1951338-cameraVideo-2022-5-1217h48m38s.webm", "/camera_video/1951338/1951338-cameraVideo-2022-5-1218h15m2s.webm", "/camera_video/1951338/1951338-cameraVideo-2022-5-1219h18m3s.webm"}]}
  
```

如果是文件数据则用我们自定义的格式进行传输：
数据示例（一次下载请求）：

```
1 | |
2 | |... "username": "wpv",
3 | |... "tid": 115,
4 | |... "offset": 0,
5 | |... "size": 200,
6 | |... "finish": 0
7 | |
```

```
ody Cookies Headers (6) Test Results
Pretty Raw Preview Visualize Text
1 | tid=115 offset=0 size=200
2 |
3 | Content-Type: application/octet-stream
4 |
5 | /*****
6 | ** Form generated from reading UI file 'rankinglist.ui'
7 | **
8 | ** Created by: Qt User Interface Compiler version 5.12.6
9 |
10 | -----WebKitFordsjluagkhuffdsgaskddhadksghfgvfhsjavdskadasmBoundary
```

对于客户端发送的数据，服务端 cgi 直接从标准输入中读取到缓存区，直到读取到 Content-Length 大小的数据才进行下一步

六、遗憾，不足

本次作业没有实现多用户上传，感觉只用 cgi 有点难以实现任务的及时分配。同时也有很多待改进的地方，比如用户上传同名文件没有设计提示是否覆盖，而是直接取消上传；此外对于目录的上传和下载没有进行任务记录，导致目录上传和下载本质上还是单个文件的传输，最后对于任务的下载速度没有很好地实现。