Protocols of sending program task and files materials via TCP

Adopting asynchronous sockets to speeding up sending; there is no command answer during the file sending process, except receiving one terminal side finish answer before and after sending.

Computer side is TCP client side, controller is server side.

1. Create sockets.

2. connecting through controller IP address, default port is 3333

Antecedent command: check the file id list that pending for download 【user sending to controller】

{"\_type":"consult","proName":"programTask.json", "idList":["id1","id2",...],"proName":"program.json","zVer":"xixun1"}

User sending above commands to controller, idList contains all files id pending for download in the programTask

Controller return data to users 【controller send to user】in below:

{"\_type":"consult", "idList":["id1","id2",...]}

While idList is the files id pending for download, users send files according to this idlist to avoid sending the same files repeatedly.

1. sending program task start command 【users send to controller】

{"\_type":"proStart","proName":"programTask.json","proSize":102828,"zVer":"xixun1"}

Sending all files in loop including programTask.json file and material files.

Sending file start command [ users send to controller]

{"\_type":"fileStart","id":"abc.xxx","relative\_path":"","size":3191,"zVer":"xixun1"}

Sending file data, no protocols, Raw data from beginning to end of file

Sending file end command [ users send to controller]

{"\_type":"fileEnd","id":"abc.xxx","zVer":"xixun1"}

Sending program task complete command [users send to controller]

{"\_type":"proEnd","proName":"programTask.json","zVer":"xixun1"}

{"\_type":"AckSuccess"}

Receiving terminal side complete notification and then to exit thread, in this way, will avoid the thread being exited and closed sockets before server side not getting completely and caused server side get incomplete data. [controller send to users]

Close sockets and exit thread

**Controller as TCP server**

Adopting asynchronous sockets, receiving thread cache and then analysis the buf data that read, the processing function needs to solve the problems of sticking package, subcontracting package and breaking package.

please check the procedure in below:

Create service sockets, build up monitor, default port is 3333

Receiving the connection and create receiver sockets

{"\_type":"proStart","proName":"programTask.json","proSize":102828,"zVer":"xixun1"}

Receiving the programtask start command, “proName” key value is the current program task name

{"\_type":"fileStart","id":"abc.xxx","relative\_path":"","size":3191,"zVer":"xixun1"}

Receiving file starting command, create file (save file root path according to the requests of controller app, please note that"relative\_path", “”means under the root path, while "/234234344" means under the 234234344 folder of root path )

Receiving file data, no protocol, naked data from file start to end.

sending file end command

{"\_type":"fileEnd","id":"abc.xxx","zVer":"xixun1"}

{"\_type":"proEnd","proName":"programTask.json","zVer":"xixun1"}

Receiving program task finish command, controller can find program task file according the value of "proName"

{"\_type":"AckSuccess"}

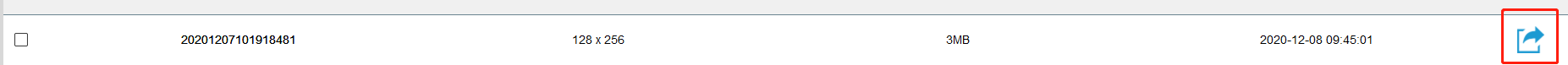
Sending device side finish notification, let computer exit the thread,

The server side thread will close sockets after sent,device side close sockets passively , prepare to close sockets operation, like exit reading thread and so on.

For program json data format please refer to <xixunplayer program JSON document.docx>

**Please download from our website and install the customized ledok Express software.**

**Add programs in ledok software, click play back via USB to export the program zip package file(password can be empty or use default 888, if use different password then need set the same password for controller)**



**zip包节目下发流程：**

**Sending procedure of zip package file:**

Sending a notice to send a zip package, tell zip file name and file size

{"\_type":"fileStart","id":"111.zip","relative\_path":"","size":141349,"zVer":"xixun1"}

Tcp data flow transfer file content, no packets, directly sending binary data

{"\_type":"fileEnd","id":"111.zip","zVer":"xixun1"}

Sending file transfer finish notice

{"\_type":"playZipTask","proName":"111.zip","zVer":"xixun1"}

Sending display zip file notice, then xixunplayer will display zip package content after unzip.

{"\_type":"DelPrograms","\_id": (如: 98e8d3cd47fad6ce8e3f7b8d42cb4d9b)","zVer":"xixun1"}

Delete program

Testing procedure:

Use the communication software NetAssist.exe for testing

Program zip file is 111.zip

Step1



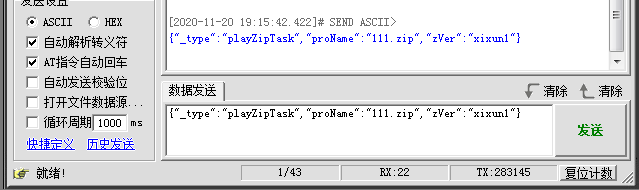
Step2, select to open test file 111.zip (contains two programs, each program is an image.)



Step3, sending file transfer finish command



Step4, sending start to unzip and display command



Finished.