```cpp
#include "widget.h"
#include "ui_widget.h"
#include <QDebug>
#include <synchapi.h>
#include <QMetaType>
#include <QTextCodec>
//
```

```cpp
#include <QTest>
#include <cctype>
#include <fstream>
#include <thread>
#include <chrono>
```

```cpp
#include <cmath>
#include <time.h>
#include <QPalette>
#include <QPixmap>
Q_DECLARE_ME
```

```cpp
TATYPE(QString)

Widget::Widget(QWidget *parent)
    : QWidget(parent)
    , ui(new Ui::Widget)
{
```

```cpp
//
QString::fromLocal8Bit("中文");
    ui->setupUi(this);

    setAutoFillBackground(true);//这句要加上,否则可能显示不出
```

背景图.

```cpp
    QPalette palette = this->palette();

    // palette.setColor(QPalette::Window, Qt::red)
```

```cpp
; // 设置背景色

// palette.setBrush(this->backgroundRole(), Qt::black);/
```

```cpp
/ 设
置背
景色
    p
alette.setBrush(QPalette::Window,

QBru
sh(Q
Pixm
ap("
:/
imag
e/
VCG2
1130
4857
146.
webp
").s
cale
d(
```

```
// 缩放背景图.

this->size(),

Qt::IgnoreAspectRatio,

Qt::SmoothTransf
```

```cpp
orma
tion
)));
//
使用
平滑
的缩
放方
式
    t
his-
>setPalette(palette);
    u
i-
>textEdit_2-
>setStyleSheet("background-
image:url(:/
image/
VCG211258127905.webp)");
    u
i-
```

```cpp
>textEdit-
>setStyleSheet("background-
image:url(:/
image/
VCG211258127905.webp)");
    u
i-
>lineEdit-
>setStyleSheet("background-
image:url(:/
image/
VCG211258127905.webp)");
    u
i-
>lineEdit_3-
>setStyleSheet("background-
image:url(:/
image/
VCG211258127905.webp)");
    /
/
```

```
qrc:/
image/
VCG211298027758.webp
    /
/
this-
>setStyleSheet("border-
image:url(:/
image/
VCG211298027758.webp)");
    /
/连接
部分
    /
/
ip::tcp::endpoint
ep(ip::address::from_string(add.c_str()),
portt);
//
ip::tcp::endpoint
ep(ip::address::from_string("127.0.0.1"),
```

```cpp
8006);
//
sock_=make_shared<ip::tcp::socket>(service);
//
sock_-
>connect(ep);
    q
RegisterMetaType<QString
>
("QString");
    /
/
void
(Widget::*sig_mes)
(QString,unsigned
int)=&Widget::signal_mes;
    /
/
connect(this,sig_mes,this,&Widget::signal_mes_solt);
    c
onnect(this,SIGNAL(signal_mes(QString,unsigned
```

```cpp
int)),this,SLOT(signal_mes_solt(QString,unsigned
int)));
    c
onnect(this,SIGNAL(signal_per(QString)),this,SLOT(si
//
exit_button=new
QPushButton(this);
//
qDebug()<<this-
>x()<<"
"<<this-
>y()<<endl;
//
qDebug()<<this-
>width()<<"
"<<this-
>height()<<endl;
//
qDebug()<<exit_button-
>width()<<"
"<<exit_button-
```

```cpp
>y()<<endl;
//
exit_button-
>move(this-
>width()-
(exit_button-
>width()/
2),this-
>y());
}
void
Widget::signal_mes_solt(QString
mes,unsigned
int
from)
{
//
for(int
i=0;i<mes.size();
++i)
{
```

```cpp
//
qDebug()<<(int)mes[i].toLatin1()<<endl;
//
}
    m
es=mes+"
--来
自"+QString::number(from);
    /
/
this-
>ui-
>lineEdit-
>setText(mes);
//
textEdit_2
    Q
String
s=ui-
>textEdit_2-
>toPlainText();
```

```cpp
    //
    s+='\n';
    s
+=mes;
    s
+="
\n";

    t
his-
>ui-
>textEdit_2-
>setText(s);
}
void
Widget::signal_per_solt(QString
mes)
{
    Q
String
```

```cpp
s=ui-
>textEdit_2-
>toPlainText();
    s
+="
\n";
    s
+=mes;
    s
+="
\n";
    t
his-
>ui-
>textEdit_2-
>setText(s);
}
Widg
et::
~Wid
get(
```

```cpp
)
{
    delete ui;
}

void Widget::init_start()
{ // 启动线程
    cl=make_shared<clint>(sock_);
    //this->ui->textEdit_2->setText(s);
    /
```

```
/ui-
>label_3-
>setText(acc);
    t
his-
>ui-
>label_3-
>setText(QString(QString::number(acc)));
    /
/ui-
>
    /
/
thread
th(&Widget::handle_message,this);
    v
e_th.push_back(std::thread(&Widget::handle_message,t
    v
e_th[0].detach();
}
void
```

```cpp
Widget::on_pushButton_clicked()
{
    t
his-
>close();
}

void
Widget::on_pushButton_6_clicked()
{
//
int
x=this-
>width();
//
int
y=this-
>height();
//
for
(;x>0||
```

```
y>0;--
x,--
y) {
//
if(x<=0)
{
//
x=0;
//
}
//
if(y<=0)
{
//
y=0;
//
}
//
setFixedSize(x,y);
//
//
```

```cpp
QTest::qTest()
//
//
QTest::qSleep(500);
//
};
      /
/
this-
>showMinimized();
      t
his-
>showMinimized();
      /
/
setFixedSize(20,20);
}
int
Widg
et::
head
```

```cpp
anyl
ize(
Head
&head)
{
      /
/

NetPacketHeader*
p =
(NetPacketHeader*)m_getData;
//

printf("
%c
%d
%c
%s
%c
%d
\n",
```

```
p-
>cFlag,
p-
>nLenth,
p-
>cDataType,
p-
>sFileFormat,
p-
>cSendDc,
size);
    /
/
Head*
head
=
(Head*)tmphead;
    /
/
printf("type:
%c
```

```cpp
,length%d
,id%d
,packid%d",
head-
>type,
head-
>length,
head-
>id,
head-
>packid);
      i
f
(head.type
==
'm')
{

retu
rn
1;
```

```cpp
        }
        e
lse
if
(head.type
==
'f')
{

retu
rn
2;
        }
else
if(head.type=='r')
{

retu
rn
3;
        }
```

```cpp
else
if(head.type=='d')
{

retu
rn
4;
    }
    r
eturn
0;
}
void
Widget::handle_message()
{
    s
tring
han_str;
    H
ead
re_he;
```

```cpp
for
(;;)
{

{

unique_lock<mutex>tr
lock
(cl->re_mutex,std::defer_lock);
```

```cpp
if(t
rloc
k.tr
y_lo
ck()
){

if(c
l-
>rea
dqu.
size
()>0
){

han_
str=
cl-
>rea
dqu.
```

```cpp
front();

cl->readqu.pop();

// qDebug()<<"接收到消息"<<QString::fromStdStdStr
```

```cpp
ing(
han_
str)
<<en
dl;

}

}

}

re_h
e=ge
tHea
d(han_st
r);

int
ret=
```

```cpp
head
anyl
ize(
re_h
e);

if(r
et==
1){

string
ng
mes(
han_
str.
begi
n()
+siz
eof(
Head
),ha
```

```cpp
n_str.begin()+(sizeof(Head)+re_he.length));

int from=re_he.account;
```

```cpp
emit
this-
>signal_mes(str2qstr(mes),from);

//
emit
this-
>signal_mes(QString::fromStdString(mes),from);

//
//
string
tmp(buf.begin()
+
sizeofhead,
buf.begin()
+
(sizeofhead
+
leng));
```

```
han_
str.
clea
r();
//清
空字
符串
否则
一直
进入
这个
循环

}
else
if(ret==2)
{

stri
```

```
ng
mes1
(han
_str
.beg
in()
+siz
eof(
Head
),ha
n_st
r.be
gin(
)+
(siz
eof(
Head
)
+re_
he.l
engt
```

```cpp
h));

QString mes(str2qstr(mes1));

QString status;

if(re_he.status==1){
```

```cpp
        status="待接收";

        }
        else if(re_he.status==2)
        {

        status="已接收";

        }
        else if(re_he.status==3)
        {
```

```cpp
            status="
            接收
            失
            败";

            }
            else
            {

            status="
            未知
            错
            误";

            }

            mes+
            ="
```

```cpp
status:
"+status;

// mes+="status:"+std::to_string(re_he.status);

int
```

```cpp
from
=re_
he.a
ccou
nt;

emit
this-
>signal_mes(mes,from);

//
emit
this-
>signal_mes(QString::fromStdString(mes),from);

//
//
string
tmp(buf.begin()
+
sizeofhead,
```

```cpp
buf.begin()
+
(sizeofhead
+
leng));

han_
str.
clea
r();
//清
空字
符串
否则
一直
进入
这个
循环

}
```

```cpp
else
if(ret==3)
{

qDeb
ug()
<<"
接收
到的
服务
器的
回
馈"<
<re_
he.t
ype<
<"
leng
th:"
<<re
```

```cpp
_he.
leng
th<<
"
stat
us:"
<<re
_he.
stat
us<<
endl
;

if(r
e_he
.sta
tus<
=0)
{//
说明
是请
```

```cpp
求到
的文
件id

{

lock
_gua
rd<m
utex
>loc
kk(fireid_mu
);

fire
id.p
ush(
re_h
e.st
```

```
atus
);

}

}

han_
str.
clea
r();
//清
空字
符串
否则
一直
进入
这个
循环
```

```cpp
}
else
if(ret==4)
{

string
mes1
(han
_str
.beg
in()
+siz
eof(
Head
),ha
n_st
r.be
gin(
)+
(siz
```

```cpp
eof(
Head
)
+re_
he.l
engt
h));

//
QStr
ing
mes(
str2
qstr
(mes
1));

int
from
=re_
he.a
```

```cpp
ccount;

emit this->signal_mes(QString::fromStdString(mes1),from);

han_str.clear();//清空字符串否则一直进入这个循环
```

```cpp
}

        }
;
}

void
Widget::on_pushButton_2_clicked()
{
    Q
String
s=ui-
>textEdit-
>toPlainText();
    i
nt
chnum=0;
    f
or(auto
a:s)
{
```

```
if(i
salp
ha(a
.toL
atin
1())
){

}

else
+
+chnum;
    }
//
cout<<endl;
    s
+=QString(chnum,'
');//
还要
```

加和字符一样长度的空格才能结局问题几个中文就需要几个空格

```
        /
/
message+='\n';
        i
f(s.size()>500)
{
```

```cpp
ui->textEdit->setText("字数超过限制");

        return;
    }

    Headwr_he;
    w
```

```cpp
    r_he.type='m';
        w
    r_he.account=this-
>acc;
        w
    r_he.sendto=ui-
>lineEdit_3-
>text().toInt();
        w
    r_he.length=s.size();
        c
har
ttmphe[sizeof(Head)];
        m
emcpy(&ttmphe,&wr_he,sizeof(Head));
        s
tring
mes_buf=string(ttmphe,sizeof(Head))
+qstr2str(s);
        {
```

```cpp
lock_guard<mutex>wr_lock(cl->wr_mutex);

cl->writequ.push(mes_buf);

cl->is_
```

```cpp
have
_tas
k=tr
ue;
    }
    c
l-
>wr_cond.notify_all();
}
void
Widget::lifilethread(string
path,string
name)
{
    c
onst
int
READ_BUFF_SIZ=10*1024;
    i
o_service
servicef;
```

```cpp
    shared_ptr<ip::tcp::socket>sockf;
    shared_ptr<ip::tcp::endpoint>ep;
    try {

ep=make_shared<ip::tcp::endpoint>(ip::address::from_stri
```

```cpp
ng(i
pstr
.c_s
tr()
),
8007
);

qDeb
ug()
<<"
文件
线程
要连
入的
ip"<
<ips
tr.c
_str
()<<
endl
```

```cpp
;

sockf=make_shared<ip::tcp::socket>(servicef);

sockf->connect(*ep);
```

```cpp
    }
catch
(...)
{

//
ui-
>tex
tEdi
t-
>set
Text
("li
nk
195
erro
r");

qDeb
ug()
<<__
```

```cpp
LINE
__<<
"
erro
r"<<
endl
;

retu
rn;
    }


    H
ead
he;
    f
stream
ff;
    s
```

```
tring
buf=path+name;
        i
nt
length;
        i
nt
sen;
        t
ry{

ff.o
pen(
buf.
c_st
r(),
ios:
:in|
ios:
:bin
ary)
```

```
;

ff.seekg(0, ff.end); //追溯到流的尾部

length = ff.tellg(); //获取流的长度
```

```cpp
ff.seekg(0, ff.beg); //回到流的头部

sen=ui->lineEdit_3->text().toInt();
    }
```

```cpp
catch(...)
{

// ui->textEdit->setText("214 error");

qDebug() <<__LINE__<< "
```

```
erro
r"<<
endl
;

retu
rn;
    }

    h
e.type='f';
    h
e.length=name.size();
    h
e.packid=-1;
    h
e.sendto=sen;
    h
e.account=acc;
    h
e.id=0;
```

```
        h
e.status=length;
        /
/
char
tmp[sizeof(Head)];
        /
/
memcpy(tmp,&he,sizeof(Head));
//
string
st=string(tmp,sizeof(Head));
        s
tring
st((char*)&he,sizeof(Head));
        s
t+=name;

        /
/
qDebug()<<"packid
```

```cpp
为-1
时
bufstring"<<QString::fromStdString(st)<<endl;

    t
ry {

sock
f-
>wri
te_s
ome(
buff
er(s
t));
    }
catch
(...)
{

ui-
```

```cpp
>textEdit->setText("235 error");

qDebug()<<__LINE__<<" error"<<endl;
```

```cpp
    return;
    }
    int packid=0;
    int ret;
    char buff[READ_BUFF_SIZ];
    bool isfalse=false;

    std::chrono::system_clock::time_point now;//= chrono::system_clock::now();
```

```cpp
    std::chrono::system_clock::time_point now1;
    std::chrono::nanoseconds d;
    //= now.time_since_epoch();
    std::chrono::nanoseconds d1;
    now1=std::chrono::system_clock::now();
    d1=now1.time_since_epoch();
    const int sizehead=sizeof(Head);

    w
```

```
hile(1)
{

//
cout
<<"
开始
read
some
"<<e
ndl;

try{

sock
f-
>rea
d_so
me(b
uffe
r(st
```

```cpp
        ));

    }
    catch(...){

        // ui->textEdit->setText("256 error");

        qDebug()
```

```
<<__
LINE
__<<
"
erro
r"<<
endl
;

retu
rn;

}

//
cout
<<"
读到
东
西"<
<end
```

```cpp
l;

he=getHead(st);

if(he.status==reback::fitrfail){

isfalse=true;
```

```
break;

}

if(he.status==reback::firetrst)
{qDebug()<<"接收到firetrst目前
```

```cpp
pack
id"<
<pac
kid<
<end
l;}


try{

ff.r
ead(
buff
,siz
eof(
buff
));

}
catc
h(..
```

```cpp
.){

    // ui->textEdit->setText("249 error");

    qDebug()<<__LINE__<<"
    erro
```

```cpp
r"<<
endl
;

retu
rn;

}

ret=
ff.g
coun
t();

//
qDeb
ug()
<<"
写
入"<
<ret
```

```cpp
<<"个字节"<<endl;

if(ret==0)break;

he.type='f';

he.length=ret;

he.p
```

```
acki
d=pa
ckid
++;

he.s
endt
o=0;

he.a
ccou
nt=a
cc;

he.i
d=0;

he.s
tatu
s=1;
```

```cpp
if(ret<READ_BUFF_SIZ){

he.status=2;

}

//qDebug()<<"当前文件packid"<
```

```cpp
<pac
kid<
<end
l;

//
memc
py(t
mp,&
he,s
izeo
f(He
ad))
;


try{

//
st=s
```

```cpp
tring(tmp,sizeof(Head));

st=string((char*)&he,sizehead)+string(buff,ret);
```

```cpp
}
catch(std::exception &e){

ui->textEdit->setText(QString::fromStdString(e.wh
```

```cpp
at()
)
+"27
3err
or")
;

qDeb
ug()
<<QS
trin
g::f
romS
tdSt
ring
(e.w
hat(
))+"
error"<<endl;

qDeb
```

```cpp
ug()<<__LINE__<<"error"<<endl;

    return;

}

try{

sockf->write_s
```

```cpp
ome(
buff
er(s
t));

}
catc
h(..
.){

//
ui-
>tex
tEdi
t-
>set
Text
("26
7
erro
r");
```

```cpp
qDebug() <<__LINE__<<"error"<<endl;

        return;

    }

    // qDebug()<<"
```

```
写
入"<
<ret
<<"
个字
节"<
<end
l;
    }
    i
f(ff.bad())
{

qDeb
ug()
<<"
文件
发送
失败
fail
```

```cpp
"<<endl;

// ui->textEdit->setText("305 error");

qDebug()<<__LINE__<<"
```

```cpp
error"<<endl;
        return;
    }
    if(isfalse)
    {
        qDebug()<<"文件发送失败fail"<<endl;
```

```
// ui->textEdit->setText("310 error");

qDebug()<<__LINE__<<" error"<<
```

```cpp
endl
;

retu
rn;
    }
    q
Debug()<<"文
件发
送成
功
success"<<endl;



    t
ry {

now
=
std:
```

```cpp
:chrono:
:system_
clock::n
ow()
;

d =
now.
time
_sin
ce_e
poch
();

double
dd =
((d
```

```
-
d1).count()
*
pow(10,
-9));

int
time
sudu
=0;

if
(dd
<=
0)re
turn
;

time
sudu
=
```

```
(length
/
dd)
/
1024.0
/
1024.0;

qDeb
ug()
<<
"下载
速度
为"
<<
time
sudu
<<
"mb/
s"
<<
```

```cpp
endl;

QString speed=QString("速度为%1mb/s").arg(timesudu);

QString s;

s+=QString::from
```

```cpp
StdS
trin
g(na
me)
+"
发送
成功
"+sp
eed;
//
QString
s=ui-
>textEdit_2-
>toPlainText();
//
s+="
\n";
//
s+=QString::fromStdString(name)
+"
下载
```

```
成功
"+speed;
//
s+="
\n";
//
this-
>ui-
>textEdit_2-
>setText(s);

sign
al_p
er(s
);
    }
    c
atch
(...)
{
```

```cpp
    return;
    }



    //发送向对方发送文件的消息
    Head wr_he;
    wr_he.type='f';
    wr_he.account=this->acc;
```

```cpp
    w
r_he.sendto=sen;
    w
r_he.length=name.size();
    w
r_he.status=1;//一是发送，2是发送成功的反馈，3是发送失败的反馈//
char
```

```cpp
ttmphe[sizeof(Head)];
//
memcpy(&ttmphe,&wr_he,sizeof(Head));
      s
tring
mes_buf=string((char*)&wr_he,sizeof(Head))
+name;
      {

lock
_gua
rd<m
utex
>wr_
lock
(cl-
>wr_
mute
x);

cl-
```

```
>wri
tequ
.pus
h(me
s_bu
f);

cl-
>is_
have
_tas
k=tr
ue;
    }
    c
l-
>wr_cond.notify_all();

    f
f.close();
```

```cpp
}

void
Widget::refilethread(string
path,string
name)
{
    c
onst
int
BUFSIZE=1024;
    i
o_service
servicef;
    s
hared_ptr<ip::tcp::socket>sockf;
    s
hared_ptr<ip::tcp::endpoint>ep;
    e
p=make_shared<ip::tcp::endpoint>(ip::address::from_s
```

```
8007);
    q
Debug()<<"文
件线
程要
连入
的
ip"<<ipstr.c_str()<<endl;
    s
ockf=make_shared<ip::tcp::socket>(servicef);
    s
ockf-
>connect(*ep);

    H
ead
he;
    f
stream
ff;
    s
```

```
tring
buf=path+name;
     f
f.open(buf.c_str(),ios::out|
ios::binary|
ios::trunc);


     i
nt
sen=ui-
>lineEdit_3-
>text().toInt();
     h
e.type='y';
     h
e.length=name.size();
     h
e.status=1;
     b
uf=string((char*)&he,sizeof(Head));
```

```
    b
uf+=name;
    t
ry {

sock
f-
>wri
te_s
ome(
buff
er(b
uf))
;
    }
catch
(...)
{

qDeb
ug()
```

```cpp
<<"3
87
writ
e
erro
r"<<
endl
;


reba
ckfa
lse(
sen,
name
);


retu
```

```
rn;
    }

    q
Debug()<<"已
向对
方发
布初
始要
接受
文件
的信
息,等
待接
收文
件大
小"<<endl;

    t
ry{
```

```cpp
        sockf->read_some(buffer(buf));
        }
catch(...)
{

qDebug() << __LINE__ <<"write error"<<endl;
```

```
reba
ckfa
lse(
sen,
name
);

retu
rn;
    }
    t
ry {

he=g
etHe
ad((
buf)
);
    }
catch
```

```
(...)
{

qDeb
ug()
<<
__LI
NE__
<<"
write
error"<<endl;

reba
ckfa
lse(
sen,
name
);

retu
rn;
```

```cpp
    }
    /
/
he=getHead((buf));
    i
nt
filelength=he.length;
    i
nt
sizeofhead=sizeof(Head);
    i
nt
cursize=0;
    b
ool
trfalse=false;
    q
Debug()<<"待
接收
文件
大
```

```cpp
小"<<filelength<<endl;
    h
e.status=reback::firetrst;
    s
tring
a((char*)&he,sizeof(Head));
    s
tring
s;
    t
ry {

s.re
size
(BUF
SIZE
+204
8);
    }
catch
(std::exception
```

```cpp
&e)
{

qDeb
ug()
<<QS
trin
g::f
romS
tdSt
ring
(e.what(
))<<
endl
;

qDeb
ug()
<<
__LI
```

```cpp
NE__
<<"
write
error"<<endl;

reba
ckfa
lse(
sen,
name
);


retu
rn;
    }
    /
/
s.resize(7000);
```

```cpp
    std::chrono::system_clock::time_point now;//= chrono::system_clock::now();
    std::chrono::system_clock::time_point now1;
    std::chrono::nanoseconds d;//= now.time_since_epoch();
    std::chrono::nanoseconds d1;
    now1=std::chrono::system_clock::now();
    d1=now1.time_since_epoch();
```

```cpp
while(1)
{

// qDebug()<<"";

try{

sockf->write_some(buffer(a
```

```cpp
        ));

        }
        catch(...){

        // ui->textEdit->setText("357 error");

        qDebug()
```

```
<<
__LI
NE__
<<"
write
error"<<endl;

reba
ckfa
lse(
sen,
name
);

retu
rn;

}

//
qDeb
```

```
ug()
<<"
已发
送
fire
trst
"<<e
ndl;


//
sock
f-
>rea
d_so
me(b
uffe
r(bu
f,bu
f.si
ze()
```

```cpp
));

try{

buf.clear();

buf.resize(BUFSIZE+1024);

}catch(...){
```

```cpp
//    ui->textEdit->setText("357 error");

    qDebug() << __LINE__ <<"write error"<<endl;
```

```
rebackfalse(sen, name);

return;

}

try{

read(*sockf, buffer(buf,buf.s
```

```cpp
ize(
)),transfer_at_least(20));

}catch(...){

qDebug()<<__LINE__<<"write
```

```cpp
error"<<endl;

rebackfalse(sen,name);

return;

}

//qDebug()<<QString::fromS
```

```
tdSt
ring
(buf
)<<e
ndl;

try{

he=g
etHe
ad(buf);

}
catc
h(..
.){

//
ui-
```

```cpp
>textEdit->setText("357 error");

qDebug()<<__LINE__<<"write error"<<endl;

rebackfa
```

```cpp
lse(
sen,
name
);

retu
rn;

}

try{

s=st
ring
(buf
.beg
in()
+siz
eofh
ead,
buf.
```

```cpp
begin()+(sizeofhead+he.length));

} catch(...){

int i;

for(i=sizeofhead
```

```
;i<b
uf.s
ize(
);+
+i){

if(b
uf[i
]=='
\0')
brea
k;

}

qDeb
ug()
<<"
接收
错误
时接
```

```
收到
的数
据
量"<
<i<<
endl
;

Head
*h=(Head*)
(string(buf.begin(),buf.begin()
+sizeof(Head)).c_str());

qDeb
ug()
<<"h
-
>len
gth:
"<<h
-
```

```
>length<<endl;

qDebug()<<"buf"<<buf.size()<<"he.length:"<<he.length<<"内容: "<<QString::fromStdString(buf)<<endl;

qDebug()<<__LINE__
```

```cpp
<<"
write
error"<<endl;


reba
ckfa
lse(
sen,
name
);



retu
rn;

}

//
```

```
qDebug()<<"读取到"<<he.length<<"字节"<<endl;

cursize += he.length;
```

```cpp
// qDebug()<<"cursize:"<<cursize<<endl;

if (ff.is_open()) {

ff.write(s.c_str
```

```
(),h
e.le
ngth
);

}

else
{

trfa
lse
=
true
;

qDeb
ug()
<<
__LI
NE__
```

```cpp
<<"
write
error"<<endl;

}

if
(he.
stat
us
==
2)
{//
最后
一个
包

if
(cur
size
==
```

```cpp
filelength)
{//
接收
完毕

ff.c
lose
();

he.s
tatu
s=re
back
::lo
gsuc
c;

stri
ng
a((c
har*
```

```cpp
)&he
,siz
eof(
Head
));

//
try{
//
sockf-
>write_some(buffer(a));
//
}
catch(...)
{
//
//
ui-
>textEdit-
>setText("401
error");
```

```cpp
//
qDebug()<<"486
error"<<endl;
//
rebackfalse(sen,name);
//
return;
//
}

break;

}

else
{

trfa
lse
=
```

```cpp
        true
        ;

        qDeb
        ug()
        <<"4
        93
        erro
        r"<<
        endl
        ;

    }

    }

    if
    (ff.
    bad(
    )) {
```

```cpp
trfalse = true;

qDebug()<<"344 error"<<endl;

ff.close();

qDebug()
```

```
<<"
文件
传输
失
败"<
<end
l;


//发
送向
对方
发送
文件
的消
息

reba
ckfa
```

```cpp
        lse(
        sen,
        name
        );


        retu
        rn;

        }

        if
        (trf
        alse
        )
        {//
        出问
        题了
        on_w
        rite
        error
```

然后
直接
关闭

```cpp
ff.close();

qDebug()<<"文件传输失败"<<endl;

qDeb
```

```
ug()
<<
__LI
NE__
<<"
write
error"<<endl;

//发
送向
对方
发送
文件
的消
息

reba
ckfa
lse(
sen,
name
```

```
    );


    retu
    rn;

    }
        }
        q
Debug()<<"文
件传
输成
功"<<endl;
        s
ockf-
>close();

        t
ry {
```

```cpp
now = std::chrono::system_clock::now();

d = now.time_since_epoch();

doub
```

```cpp
le
dd =
((d
-
d1).count()
*
pow(10,
-9));

int
time
sudu
=0;

if
(dd
<=
0)re
turn
;
```

```
time
sudu
=
(filelength
/
dd)
/
1024.0
/
1024.0;

qDeb
ug()
<<
"下载
速度
为"
<<
time
sudu
<<
```

```
"mb/
s"
<<
endl;

QStr
ing
spee
d=QS
trin
g("
速度
为%1
mb/
s").arg(timesudu);

QStr
ing
s;

s+=Q
```

```cpp
String::fromStdString(name)+"接收成功"+speed;
//QString s=ui->textEdit_2->toPlainText();
//s+="\n";
//
```

```
s+=QString::fromStdString(name)
+"
下载
成功
"+speed;
//
s+="
\n";
//
this-
>ui-
>textEdit_2-
>setText(s);

sign
al_p
er(s
);
    }
    c
atch
```

```
(...)
{

retu
rn;
    }



    /
/发送
向对
方发
送文
件的
消息
    H
ead
wr_he;
    w
r_he.type='f';
```

```cpp
        w
r_he.account=this-
>acc;
        w
r_he.sendto=sen;
        w
r_he.length=name.size();
        w
r_he.status=2;//一是发送，2是发送成功的反馈，3是发送失败的
```

反馈

```cpp
//
char
ttmphe[sizeof(Head)];
//
memcpy(&ttmphe,&wr_he,sizeof(Head));
     s
tring
mes_buf=string((char*)&wr_he,sizeof(Head))
+name;
     {

lock
_gua
rd<m
utex
>wr_
lock
(cl-
>wr_
mute
```

```cpp
x);

cl->writequ.push(mes_buf);

cl->is_have_task=true;
    }
    cl->wr_cond.notify_all();
```

```cpp
}

void
Widget::rebackfalse(int
sen,string
name)
{
    Head
wr_he;
    w
r_he.type='f';
    w
r_he.account=this-
>acc;
    w
r_he.sendto=sen;
    w
r_he.length=name.size();
    w
r_he.status=3;//
```

一是
发
送，2
是发
送成
功的
反
馈，3
是发
送失
败的
反馈

```cpp
    s
tring
mes_buf=string((char*)&wr_he,sizeof(Head))
+name;
    {

lock
_gua
```

```
rd<m
utex
>wr_
lock
(cl-
>wr_
mute
x);

cl-
>wri
tequ
.pus
h(me
s_bu
f);

cl-
>is_
have
_tas
```

```cpp
k=tr
ue;
    }
    c
l-
>wr_cond.notify_all();

}
void
Widget::on_pushButton_5_clicked()
{
    /
/
char*
path=(char
*)"test.txt";
    /
/
string
name="tongji.cpp";
    s
```

```cpp
tring
name=ui-
>lineEdit-
>text().toStdString();
    /
/
string
path="C:/
Users/
trn123456/
Desktop/
build-
tfClint-
Desktop_Qt_5_14_2_MinGW_64_bit-
Debug/
debug/";
    s
tring
path="./";
    s
td::thread
```

```cpp
th(&Widget::lifilethread,this,path,name);
    /
/
std::thread
th(&Widget::refilethread,this,path,name);
    t
h.detach();
}

void
Widget::on_pushButton_4_clicked()
{
    s
tring
name=ui-
>lineEdit-
>text().toStdString();
    /
/
string
path="C:/
```

```
Users/
trn123456/
Desktop/
build-
tfClint-
Desktop_Qt_5_14_2_MinGW_64_bit-
Debug/
debug/";
    s
tring
path="./";
    /
/
std::thread
th(&Widget::lifilethread,this,path,name);
    s
td::thread
th(&Widget::refilethread,this,path,name);
    t
h.detach();
}
```

```
void
Widget::on_pushButton_3_clicked()
{
    u
i-
>textEdit-
>setText("");
}

void
Widget::on_pushButton_7_clicked()
{
    /
/发送
向对
方发
送文
件的
消息
    H
```

```cpp
ead
wr_he;
    w
r_he.type='d';
    w
r_he.account=this-
>acc;
    w
r_he.sendto=this-
>acc;
    w
r_he.length=0;
    s
tring
mes_buf=string((char*)&wr_he,sizeof(Head));
    {

lock
_gua
rd<m
utex
```

```
>wr_
lock
(cl-
>wr_
mute
x);

cl-
>wri
tequ
.pus
h(me
s_bu
f);

cl-
>is_
have
_tas
k=tr
ue;
```

```cpp
    }
    c
l-
>wr_cond.notify_all();

}
//
void
Widget::mousePressEvent(QMouseEvent
*event)
//{
//
//
如果
是鼠
标左
键按
下
//
if(event-
>button()
```

```
==
Qt::LeftButton)
//
{
//
this-
>move(this-
>x()
+event-
>x(),this-
>y()
+event-
>y());
//
}
//
//
如果
是鼠
标右
键按
```

```cpp
下
////
else
if(event-
>button()
==
Qt::RightButton)
////
{
////
...
////
}
//}


void
Widget::mousePressEvent(QMouseEvent
*event)
{
```

```cpp
    if(event->button() == Qt::LeftButton)
    {
        mousePress = true;
    }
    //窗口移动距离
    movePoint = event->globalPos()
```

```cpp
-
pos();
}

void
Widget::mouseReleaseEvent(QMouseEvent
*event)
{
    Q
_UNUSED(event)
    m
ousePress
=
false;
}

void
Widget::mouseMoveEvent(QMouseEvent
*event)
{
    i
```

```cpp
f(mousePress)
    {

QPoi
nt
move
Pos
=
even
t-
>glo
balP
os()
;

move
(mov
ePos
-
movePoint);
    }
```

```
}
```