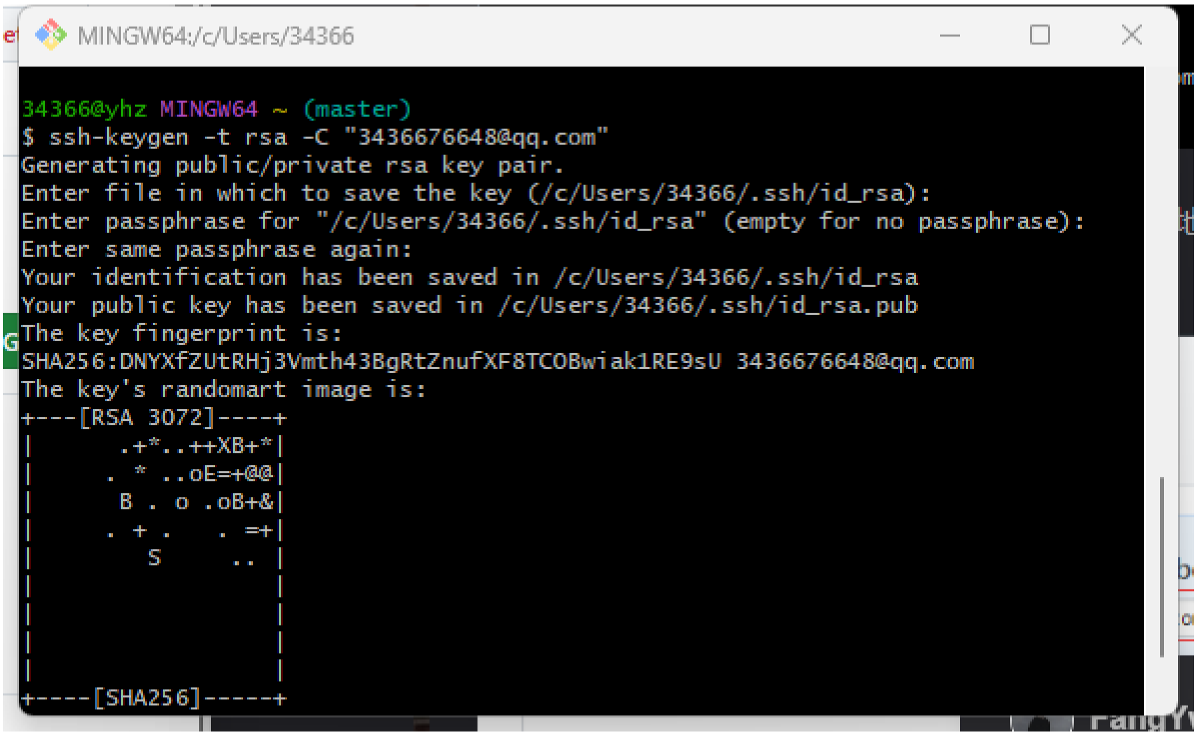
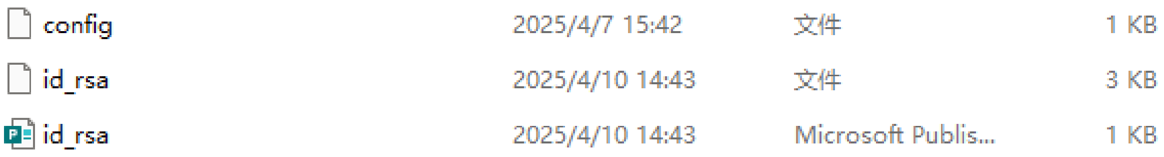
一：github仓库建立与配置

1. 注册登录后右上角创建new repository，设置仓库名称、描述、是否公开等信息
2. 安装git bash
3. 获取配置密钥
   - 打开git bash 输入`ssh-keygen -t rsa -C "3436676648@qq.com`



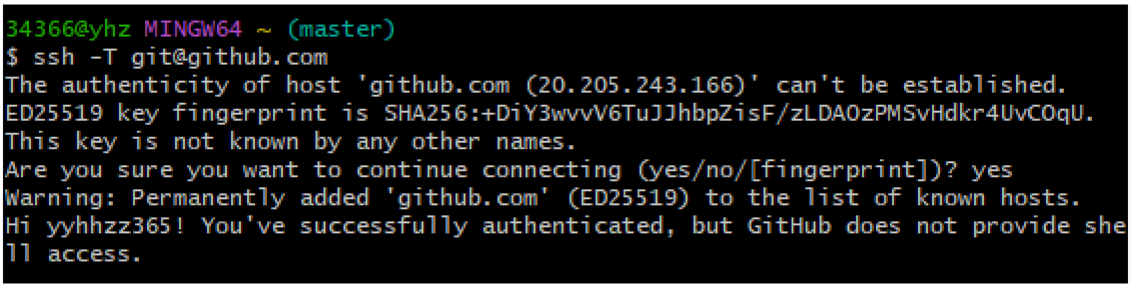
```
34366@yhz MINGW64 ~ (master)
$ ssh-keygen -t rsa -C "3436676648@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/34366/.ssh/id_rsa):
Enter passphrase for "/c/Users/34366/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/34366/.ssh/id_rsa
Your public key has been saved in /c/Users/34366/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:DNYXfZUtRHj3Vmth43BgRtZnufXF8TCOBwiak1RE9sU 3436676648@qq.com
The key's randomart image is:
+---[RSA 3072]----+
|      .+*..++XB+*|
|     . * ..oE=+@@|
|      B . o .oB+&|
|     . + .   . =+|
|        S     .. |
|                 |
|                 |
|                 |
|                 |
+----[SHA256]-----+
```

   - 打开.ssh文件位置，打开id_rsa.pub文件，复制密钥



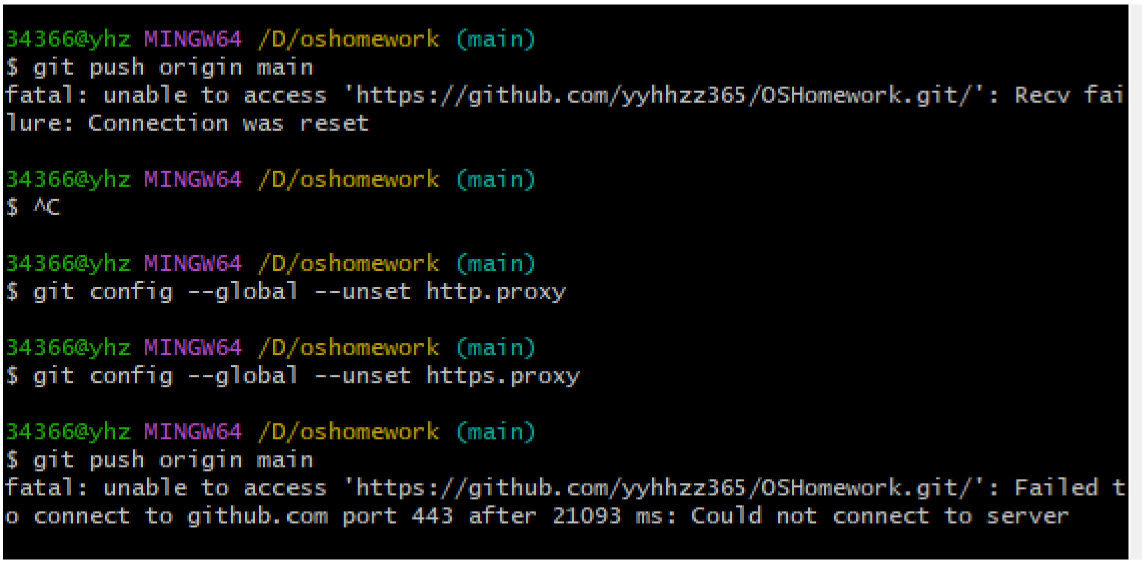
config	2025/4/7 15:42	文件	1 KB
id_rsa	2025/4/10 14:43	文件	3 KB
id_rsa	2025/4/10 14:43	Microsoft Publis...	1 KB

   - 打开github设置，在SSH and GPG keys中添加SSH key并粘贴密钥
   - 打开git bash，输入`ssh -T git@github.com`测试是否成功



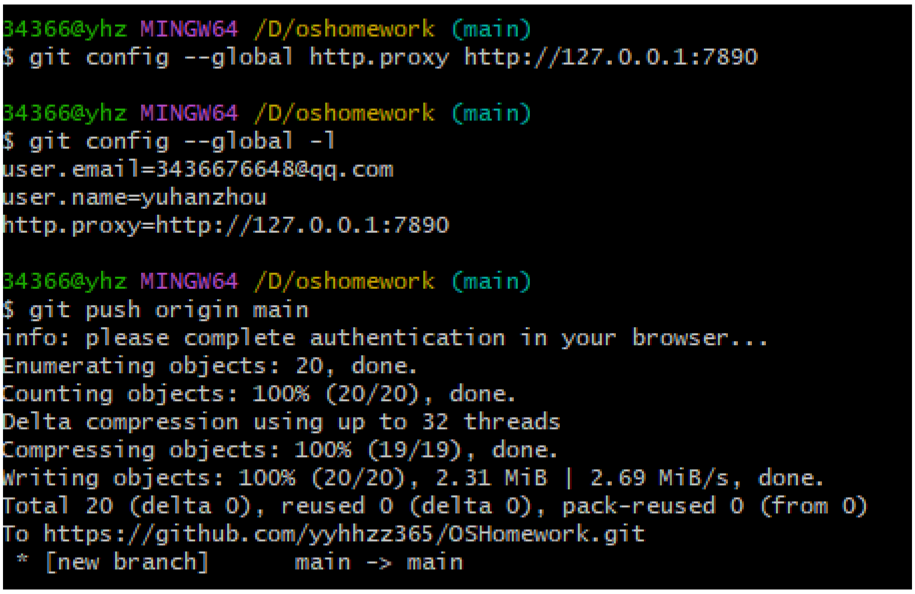
```
34366@yhz MINGW64 ~ (master)
$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi yyhhzz365! You've successfully authenticated, but GitHub does not provide she
ll access.
```

4. 上传文件
   - 输入`git config --global user.name "your name"`和`git config --global user.email "your email"`设置账号名和邮箱
   - 将github创建仓库git clone到本地
   - 打开git bash定位到本地仓库，输入`git add hw1 hw2` 再输入`git commit -m "first commit"` 然后输入`git push origin master`将本地仓库推送到github仓库。报错



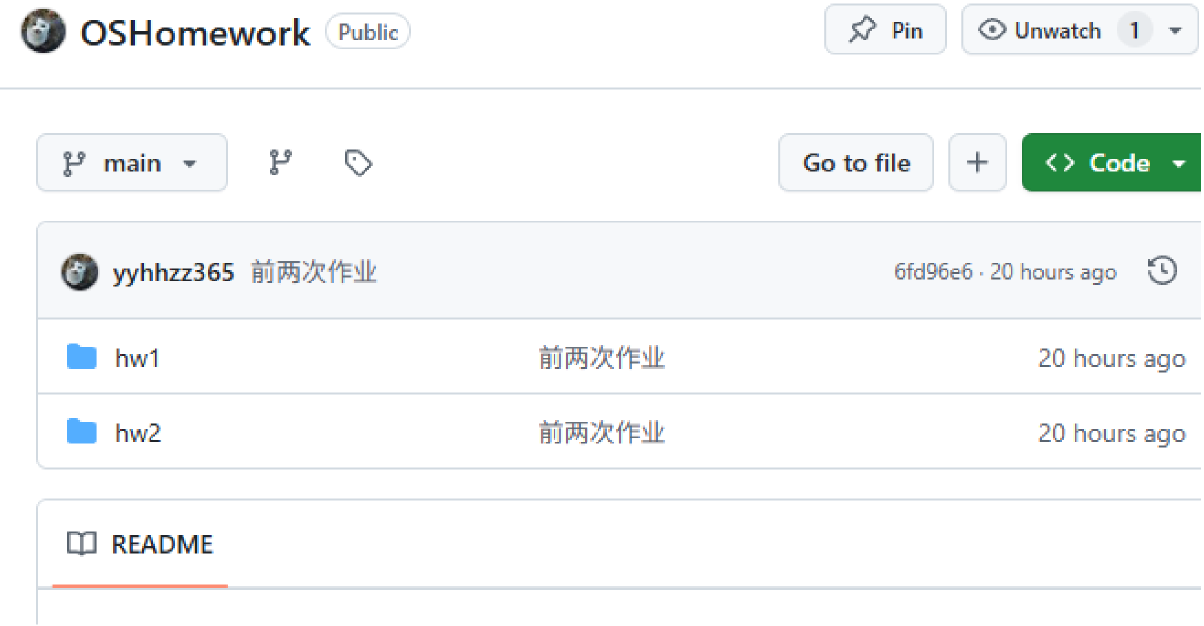
```
34366@yhz MINGW64 /D/oshomework (main)
$ git push origin main
fatal: unable to access 'https://github.com/yyhhzz365/OSHomework.git/': Recv fai
lure: Connection was reset

34366@yhz MINGW64 /D/oshomework (main)
$ ^C

34366@yhz MINGW64 /D/oshomework (main)
$ git config --global --unset http.proxy

34366@yhz MINGW64 /D/oshomework (main)
$ git config --global --unset https.proxy

34366@yhz MINGW64 /D/oshomework (main)
$ git push origin main
fatal: unable to access 'https://github.com/yyhhzz365/OSHomework.git/': Failed t
o connect to github.com port 443 after 21093 ms: Could not connect to server
```

  - 进行代理设置`git config --global http.proxy http://127.0.0.1:7890`，再次尝试成功



```
34366@yhz MINGW64 /D/oshomework (main)
$ git config --global http.proxy http://127.0.0.1:7890

34366@yhz MINGW64 /D/oshomework (main)
$ git config --global -l
user.email=3436676648@qq.com
user.name=yuhanzhou
http.proxy=http://127.0.0.1:7890

34366@yhz MINGW64 /D/oshomework (main)
$ git push origin main
info: please complete authentication in your browser...
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 32 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (20/20), 2.31 MiB | 2.69 MiB/s, done.
Total 20 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/yyhhzz365/OSHomework.git
 * [new branch]      main -> main
```

输入账号密码后成功



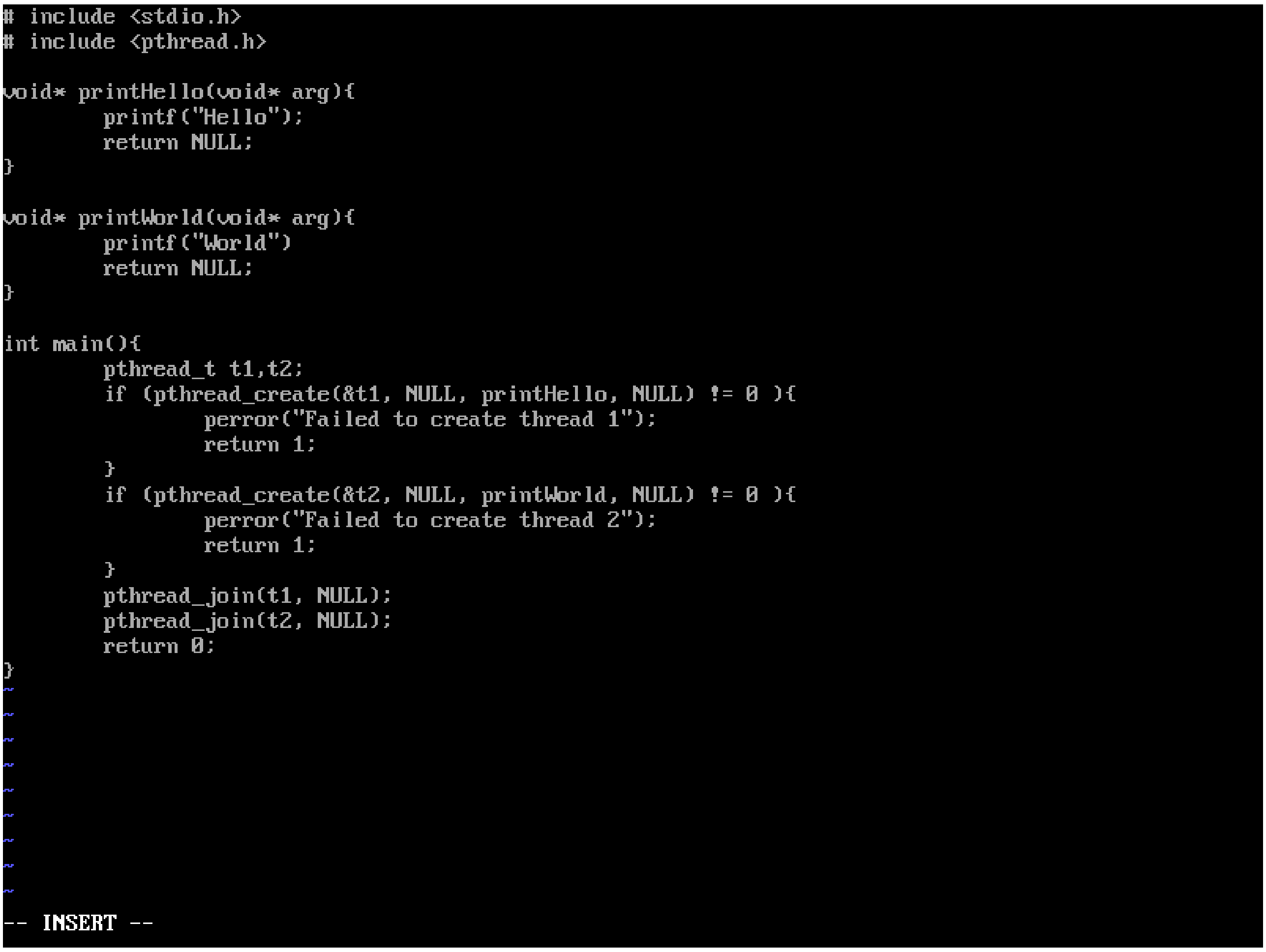
yyhhzz365 前两次作业 6fd96e6 · 20 hours ago

hw1 前两次作业 20 hours ago

hw2 前两次作业 20 hours ago

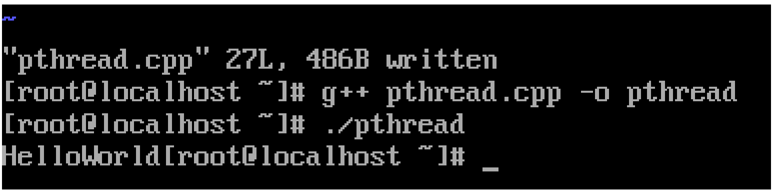
二：多线程测试

1. 打开openeuler
2. 输入`vi pthread.cpp`开始编写代码



```
# include <stdio.h>
# include <pthread.h>

void* printHello(void* arg){
        printf("Hello");
        return NULL;
}

void* printWorld(void* arg){
        printf("World")
        return NULL;
}

int main(){
        pthread_t t1,t2;
        if (pthread_create(&t1, NULL, printHello, NULL) != 0 ){
                perror("Failed to create thread 1");
                return 1;
        }
        if (pthread_create(&t2, NULL, printWorld, NULL) != 0 ){
                perror("Failed to create thread 2");
                return 1;
        }
        pthread_join(t1, NULL);
        pthread_join(t2, NULL);
        return 0;
}
~
~
~
~
~
~
~
~
~
-- INSERT --
```

3. 保存编译后运行，输出



```
~
"pthread.cpp" 27L, 486B written
[root@localhost ~]# g++ pthread.cpp -o pthread
[root@localhost ~]# ./pthread
HelloWorld[root@localhost ~]# _
```