

OpenEuler配置git并编译pthread代码

一、检查和配置git

1.检查git环境

```
git --version
```

2.全局配置git仓库

```
git config --global user.name "GYF"  
git config --global user.email "gyf2528357407@foxmail.com"
```

```
[root@GYF Desktop]# git --version  
git version 2.43.0  
[root@GYF Desktop]# git config --global user.name "GYF"  
[root@GYF Desktop]# git config --global user.email "gyf2528357407@foxmail.com"
```

3.检查刚刚配置的git环境

```
git config --global --list
```

```
[root@GYF Desktop]# git config --global --list  
user.name=GYF  
user.email=gyf2528357407@foxmail.com
```

4.获取ssh秘钥并且查看生成的公钥

```
ssh-keygen -t rsa -b 4096 -C "gyf2528357407@foxmail.com"  
cat ~/.ssh/id_rsa.pub
```

5.点击github的setting配置ssh并在github上新建一个仓库并且clone到本地

```
git clone git@github.com:Peppap12138/OS_HW.git
```

```
[root@GYF Desktop]# git clone git@github.com:Peppap12138/OS_HW.git  
正克隆到 'OS_HW'...  
警告：您似乎克隆了一个空仓库。
```

二、代码编写

1.移动到OW_HW仓库

```
cd OS_HW
```

2.移动之前的文件到新文件夹

```
mkdir hw_os2
mv ../../openEuler_school_practice/week3_work/fork1.c ./
mv ../../openEuler_school_practice/week3_work/fork2.c ./
```

3.重新编译两个文件

```
gcc fork1.c -o fork1
gcc fork2.c -o fork2
```

4.编写pthread.c

```
cd ../
mkdir hw_os3
touch pthread.c
vim pthread.c
```

然后就是愉快地代码编写时间

```
#include <pthread.h>
#include <stdio.h>
void *worker(void *arg)
{
    printf("Hello World!\n");
    return NULL;
}
int main(int argc, char *argv[])
{
    pthread_t tid;

    if (pthread_create(&tid, NULL, worker, NULL))
    {
        printf("Cannot create thread\n");
        exit(1);
    }

    printf("Main waits for thread to finish...\n");

    pthread_join(tid, NULL);

    exit(0);
}
```

"pthread.c" [noeol] 23L, 389B



1,1

全部

5.编译并运行pthread

```
gcc pthread.c -o pthread
./pthread
```

```
[root@GYF hw_os3]# ./pthread
Main waits for thread to finish...
Hello World!
```

三、上传代码至git仓库

```
cd ../
git add .
git commit -m "OS第二三次作业代码"
git push
```

```
[root@GYF OS_HW]# git add .
[root@GYF OS_HW]# git commit -m "OS第二三次作业代码"
[main (根提交) 3203129] OS第二三次作业代码
6 files changed, 59 insertions(+)
create mode 100755 hw_os2/fork1
create mode 100644 hw_os2/fork1.c
create mode 100755 hw_os2/fork2
create mode 100644 hw_os2/fork2.c
create mode 100755 hw_os3/pthread
create mode 100644 hw_os3/pthread.c
```

```
[root@GYF OS_HW]# git push
枚举对象中: 10, 完成.
对象计数中: 100% (10/10), 完成.
使用 4 个线程进行压缩
压缩对象中: 100% (10/10), 完成.
写入对象中: 100% (10/10), 5.31 KiB | 5.31 MiB/s, 完成.
总共 10 (差异 2), 复用 0 (差异 0), 包复用 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:Peppia12138/OS_HW.git
 * [new branch]      main -> main
```

四、总结

本次作业在OpenEuler上配置了git，并且成功提交了pthread代码，初步了解了pthread函数的作用