作业：MQTT实现命令控制

> 学号：21951159
>
> 姓名：邹佳坤
>
> Tips：源码在src目录下，cmake可以直接编译，第三方MQTT库使用paho.mqtt.c

MQTT原来的通信方式是：

1. 服务器启动，等待转发消息
2. 订阅端向服务器订阅某个话题，并传入自己的id，等待话题推送
3. 发布端根据话题向服务器发布消息
4. 服务端收到发布端发布的消息，根据话题将消息推送到订阅该话题的客户端
5. 客户端收到消息，一次交互完成！

可以看到，MQTT的这种通信模式其实是“单向通信”，即同时只能服务端向客户端发送消息的，而如果要实现命令控制的话，得实现以下流程：

- 被控制端首先订阅“command”话题
- 控制端在“command”话题下发布控制命令消息
- 被控制端收到命令消息，根据命令消息进行命令控制
- 同时，控制端订阅了“return”话题
- 被控制端执行完命令后，向了“return”话题发布执行结果的消息
- 控制端收到执行结果，控制完成

EXAMPLE

下面给出一个例子：

控制端：

```
#include "base.h"

int main(int argc, char **argv) {
    char message[1000000];
    send_command(argv[1], TOPIC1, "client");    // 省略实现细节，详细查看源码文件
base.h
    recv_command(message, TOPIC2, "client");    // 省略实现细节，详细查看源码文件
base.h
    printf("%s\n", message);
    return 0;
}
```

被控制端：

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

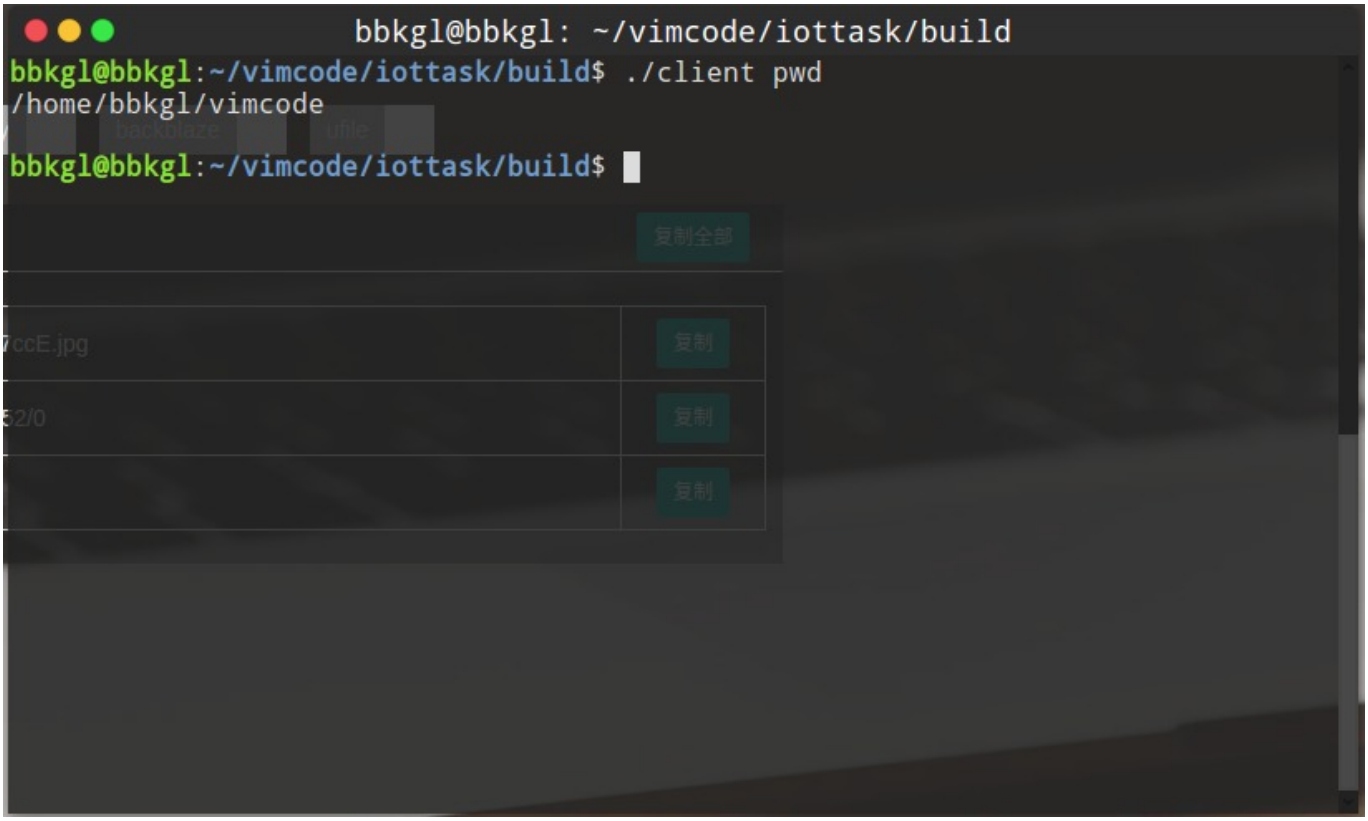
```
#include <cstring>
#include "base.h"

int main() {
    char message[100];
    char command[100];
    char info[1000000];
    recv_command(message, TOPIC1, "server");    // 省略实现细节，详细查看源码文件
base.h
    sprintf(command, "%s 1> a.txt", message);
    system(command);
    usleep(100000);
    int fd = open("a.txt", O_RDONLY);
    int len = -1;
    while (len) {
        char buff[1024] = {'\0'};
        len = read(fd, buff, sizeof(buff));
        strcat(info, buff);
    }
    printf("%s\n", info);
    send_command(info, TOPIC2, "server");    // 省略实现细节，详细查看源码文件base.h
    close(fd);
    system("rm a.txt");
    return 0;
}
```

这里有个细节，被控制端收到命令后，阻塞一段时间后才继续，这是因为在往某个话题发布消息前，必须已经有客户端先订阅了该话题，不然消息会发不出去，客户端也接收不到。

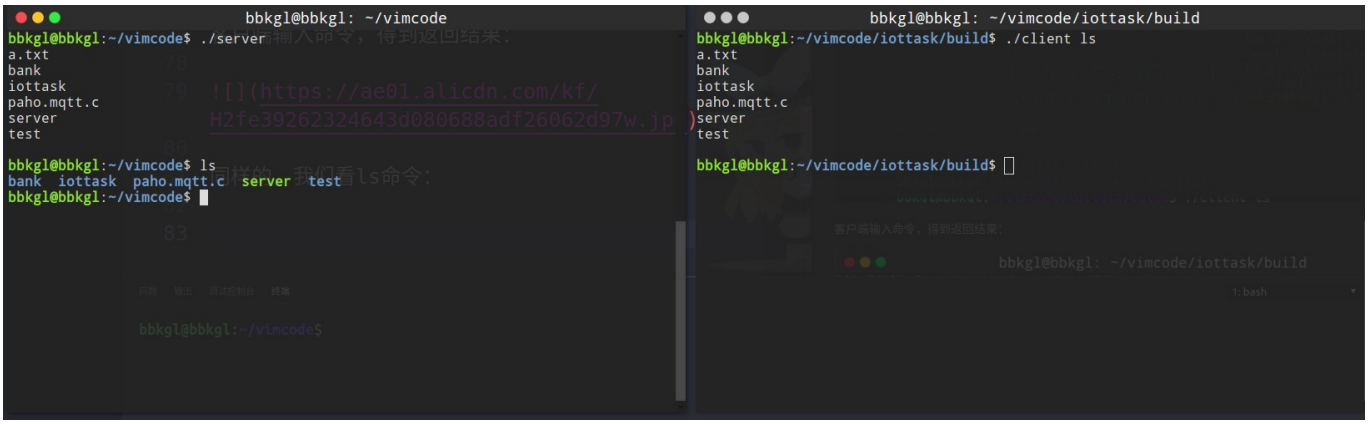
效果

启动服务端，注意服务端所在路径：

客户端输入命令，得到返回结果：



同样的，我们看ls命令：



可以看到，客户端返回了服务端所在目录下的文件和子目录。