

bash script 阅读仿写题

bash 是一种在 GNU/Linux 中常用的脚本语言。阅读以下材料，并完成题目。

1

重庆大学开源软件镜像站（以下简称“镜像站”）的文档主要以 markdown 格式托管在 GitHub 上以便于协作，而当文档在 GitHub 上被修改时，GitHub 会以某种方式通知镜像站服务器将修改后文档部署到镜像站上。

当服务器收到通知后，会运行一个 bash 脚本来完成部署操作，以下为一个简化了的示例：

```
echo "开始部署"  
cd doc  
git pull  
hexo generate  
cp -rf public/* /var/www/wiki/  
echo "完成"
```

以上脚本每一行都是一条指令，运行时会按照顺序逐行依次执行，以下为各行代码的含义：

1. 输出文本“开始部署”
2. 切换到当前目录下的 `doc` 目录（该目录已被提前创建好并存有修改前的文档）
3. 将当前目录下的文档更新到最新版
4. 生成相应的网页（默认保存至 `public` 文件夹）
5. 将 `public` 下的生成好的网页部署到镜像站的 http 服务器上
6. 输出文本“完成”

若

- 镜像站服务器保存文档的目录不是 `doc`，而是 `docs`
- 开始部署和完成部署时输出英文文本而不是中文文本

请根据材料写出同时符合上述两个要求的 bash 脚本（答案不唯一）

2

上一问材料中的脚本可改为：

```
echo "开始部署"
cd doc
if git pull
then
    hexo generate
    cp -rf public/* /var/www/wiki/
    echo "完成"
else
    echo "pull 失败"
fi
```

该脚本在原脚本的基础上添加了一个错误处理功能：运行至 `if git pull` 一行时，`bash` 会运行 `if` 后的 `git pull` 指令，然后

- 若运行成功，脚本跳转至 `then` 处继续执行直到 `else` 处，而 `else` 到 `fi` 之间的指令不会被执行。
- 若运行失败，`then` 和 `else` 之间的内容不会被执行，脚本会跳转至 `else` 处执行 `else` 和 `fi` 之间的指令，即输出文本"pull 失败"。

若

- 文档使用 `hugo` 而不是 `hexo` 来生成网页（对应生成网页的指令为 `hugo` 而非 `hexo generate`）
- 生成网页若失败，则输出提示错误发生的文本，并且不再继续执行将网页部署至 `http` 服务器的操作
- 生成网页若成功，则继续完成部署操作

请根据材料，基于**第 1 问材料中的脚本**写出同时符合上述三个要求的 `bash` 脚本（答案不唯一）

