



山东大学 计算机科学与技术 学院

云计算技术 课程实验报告

学号：201918202206	姓名：杨逸	班级：20 级计科二班
实验题目：利用云平台搭建个人博客		
实验学时：2	实验日期：2023. 03. 14	
<p>实验目的：熟悉个人博客系统的搭建。</p> <p>具体包括：</p> <p>参考方案：注册 Github 账号，搭建 Hexo 环境并实现个人博客搭建，撰写实验报告。</p>		
<p>硬件环境：</p> <p>联网的计算机一台</p>		
<p>软件环境：</p> <p>Windows or Linux</p>		
<p>实验步骤与内容：</p> <p>一、Hexo 搭建步骤</p> <p>1. 安装 Git。</p> <p>为了把本地的网页文件上传到 github 上面去，需要用到工具 Git。在 Windows 环境下需要在到 git 官网下载 .exe 文件。</p>  <p>安装选项选择默认安装即可，最终安装完成运行情况如下。</p> 		

## 进入 cmd 下查看 git 环境

```
C:\Users\86159>git --version
git version 2.40.0.windows.1
```

## 2. 安装 node.js

由于 Hexo 是基于 node.js 编写的,所以需要安装一下 node.js 和里面的 npm 工具。去 node.js 官网下载 Windows 安装包,同样默认安装即可。



## 进入 cmd 下查看 node.js 环境

```
C:\Users\86159>node -v
v16.19.1
```

```
C:\Users\86159>npm -v
8.19.3
```

## 3. 安装 Hexo

在新文件夹 myblog 中存放自己的博客文件。在该文件夹下打开 git bash, 输入 npm install -g hexo-cli 完成 hexo 的安装。并用 hexo -v 判断是否安装成功。

```
86159@DESKTOP-57U1STV MINGW64 /f/MyBlog
$ npm install -g hexo-cli

changed 59 packages, and audited 60 packages in 3s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
MINGW64:/f/MyBlog
86159@DESKTOP-S7UISTV MINGW64 /f/MyBlog
$ hexo -v
hexo-cli: 4.3.0
os: win32 10.0.19044
node: 16.19.1
v8: 9.4.146.26-node.24
uv: 1.43.0
zlib: 1.2.11
brotli: 1.0.9
ares: 1.18.1
modules: 93
nghttp2: 1.47.0
napi: 8
llhttp: 6.0.10
openssl: 1.1.1t+quic
cldr: 41.0
icu: 71.1
tz: 2022f
unicode: 14.0
ngtcp2: 0.8.1
nghttp3: 0.7.0
86159@DESKTOP-S7UISTV MINGW64 /f/MyBlog
$
```

输入 `hexo init` 初始化文件夹，其包含内容如下。

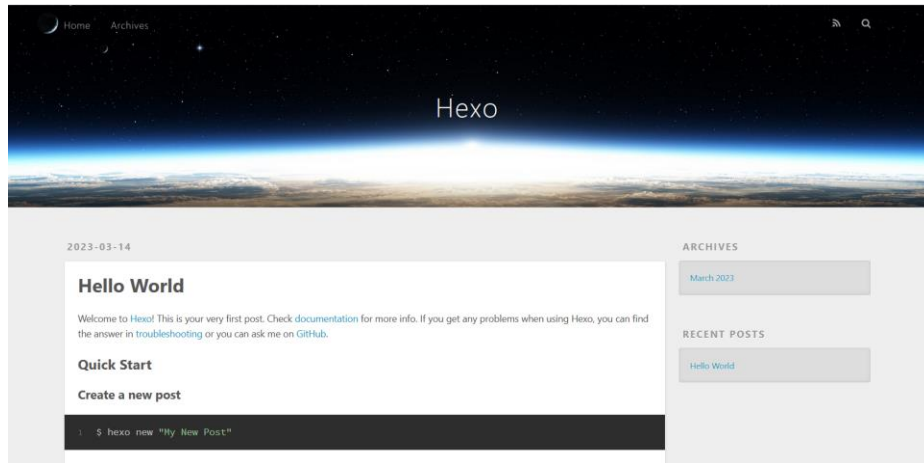
```
86159@DESKTOP-S7UISTV MINGW64 /f/MyBlog
$ hexo init
INFO Cloning hexo-starter https://github.com/hexojs/hexo-starter.git
INFO Install dependencies
INFO Start blogging with Hexo!
```

.github	2023/3/14 20:14	文件夹	
node_modules	2023/3/14 20:14	文件夹	
scaffolds	2023/3/14 20:14	文件夹	
source	2023/3/14 20:14	文件夹	
themes	2023/3/14 20:14	文件夹	
.gitignore	2023/3/14 20:14	文本文档	1 KB
_config.landscape.yml	2023/3/14 20:14	Yaml 源文件	0 KB
_config.yml	2023/3/14 20:14	Yaml 源文件	3 KB
package.json	2023/3/14 20:14	JSON File	1 KB
package-lock.json	2023/3/14 20:14	JSON File	166 KB

输入 `hexo g` 生成静态网页，然后输入 `hexo s` 打开本地服务器。

```
86159@DESKTOP-S7UISTV MINGW64 /f/MyBlog
$ hexo g
INFO Validating config
INFO Start processing
INFO Files loaded in 128 ms
INFO Generated: archives/2023/index.html
INFO Generated: archives/index.html
INFO Generated: archives/2023/03/index.html
INFO Generated: js/script.js
INFO Generated: index.html
INFO Generated: css/fonts/fontawesome-webfont.woff2
INFO Generated: fancybox/jquery.fancybox.min.css
INFO Generated: js/jquery-3.4.1.min.js
INFO Generated: fancybox/jquery.fancybox.min.js
INFO Generated: css/style.css
INFO Generated: css/fonts/fontawesome.otf
INFO Generated: css/fonts/fontawesome-webfont.eot
INFO Generated: css/fonts/fontawesome-webfont.ttf
INFO Generated: css/fonts/fontawesome-webfont.woff
INFO Generated: css/images/banner.jpg
INFO Generated: 2023/03/14/hello-world/index.html
INFO Generated: css/fonts/fontawesome-webfont.svg
INFO 17 files generated in 307 ms
```

```
86159@DESKTOP-57UISTV MINGW64 /f/MyBlog
$ hexo s
INFO Validating config
INFO Start processing
INFO Hexo is running at http://localhost:4000/ . Press Ctrl+C to stop.
```

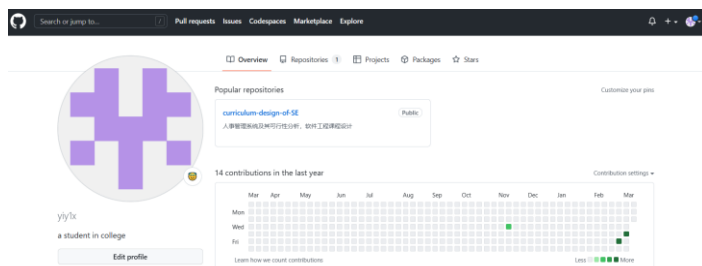


按 ctrl+c 关闭本地服务器。

```
INFO Start processing
INFO Hexo is running at http://localhost:4000/ . Press Ctrl+C to stop.
INFO Have a nice day
```

```
86159@DESKTOP-57UISTV MINGW64 /f/MyBlog
$
```

4. 注册 Github 账号创建个人仓库  
注册过程比较容易，最终注册账号如下。




新建一个项目仓库 New repository。

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner \* Repository name \*

 yiy1x / yiy1x.github.io ✓

Great repository name yiy1x.github.io is available. ble. Need inspiration? How about [laughing-fishstick?](#)

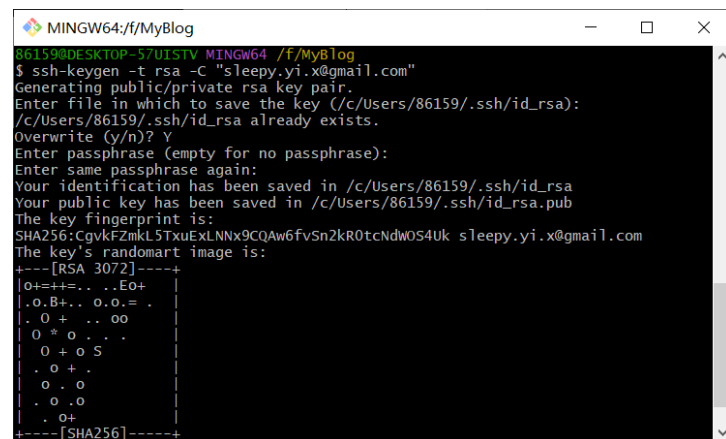
## 5. 生成 SSH 添加到 GitHub

生成 SSH 添加到 GitHub，连接 Github 与本地。用 git bash 输入以下命令，进行 github 的匹配。

```
86159@DESKTOP-57UISTV MINGW64 /f/MyBlog
$ git config --global user.name "yiy1x"

86159@DESKTOP-57UISTV MINGW64 /f/MyBlog
$ git config --global user.email "sleepy.yiy.x@gmail.com"
```

最终生成秘钥如下。



```
MINGW64:/f/MyBlog
86159@DESKTOP-57UISTV MINGW64 /f/MyBlog
$ ssh-keygen -t rsa -C "sleepy.yi.x@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/86159/.ssh/id_rsa):
/c/Users/86159/.ssh/id_rsa already exists.
Overwrite (y/n)? Y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/86159/.ssh/id_rsa
Your public key has been saved in /c/Users/86159/.ssh/id_rsa.pub
The key's fingerprint is:
SHA256:CgvkFZmkL5TxuExLNNx9CQAw6fvSn2kR0tcNdW0S4Uk sleepy.yi.x@gmail.com
The key's randomart image is:
+---[RSA 3072]-----+
|o+=+=...Eo+
|.o.B+..o.o.=.
|.O+..oo
|o*o...
|o+oS
|.o+.
|.o.o
|.o.o
|.o+
+---[SHA256]-----+
```


将 .ssh 文件夹中内容复制，并在 github 上新建一个 SSH，将该文件夹中的公钥信息复制进去。

### SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

#### Authentication Keys



**test1**  
SHA256:CgvkFZmkL5TxuExLNNx9CQAw6fvSn2kR0tcNdW0S4Uk  
Added on Mar 14, 2023  
Never used — Read/write

Delete

输入 `ssh -T git@github.com`，最后显示用户名，成功。

```
86159@DESKTOP-57UISTV MINGW64 /f/MyBlog
$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhpZisF/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 yiy1x! You've successfully authenticated, but GitHub does not provide shell access.
```

## 6. 将 hexo 部署到 GitHub

打开\_config.yml 文件，将最后部分修改如下。

```
deploy:
  type: git
  repository: https://github.com/yiy1x/yiy1x.github.io.git
  branch: master
```

用 `npm install hexo-deployer-git --save` 安装 deploy-git, 将命令部署到 github 上。

```
86159@DESKTOP-57U1STV MINGW64 /f/MyBlog
$ npm install hexo-deployer-git --save

added 3 packages, and audited 243 packages in 3s

22 packages are looking for funding
  run `npm fund` for details

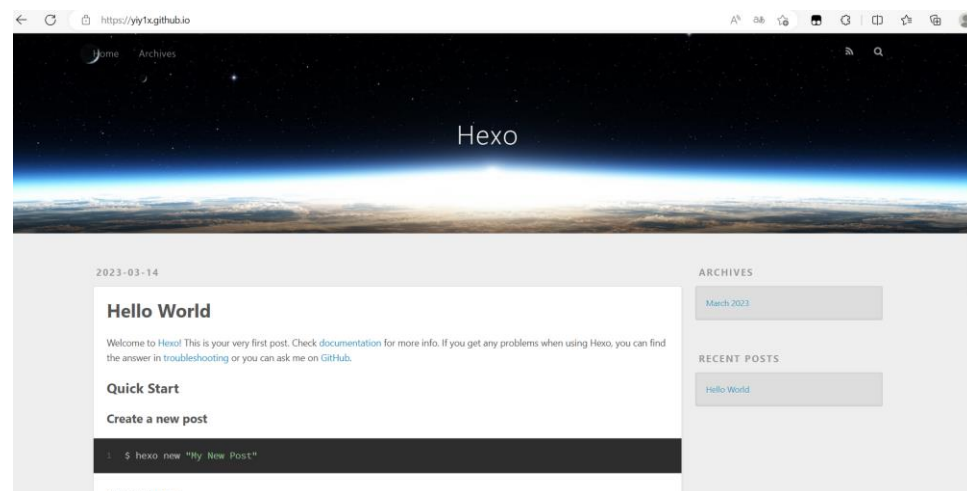
found 0 vulnerabilities
```

用 `hexo g` 生成静态文章，`hexo d` 部署文章。

```
86159@DESKTOP-57U1STV MINGW64 /f/MyBlog
$ hexo g
INFO Validating config
INFO Start processing
INFO Files loaded in 86 ms
INFO 0 files generated in 19 ms

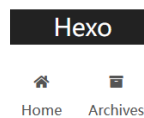
86159@DESKTOP-57U1STV MINGW64 /f/MyBlog
$ hexo d
INFO Validating config
INFO Deploying: git
INFO Setting up Git deployment...
Initialized empty Git repository in F:/MyBlog/.deploy_git/.git/
[master (root-commit) 4c8fd61] First commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 placeholder
INFO Clearing .deploy_git folder...
INFO Copying files from public folder...
INFO Copying files from extend dirs...
```

连接完毕后，打开效果如下



可以通过更改主题使其有更精美的格式，也可以向其上传文件、博客内容，每次更新都需要重新 g 生成，d 上传，之后可以从网页访问得到。下图就是使用 next 格式，并上传了我的汇编作业后的博客页面。

```
86159@DESKTOP-57UISTV MINGW64 /f/MyBlog/themes
$ git clone https://github.com/theme-next/hexo-theme-next themes/next
Cloning into 'themes/next'...
remote: Enumerating objects: 12582, done.
remote: Total 12582 (delta 0), reused 0 (delta 0), pack-reused 12582
Receiving objects: 100% (12582/12582), 8.04 MiB | 4.26 MiB/s, done.
Resolving deltas: 100% (7987/7987), done.
```



201918202206-计机二班-杨逸-10号作业

9.1解：AH=1是从键盘输入一个字符并回显到屏幕上，AH=8是从键盘输入一个字符，不回显，检查Ctrl-Break。

9.7解：代码如下：

```
1 ;课本作业9.7
2 ;显示一段信息
```

## 二、实验报告 PDF 版的上传

用同样的方法上传实验报告到网站上。由于实验报告为 PDF 版本，发现 hexo 并未显示实验报告，因此需要安装 hexo-pdf。

```
86159@DESKTOP-57UISTV MINGW64 /f/MyBlog
$ npm install --save hexo-pdf
npm WARN deprecated ejs@1.0.0: Critical security bugs fixed in 2.5.5
added 2 packages, and audited 245 packages in 3s

22 packages are looking for funding
  run `npm fund` for details

2 critical severity vulnerabilities

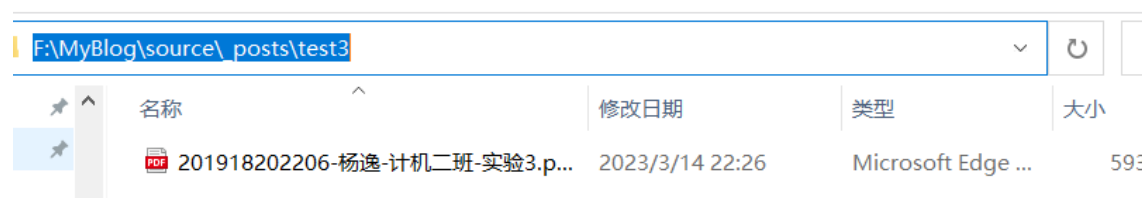
To address all issues, run:
  npm audit fix

Run `npm audit` for details.
```

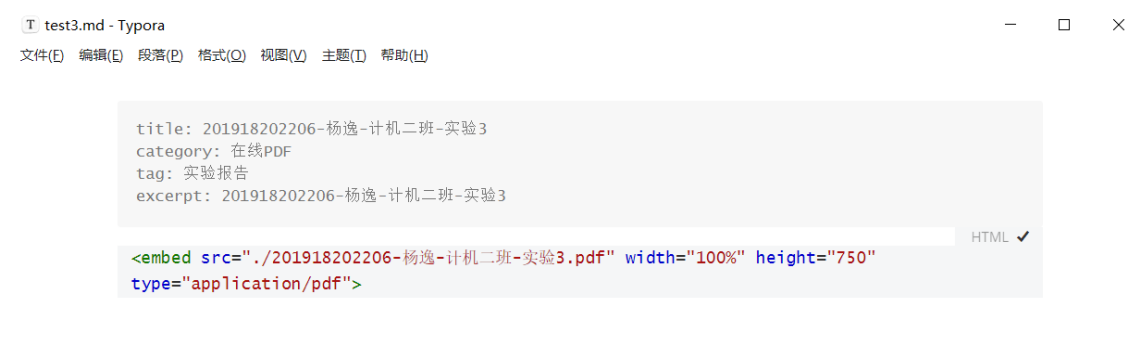
将\_config.yml 的 post\_asset\_folder 属性修改为 true。

```
! _config.yml X
F: > MyBlog > ! _config.yml
41 filename_case: 0
42 render_drafts: false
43 post_asset_folder: true
44 relative_link: false
45 future: true
46 highlight:
47   enable: true
```

在 source 文件夹中创建新文件目录（此处为 test3）。



在 /source/\_posts 创建同名的 md 文件，存放如下内容。其中 embed src 指明了 pdf 文件的路径，由于 md 创建的是和文件夹同名的文件，因此不需要用 test3 前缀，test3.md 会默认寻找 test3 文件夹下的文件。



用 hexo g 和 hexo d 上传到博客上。最终情况如下图。





- 结论分析与体会:
- Hexo 是一款基于 Node.js 的静态博客框架, 依赖少易于安装使用, 可以方便的生成静态网页托管在 GitHub 和 Coding 上, 是搭建博客的首选框架。
  - Git 是目前世界上最先进的分布式版本控制系统, 可以有效、高速的处理从很小到非常大的项目版本管理。而关于 git 的下载, 对于 Windows 需要按照上述实验步骤下载, 但是对于 linux 环境下只需在终端中输入 `sudo apt-get install git` 即可完成安装。这是因为最早的 git 就是在 linux 上编写的。
  - Hexo 初始化后, 生成了很多文件, 其中部分文件的作用如下。`node_modules`: 依赖包, 包含 node.js 各种库的目录; `public`: 存放生成的页面; `scaffolds`: 生成文章的一些模板; `source`: 存放着文章、各类页面、图像等文件; `themes`: 存放着主题文件, 一般也用不到; `_config.yml`: 博客的配置文件。
- | 文件名称                  | 时间              | 文件类型      | 大小     |
|-----------------------|-----------------|-----------|--------|
| .github               | 2023/3/14 20:14 | 文件夹       |        |
| node_modules          | 2023/3/14 20:14 | 文件夹       |        |
| scaffolds             | 2023/3/14 20:14 | 文件夹       |        |
| source                | 2023/3/14 20:14 | 文件夹       |        |
| themes                | 2023/3/14 20:14 | 文件夹       |        |
| .gitignore            | 2023/3/14 20:14 | 文本文档      | 1 KB   |
| _config.landscape.yml | 2023/3/14 20:14 | Yaml 源文件  | 0 KB   |
| _config.yml           | 2023/3/14 20:14 | Yaml 源文件  | 3 KB   |
| package.json          | 2023/3/14 20:14 | JSON File | 1 KB   |
| package-lock.json     | 2023/3/14 20:14 | JSON File | 166 KB |
- 要创建一个和你用户名相同的仓库, 后面加. [http://github.io, 只有这样, 将来部署到 GitHub page 的时候, 才会被识别, 也就是 http://xxxx.github.io, 其中 xxx 就是注册 GitHub 的用户名。

5. ssh, 简单来讲, 就是一个秘钥。其中, id\_rsa 是你这台电脑的私人秘钥, id\_rsa.pub 是公共秘钥。把公钥放在 GitHub 上, 这样当你链接 GitHub 自己的账户时, 它会根据公钥匹配你的私钥, 当能够相互匹配时, 才能够顺利的通过 git 上传你的文件到 GitHub 上。

6. 常用 hexo 命令如下:

hexo n "postName" #新建文章

hexo n page "pageName" #新建页面

hexo g #生成静态页面至 public 目录

hexo s #开启预览访问端口 (默认端口 4000, 'ctrl + c' 关闭 server)

hexo d #部署到 GitHub

7. 为什么用 hexo d 部署的时候会出现“warning: LF will be replaced by CRLF”?

```
$ hexo d
INFO Validating config
INFO Deploying: git
INFO Clearing .deploy_git folder...
INFO Copying files from public folder...
INFO Copying files from extend dirs...
warning: in the working copy of '2023/03/14/201918202206-计机二班-杨逸-10号作业/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '2023/03/14/hello-world/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'archives/2023/03/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'archives/2023/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'archives/index.html', LF will be replaced by CRLF the next time Git touches it
```

答: (1) LF: Line Feed 换行; CRLF: Carriage Return Line Feed 回车换行; 在过去的机械打字机上有有个部件叫“字车”。每打一个字符, 字车前进一格, 打完一行后, 我们需要让字车回到起始位置, 称为“Carriage Return”。

(2) 在 Windows 平台中一些软件会生成换行是 LF 的代码文本, 而 Windows 下是使用 CRLF 来结束一行。(Unix 和 Max 用的则是 LF)。

(3) git 中负责处理换行的变量是 core.autocrlf, 当“core.autocrlf=true”时: 如果把 modified 修改过的文件 git add 到暂存区 stage, Git 自动把 LF 转换成 CRLF, 并给出那条警告“LF will be replaced by CRLF”。

(4) Windows 出现该警告并不影响工作, 而且它能保证我们项目团队正常跨系统 git 操作代码: Windows 上的检出文件中会保留 CRLF, 而在 Mac 和 Linux 上, 以及版本库中会保留 LF。

8. 上传到博客的 PDF 有两种格式:

(1) <!-- {% pdf ./201918202206-杨逸-计机二班-实验 3.pdf.pdf %} -->

(2) <embed src="./201918202206-杨逸-计机二班-实验 3.pdf" width="100%" height="750" type="application/pdf">

两种格式均可, 但是方法二可调整 pdf 显示的大小, 比较灵活。