

---

# wukong-simulator

发布 **v0.2**

熊鸿康

2022 年 05 月 30 日



---

## Contents

---

<b>1</b>	<b>Workflow 介绍</b>	<b>3</b>
1.1	文档	3
1.2	测试	8
1.3	代码规范	8
1.4	Usage	8



wukong-simulator 是用于开发 wukong-firmware 的模拟器，其中包含了 NVMe Controller 和 Flash Controller 的寄存器级别模拟，可以通过连接 fio 作为 host 进行测试。

---

注解: This project is under active development.

---



## 1.1 文档

### 1.1.1 NEW-TO-Sphinx

我们使用 Sphinx 来生成文档, 其中 Sphinx 使用的一种叫 reStructuredText 标记语言。

Sphinx 可以自动将文档生成 pdf, epub, html 等不同格式, 也能进行文档的一键式部署, 这些特性使得其广受开源社区欢迎, 包括 Linux, fio 等开源项目均采用 Sphinx 作为文档编写工具。

#### Sphinx 安装

当前文档使用 Sphinx 4.5.0 或者更高版本生成。

安装 Sphinx, 可以使用 pip 进行安装, 方便管理 python 包。

```
pip install -U Sphinx
```

注解: 推荐在 python 的虚拟机 venu 中进行安装, 可以更加方便地移植, 备份你的 python 环境

#### Sphinx 依赖的 python 包

使用 pip 检查依赖包

```
pip install /Documentation/requirements.txt
```

警告: 目前需要的 python 包仅有 furo, 需要自己用 pip 安装一下

我还在尝试修改一个 perl 脚本 scripts/sphinx-pre-install 来实现自动检查 Sphinx 版本, 以及需要安装的包, 但似乎目前时间上还不够充裕

## html 文档输出

```
cd Documentation
make html
```

## pdf 文档输出

---

注解: 本项目输出的 pdf 中的中文使用的是 free open source 的 WenQuanYi Zen Hei

---

---

注解: 本人使用的是 Ubuntu 20.04 LTS+aliyun mirror 可能因为系统和镜像不同, 需要安装的 tex 工具会不一样, 可以根据 make latexpdf 的输出自行安装需要的工具, 此外, 还需要安装一下自己缺少的字体, 下面是本人的环境安装流程, 可以参考一下

---

警告: 强烈建议 20 分钟内没有 make 成功, 来找 本人 讨论一下, 也正好完善一下这部分文档

```
# tex tools
sudo apt install latexmk
sudo apt install texlive-xelatex
sudo apt install texlive

# fonts
sudo apt install fonts-freefont-otf
sudo apt install fonts-wqy-zenhei

# 安装完成后
make latexpdf
```

## Sphinx online deployment

参考 官方文档

警告: 在 Github Pages 上部署文档, 可能会碰到 css 样式失效的问题, 这是因为 Github Pages 托管在 jekyll 上, 需要加上一个空的 .nojekyll 文件

### 1.1.2 HOW-WRITE-DOCS

#### 加入新的文档

下面以创建 NVMe 为例

```
cd Documentation/source && mkdir nvme && cd nvme
echo "NVMe 文档" > index.rst
echo "======" >> index.rst
```

然后需要在起始的目录中加上这个文档

```
cd Documentation/ && vim index.rst
```

在 index.rst 末尾加入 rst 代码 (也可以自己选择加在什么位置)



Workflow介绍

```
.. toctree::
: maxdepth: 2

doc-guide/index
test/index
code-style/index
usage
```

reStructuredText 语法介绍

注解：这里只列出了部分语法，如果在文档中使用到了新的语法，请在下面 补上。

更多的语法可以在 [Sphinx 官方文档找到](#), 也可以参考 Megengine 的教程<sup>2</sup>

字体样式

注解：注意当前后有其他字符时，加空格才能生效。

斜体

斜体

```
*斜体*
.. 或者
`斜体`
```

粗体

粗体

```
**粗体**
```

等宽按钮

按钮

```
` ` 按钮 ` `
```

<sup>2</sup> megengine 的 Sphinx 教程

## 多级标题

reStructuredText 提供了类似于 Markdown 中 # 功能的多级标题，如下：

# 一级标题

## 二级标题

### 三级标题

#### 四级标题

##### 五级标题

###### 六级标题

只需要通过如下代码实现：

```
一级标题
=====

二级标题
-----

三级标题
~~~~~

四级标题
+++++

五级标题
#####

六级标题
*****
```

## 列表

## 表格

## 图片

**警告：** 注意文档中所使用的图片尽量放置在 `source/_static/images` 目录内, 特别是全局可能被使用的图片, 这是为了防止交叉引用不同文件夹下图片带来的维护困难。

一般情况下请优先使用 SVG 格式的矢量图，使用位图请权衡好图片体积和清晰度。

尽可能使用 Graphviz 或 Mermaid 语法绘制流程图。

图片文件名需要有相应的语义信息，不可使用完全随机生成的字符。

## 普通图片

插入图片 rst 代码如下

```

.. image:: gnu.png
   :height: 100px (length)
   :width: 200px (length or percentage of the current line width)
   :scale: integer percentage (the "%" symbol is optional)
   :alt: alternate text
   :align: "top", "middle", "bottom", "left", "center", or "right"
   :target: text (URI or reference name)

```

## 流程图

### Graphviz 语法

### Mermaid 语法

## 注释

## 交叉引用

## 脚注

当我们引用别人的内容，最好在文档中标出 <sup>1</sup>，如下：

Code that you wrote 6 months ago is often indistinguishable.<sup>1</sup>

代码如下：

```

Code that you wrote 6 months ago is often
indistinguishable. [1]_

.. rubric:: Footnotes

.. [1] `Beginners Guide To Docs <https://www.writethedocs.org/guide/writing/
↪beginners-guide-to-docs/#you-will-be-using-your-code-in-6-months/>`_

```

## 引用

## 其他注意事项

尽量不要在 rst 文件的 中文文本中一行话中途换行，否则可能出现多余的空格。这可能和这个模板没有特别支持中文有关。

你可以使用英文逗号来解决这个问题。

比如这句话，这样就会有一个空格。

如下：

```

.. 正确的
   尽量不要在rst文件的 *中文文本* 中一行话中途换行，
   否则可能出现多余的空格。

.. 错误的
   比如这句话，这样
   就会有一个空格。

```

<sup>1</sup> [Beginners Guide To Docs](https://www.writethedocs.org/guide/writing/beginners-guide-to-docs/#you-will-be-using-your-code-in-6-months/)

## 1.2 测试

## 1.3 代码规范

在本项目中，coding-style 采用 \*\*\*\*\*, 并且使用 clang-format 来进行 format.

### 1.3.1 llvm coding-style

### 1.3.2 clang-format

linux 的代码规范 <https://github.com/torvalds/linux/blob/master/.clang-format#L71>

### 1.3.3 clangd

## 1.4 Usage

### 1.4.1 Installation

To use Lumache, first install it using pip:

```
(.venv) $ pip install lumache
```

`typedef std::vector<int> CustomList`

A typedef-like declaration of a type.

Cross reference to *CustomList*.

Check out the *Usage* section for further information, including how to *install* the project.