

openEuler Jailhouse 文档实践

目录

● 什么是 Jailhouse

Jailhouse 简介

场景示例

● openEuler 社区中的 Jailhouse

● Jailhouse ~ Tools

Jailhouse-GUI

● Jailhouse ~ Docs

● Docs ~ Tools

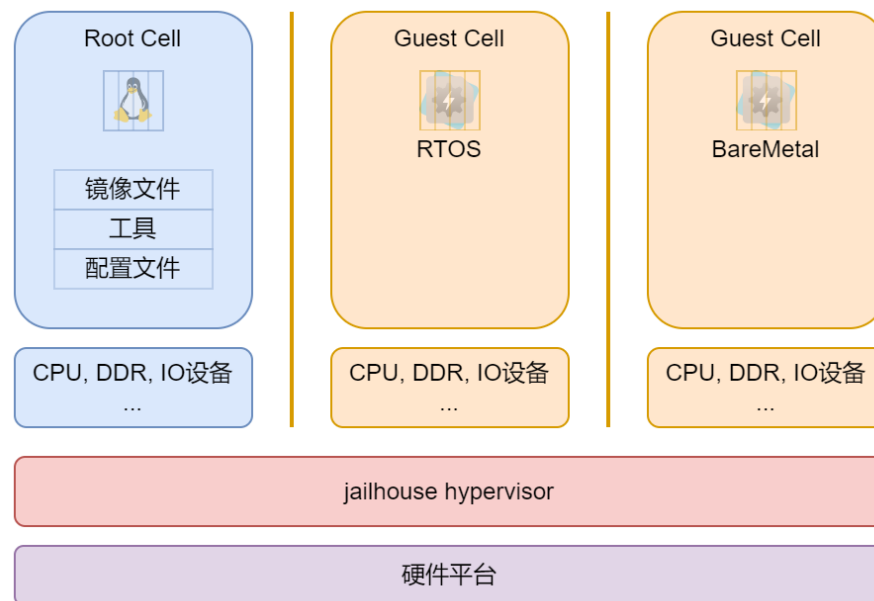
Gitee + Markdown

Read the Docs + reStructuredText

● G11 & Jailhouse

什么是 Jailhouse

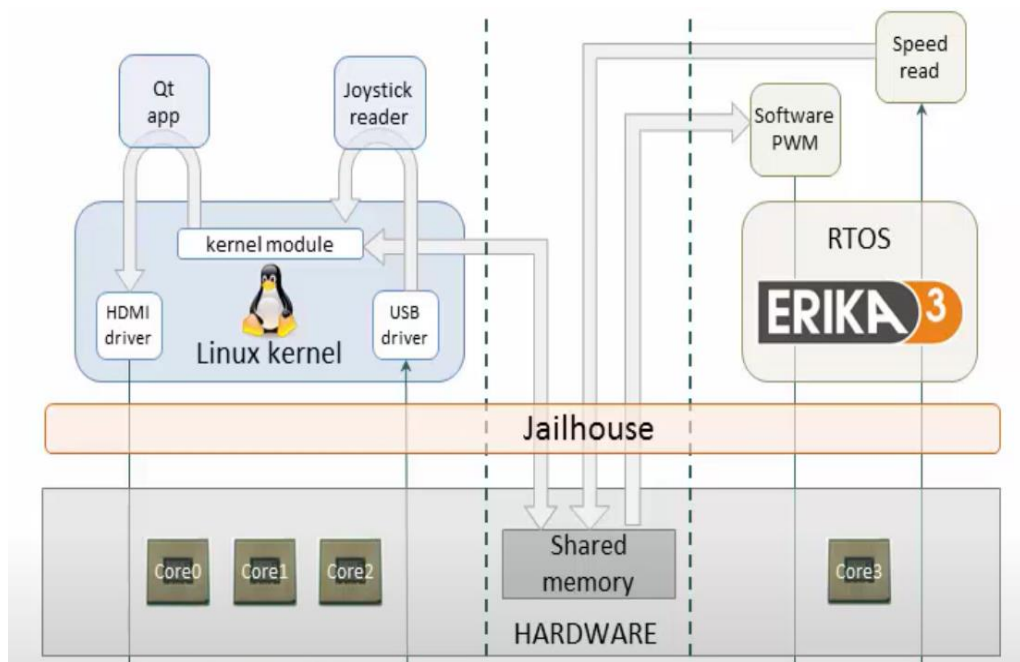
- 基于 Linux 的Type1型半虚拟化技术，它可以将一个多核处理器分割成多个虚拟硬件环境，为每个硬件环境（cell）提供独立的硬件资源，如内存，IO设备和中断信号等。
- 设计目标：为工业控制，汽车、航空航天等嵌入式场景提供高实时、可靠和安全的虚拟化环境。



What is Jailhouse

使用场景

设想单一 SOC 场景



- Linux 上积累丰富软件生态
- 但难以提供 RTOS 的实时性



Jailhouse in openEuler

src-openeuler/jailhouse

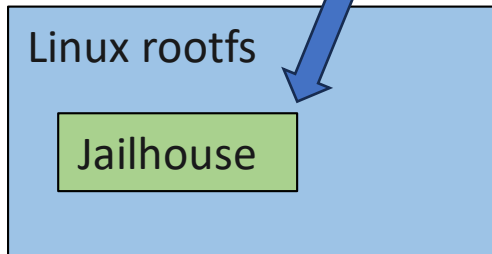
托管 Jailhouse 源码仓库



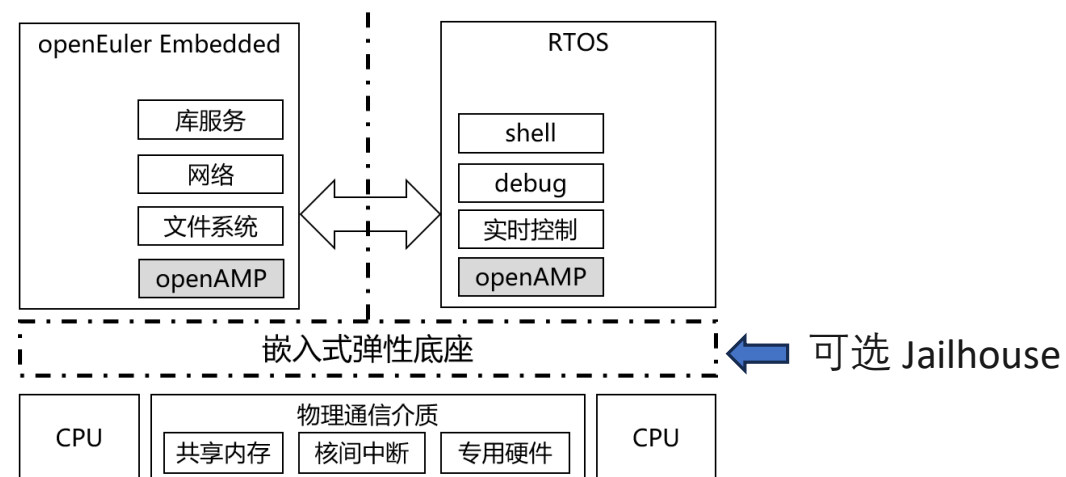
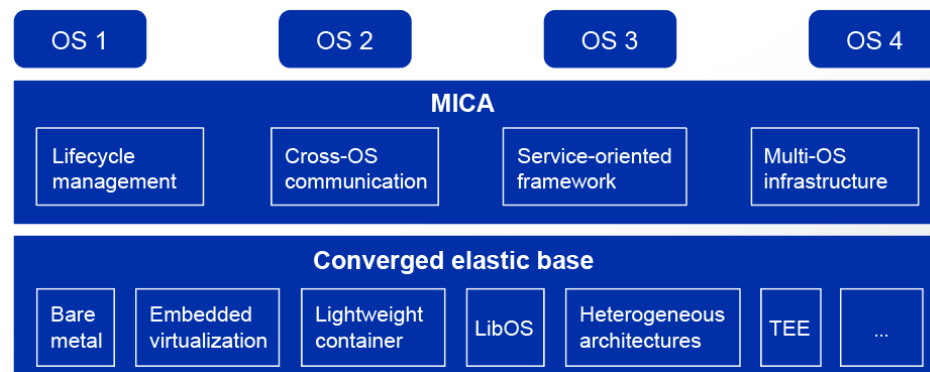
直接服务于 openEuler Embedded 社区

yocto-meta-openeuler
-> recipe-mcs/jailhouse.bb

提供构建脚本



结合 MICA 框架提供易用的混合部署



Jailhouse ~ Tools

openeuler/jailhouse-gui

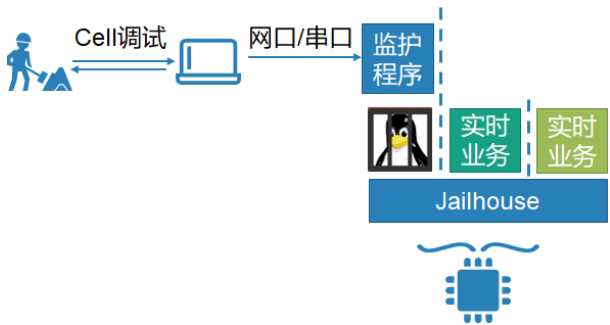


生成配置文件



```
20 struct {
21     struct jailhouse_system header;
22     u64 cpus[1];
23     struct jailhouse_memory mem_regions[12];
24     struct jailhouse_irqchip irqchips[1];
25     struct jailhouse_pci_device pci_devices[2];
26 } __attribute__((packed)) config = {
27     .header = {
28         .signature = JAILHOUSE_SYSTEM_SIGNATURE,
29         .revision = JAILHOUSE_CONFIG_REVISION,
30         .architecture = JAILHOUSE_ARM64,
31         .flags = JAILHOUSE_SYS_VIRTUAL_DEBUG_CONSOLE,
32         .hypervisor_memory = {
33             .phys_start = 0x7fc00000,
34             .size = 0x00400000,
35         },
36         .debug_console = {
37             .address = 0x09000000,
38             .size = 0x1000,
39             .type = JAILHOUSE_CON_TYPE_PL011,
40             .flags = JAILHOUSE_CON_ACCESS_MMIO |
41                 JAILHOUSE_CON_REGDIST_4,
42         },
43         .platform_info = {
44             .pci_mmconfig_base = 0x08e00000,
45             .pci_mmconfig_end_bus = 0,
46             .pci_is_virtual = 1,
47             .pci_domain = 1,
48             .arm = {
49                 .gic_version = 3,
50                 .gicd_base = 0x08000000,
51                 .gicr_base = 0x080a0000,
52                 .maintenance_irq = 25,
53             },
54         },
55         .root_cell = {
56             .name = "qemu-arm64",
57             .cpu_set_size = sizeof(config.cpus),
58             .num_memory_regions = ARRAY_SIZE(config.mem_regions),
59             .num_irqchips = ARRAY_SIZE(config.irqchips),
60             .num_pci_devices = ARRAY_SIZE(config.pci_devices),
61             .vpci_irq_base = 128-32,
62         },
63     },
64 },
65 },
66 }
```

在线维护



Jailhouse ~ Docs

仓库 README 开源软件的自我介绍

gitee

开源 企业版 高校版 私有云 Gitee AI NEW 我的

搜开源

vm_config_widget.py jailhouse:jailhouse-gui 6个月前

vm_manage_widget.py jailhouse:jailhouse-gui 6个月前

README

Jailhouse-gui使用说明

介绍

软件架构

部署jailhouse虚拟化...

编译Jailhouse-gui

Linux平台下编...

Windows平台下...

使用Jailhouse-gui

部署rpc_server...

部署rpc_server...

快速部署root cel

快速部署guest ...

参与贡献

特技

Jailhouse-gui使用说明

介绍

Jailhouse-gui用于配置和管理 Jailhouse 的图形用户界面（GUI）工具，Jailhouse 是一种基于 Linux 的管理程序，可将多核处理器分割成独立单元，用于实时关键应用程序。

软件架构

Jailhouse-gui是一个图形界面程序，由python语言开发实现，整个软件分为两个部分，一个关注业务和数据处理的模块，一个是关注交互和显示的GUI模块。整个软件的功能模块架构如下图。

gui

硬件平台信息界面

rootcell配置界面

guestcell配置界面

板级通信配置界面

PCI设备信息界面

RPC连接界面

guestcell运行界面

资源分配检查模块

提示模块

日志模块

导出模块

核心模块

生成器模块

虚拟机管理

RPC模块

硬件平台信息

资源分配对象

硬件平台管理

读取/保存

硬件平台配置文件

资源配置文件

独立 Web 网站

openEuler Embedded在线文档

Search docs

介绍与概述

总体介绍

快速上手

版本说明

常见问题

指导手册

关键特性

当前所支持的软件包

多OS混合关键性部署框架 (MICA)

外设分区管理

嵌入式ROS运行时支持

musl libc的支持

分布式软总线

软实时系统介绍

预构建工具特性

支持Rust工具链

嵌入式图形支持

ArmNN的支持

clang/llvm 编译工具链支持

轻量级虚拟化工具 Jailhouse

总体介绍

Jailhouse 构建指导

Jailhouse 使用指导

使用 Jailhouse 运行 FreeRTOS

iSula容器引擎与container os镜像

OpenBMC 镜像构建和使用

在QEMU中启动OpenBMC镜像

测试启动时长

» 关键特性 » 轻量级虚拟化工具 Jailhouse

轻量级虚拟化工具 Jailhouse

总体介绍

Jailhouse 是一种轻量级虚拟化工具，与传统的全功能虚拟化解决方案（如 KVM 和 Xen）不同，它不提供完整的虚拟机管理和抽象功能，而是一种基于Linux的静态分区虚拟化方案。Jailhouse 不支持任何设备模拟，不同客户虚拟机之间也不共享任何 CPU，所以也没有调度器。

Jailhouse 的工作是将硬件资源进行静态分区，每个分区称为一个 cell，每个 cell 之间是相互隔离的，并且拥有自己的硬件资源(CPU、内存、外设等)，运行在 cell 内的裸机应用程序或操作系统称为 inmate。Jailhouse 的第一个 cell 叫 Root Cell，这是一个特权Cell，内部运行的是一个 Linux 系统，依赖该 Linux 接管系统硬件资源，以及进行硬件的初始化和启动。除了 Root Cell 的其它 cell 统一称为 Non-root Cell，从 Root Cell 中获取系统资源，可独占或与 Root Cell 共享。

Jailhouse 构建指导

方法一：使用 oebuild 构建

openEuler Embedded 目前支持在 qemu-arm64 和 RPI4 上运行 Jailhouse，默认集成到了 openeuler-image-mcs 镜像，构建方法可参考 [mcs镜像构建指导](#)。

请注意，需要修改 oebuild 的编译配置文件 compile.yaml，把 MCS_FEATURES 中的 openamp 改成 jailhouse。

方法二：使用 MCS 镜像的 SDK 构建

按照 [mcs镜像构建指导](#) 构建出 MCS 镜像的SDK后，可以使用 SDK 交叉编译 Jailhouse，提升开发效率。步骤如下：

1. 准备 Jailhouse 源码：

下载 Jailhouse 源码：

7

开放原子开源基金会 OPENATOM FOUNDATION | OpenEuler

Docs ~ Tools

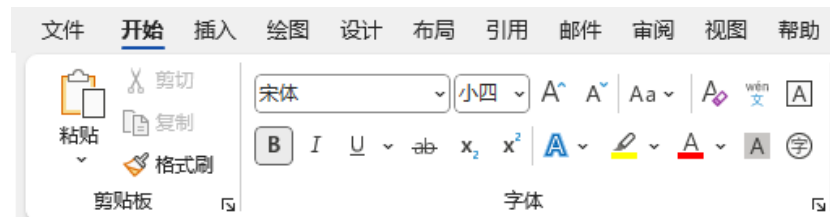
文档代码化:借鉴代码开发的方式开发文档

- **Markdown** 博客
- **Latex** 学术文献 / API文档
- **reStructureText** Python社区文档 / Linux kernel

- 易于版本管理: 可使用 git 直接文本对比改动



- 轻量化: 无需繁重的富文本客户端



- 良好的可迁移性: 文本直接可读
易于应用操作
实现web文本显示

Docs ~ Tools

1. Gitee + Markdown

2. Read the Docs + reStructuredText

Gitee 支持在线的 Markdown 浏览

大纲视图



目 README

- Jailhouse-gui使用说明
 - 介绍
 - 软件架构
 - 部署jailhouse虚拟化...
 - 编译Jailhouse-gui
 - Linux平台下编...
 - Windows平台下...
 - 使用Jailhouse-gui
 - 部署rpc_server...
 - 部署rpc_server...
 - 快速部署root cel
 - 快速部署guest ...
 - 参与贡献

仓库路径图片索引
(无需外部图库)



Jailhouse-gui使用说明

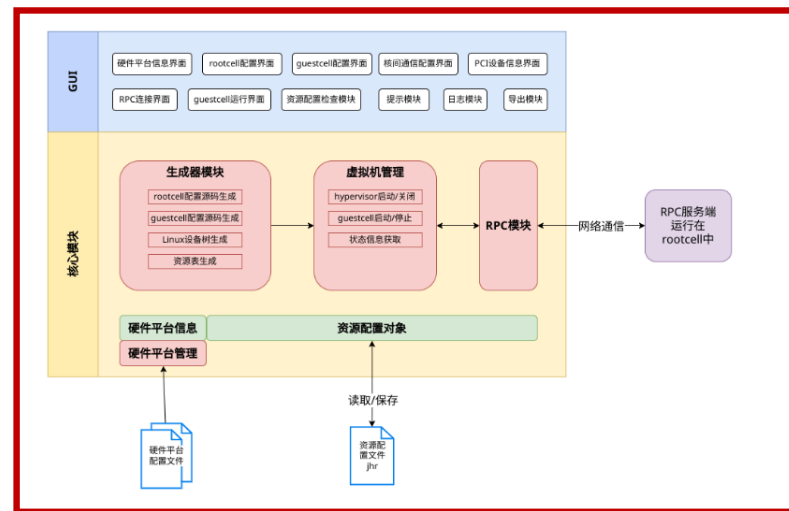
简体中文 | English

介绍

Jailhouse-gui用于配置和管理 Jailhouse 的图形用户界面 (GUI) 工具, Jailhouse 是一种基于 Linux 的管理程序, 可将多核处理器分割成独立单元, 用于实时关键应用程序。

软件架构

Jailhouse-gui是一个图形界面程序, 由python语言开发实现, 整个软件分为两个部分, 一个关注业务和数据处理的核心模块, 一个是关注交互和显示的GUI模块。整个软件的功能模块架构如下图。

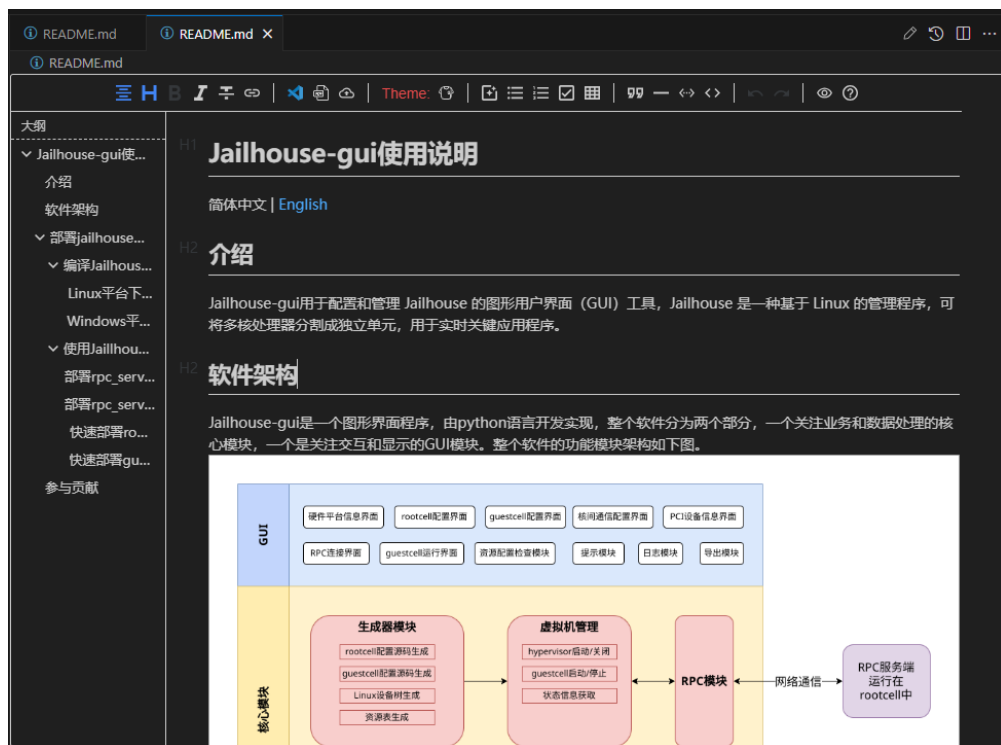


Docs ~ Tools

1. Gitee + Markdown

2. Read the Docs + reStructuredText

WYSIWYG (所见即所得)



Docs ~ Tools

1. Gitee + Markdown

2. Read the Docs + reStructuredText

本地使用 **vscode** 插件即时预览

README.md

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

① README.md > ② # Jailhouse-gui使用说明 > ③ ## 部署jailhouse虚拟化环境 > ④ ### 编译jailhouse-gui > ⑤ ### Linux平台下编译与打包

Jailhouse-gui使用说明

简体中文 | [English](./README-en.md)

介绍

Jailhouse-gui用于配置和管理 Jailhouse 的图形用户界面 (GUI) 工具, Jailhouse 是一种基于 Linux 的管理程序, 可将多核处理器分割成独立单元, 用于实时关键应用程序。

软件架构

Jailhouse-gui是一个图形界面程序, 由python语言开发实现, 整个软件分为两个部分, 一个关注业务和数据处理的核心模块, 一个是关注交互和显示的GUI模块, 整个软件的功能模块架构如下图。

![[1695286662657]](image/README/1695286662657.png)

部署jailhouse虚拟化环境

编译Jailhouse-gui

Jailhouse-gui支持在Windows和Linux平台下运行, 下面将会逐一介绍两个平台下的编译方法。

Linux平台下编译与打包

1、下载Jailhouse-gui的源码

git clone https://gitee.com/openeuler/Jailhouse-gui.git

2、安装Jailhouse-gui所需的依赖, 依赖项在requirements_linux.txt文件中。

openEuler如果没有pip工具需要自行安装, 若用户的openEuler已经安装好pip, 按照下面的步骤继续执行。

cd Jailhouse-gui

pip install -r requirements_linux.txt -i https://pypi.tuna.tsinghua.edu.cn/simple

Jailhouse-gui使用说明

简体中文 | English

介绍

Jailhouse-gui用于配置和管理 Jailhouse 的图形用户界面 (GUI) 工具, Jailhouse 是一种基于 Linux 的管理程序, 可将多核处理器分割成独立单元, 用于实时关键应用程序。

软件架构

Jailhouse-gui是一个图形界面程序, 由python语言开发实现, 整个软件分为两个部分, 一个关注业务和数据处理的核心模块, 一个是关注交互和显示的GUI模块。整个软件的功能模块架构如下图。

GUI

硬件平台信息界面

rootcell配置界面

guestcell配置界面

网络通信配置界面

PCI设备信息界面

RPC连接界面

guestcell运行界面

资源分配检查模块

提示模块

日志模块

导出模块

核心模块

生成器模块

虚拟机管理

RPC模块

RPC服务端运行在rootcell中

扩展 Markdown Preview Enhanced

Markdown Preview Enhanced

Yiyi Wang | 5,268,396 | ★★★★★(110)

Markdown Preview Enhanced ported to vscode

禁用 下载 设置

已在"SSH: 192.168.3.76"上启用扩展

细节 功能贡献 更改日志 运行时状态

Markdown Preview Enhanced

English 简体中文 繁体中文 日本語

Atom & VSCode & VSCode for the Web

LEARN VS CODE POWER USER COURSE

Supporting this project

Markdown Preview Enhanced is an open source project released under the University of Illinois/NCSA Open Source License. Its

类别

Other

扩展资源

市场 问题 仓库 许可证 Yiyi Wang

详细信息

已发布 2017-06-13, 08:43:02

上次发布 2024-03-18, 12:05:30

上次更新 2024-03-18, 12:43:32

标识符 shd11ery.markdown-preview-enhanced

Docs ~ Tools

1. Gitee + Markdown

2. Read the Docs + reStructuredText

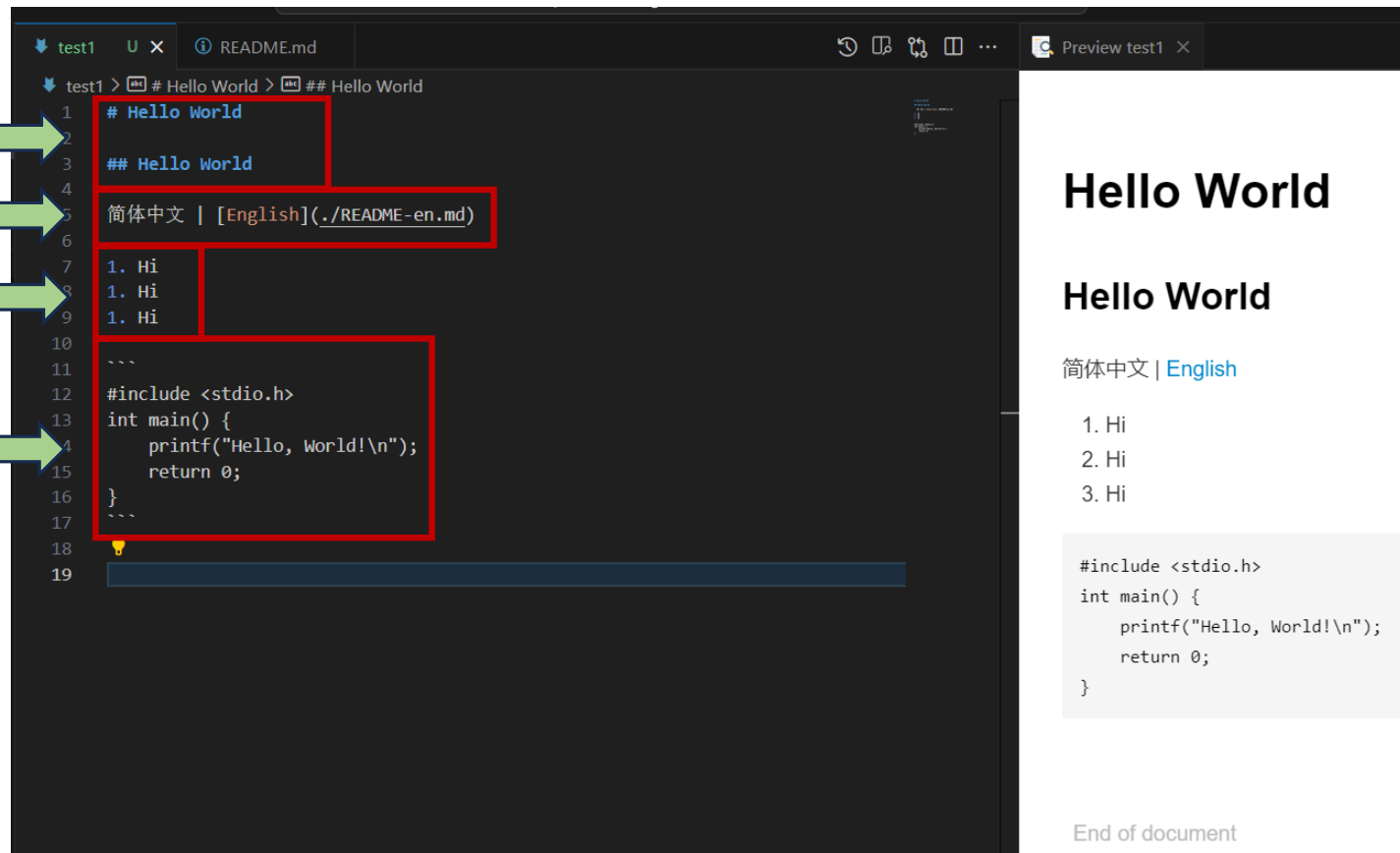
直观熟悉 Markdown 语法

标题等级

路径索引

自动序号

代码段



```
test1 > # Hello World > ## Hello World
1 # Hello World
2
3 ## Hello World
4
5 简体中文 | [English](./README-en.md)
6
7 1. Hi
8 1. Hi
9 1. Hi
10
11 ```
12 #include <stdio.h>
13 int main() {
14     printf("Hello, World!\n");
15     return 0;
16 }
17 ```
18
19
```

Preview test1

Hello World

Hello World

简体中文 | [English](#)

1. Hi
2. Hi
3. Hi

```
#include <stdio.h>
int main() {
    printf("Hello, World!\n");
    return 0;
}
```

End of document

Docs ~ Tools

Read the Docs -> 手册网页展示

1. Gitee + Markdown

2. Read the Docs + reStructuredText

reStructuredText



Sphinx: 构建生成 HTML (可以本地预览)

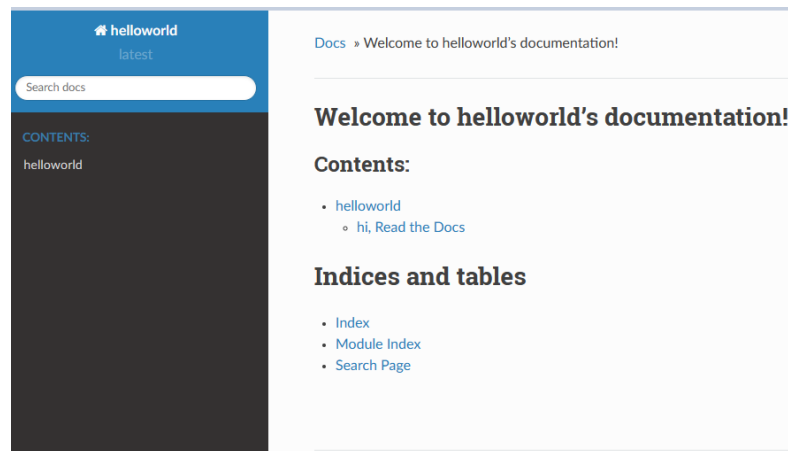


Read the Docs : 托管 HTML 网页

```
# create doc
sphinx-quickstart

# want theme ??
pip install sphinx_rtd_theme
vi /opt/docs/source/conf.py
# import sphinx_rtd_theme
# html_theme = "sphinx_rtd_theme"
# html_theme_path = [sphinx_rtd_theme.get_html_theme_path()]

make html
```



Read the Docs

项目详情

To import a project, start by entering a few details about your repository. You can set options for your documentation in a `.readthedocs.yml` file.

名称:

代码库地址:

托管文档仓库 URL

默认分支:

What branch "latest" points to. Leave empty to use the default value for your VCS.

语言:

该项目的文档所呈现的语言。注意：这影响您的项目的 URL。

下一页

Docs ~ Tools

1. Gitee + Markdown

2. Read the Docs + reStructuredText

reStructuredText

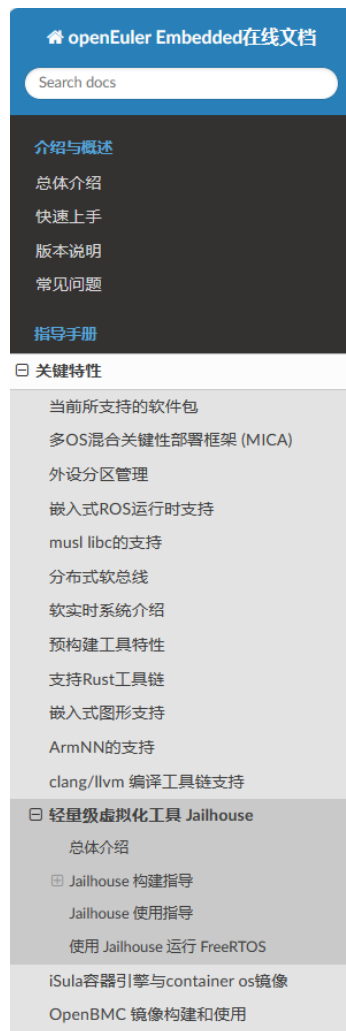


Sphinx: 构建生成 HTML (可以本地预览)



Read the Docs : 托管 HTML 网页

Read the Docs -> 手册网页展示



» 关键特性 » 轻量级虚拟化工具 Jailhouse

轻量级虚拟化工具 Jailhouse

总体介绍

Jailhouse 是一种轻量级虚拟化工具，与传统的全功能虚拟化解决方案（如 KVM 和 Xen）不同，它不提供完整的虚拟机管理和抽象功能，而是一种基于Linux的静态分区虚拟化方案。Jailhouse 不支持任何设备模拟，不同客户虚拟机之间也不共享任何 CPU，所以也没有调度器。

Jailhouse 的工作是将硬件资源进行静态分区，每个分区称为一个 cell，每个 cell 之间是相互隔离开的，并且拥有自己的硬件资源(CPU、内存、外设等)，运行在 cell 内的裸机应用程序或操作系统称为 inmate。Jailhouse 的第一个 cell 叫 Root Cell，这是一个特权Cell，内部运行的是一个 Linux 系统，依赖该 Linux 接管系统硬件资源，以及进行硬件的初始化和启动。除了 Root Cell 的其它 cell 统一称为 Non-root Cell，从 Root Cell 中获取系统资源，可独占或与 Root Cell 共享。

Jailhouse 构建指导

方法一：使用 oebuild 构建

openEuler Embedded 目前支持在 qemu-arm64 和 RPI4 上运行 Jailhouse，默认集成到了 openeuler-image-mcs 镜像，构建方法可参考 [mcs镜像构建指导](#)。

请注意，需要修改 oebuild 的编译配置文件 compile.yaml，把 MCS_FEATURES 中的 openamp 改成 jailhouse。

方法二：使用 MCS 镜像的 SDK 构建

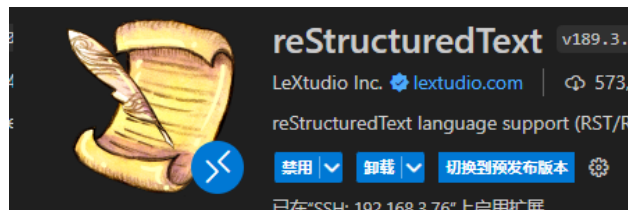
按照 [mcs镜像构建指导](#) 构建出 MCS 镜像的SDK后，可以使用 SDK 交叉编译 Jailhouse，提升开发效率。步骤如下：

Docs ~ Tools

1. Gitee + Markdown

2. Read the Docs + reStructuredText

语法稳定、丰富



Read the Docs -> 手册网页展示

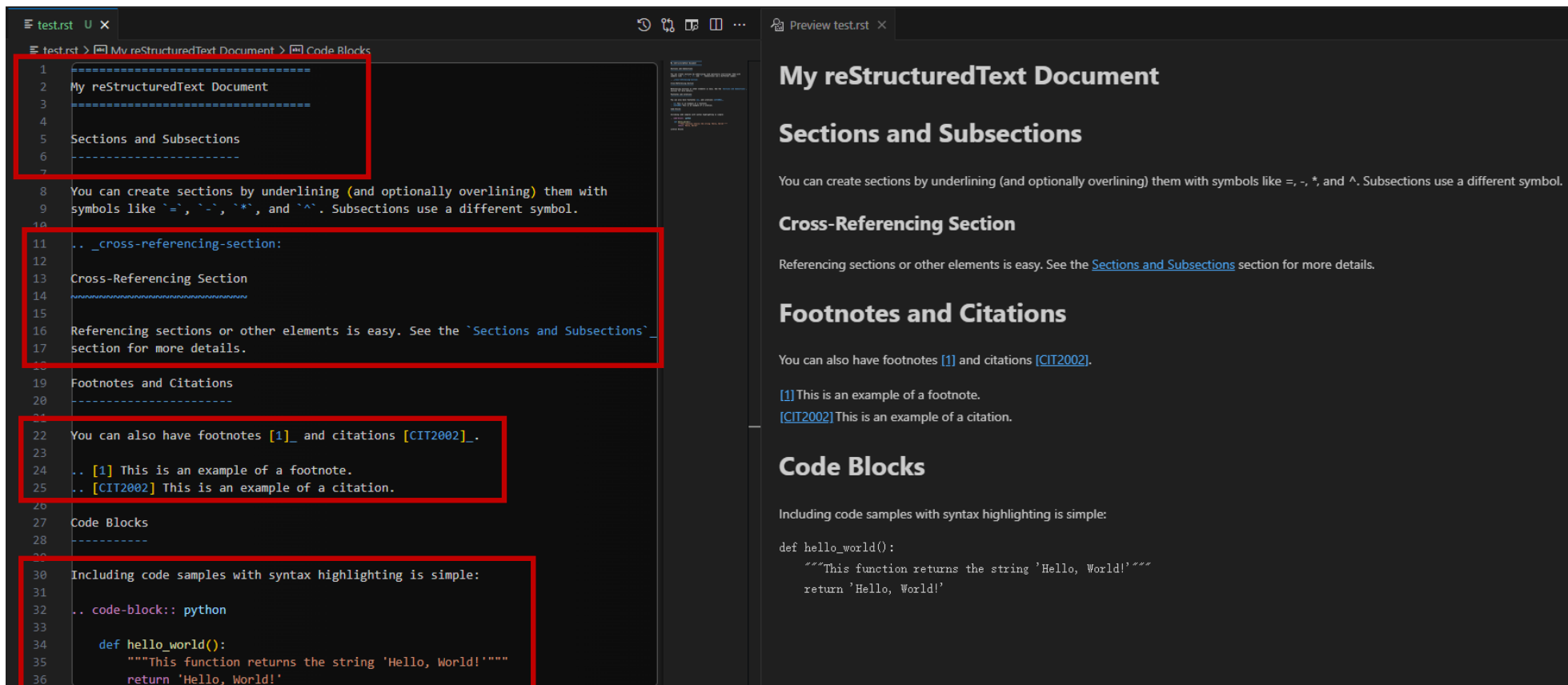
本地使用 **vscode** 插件即时预览

标题 →

交叉引用导航 →

脚注 →

代码块 →



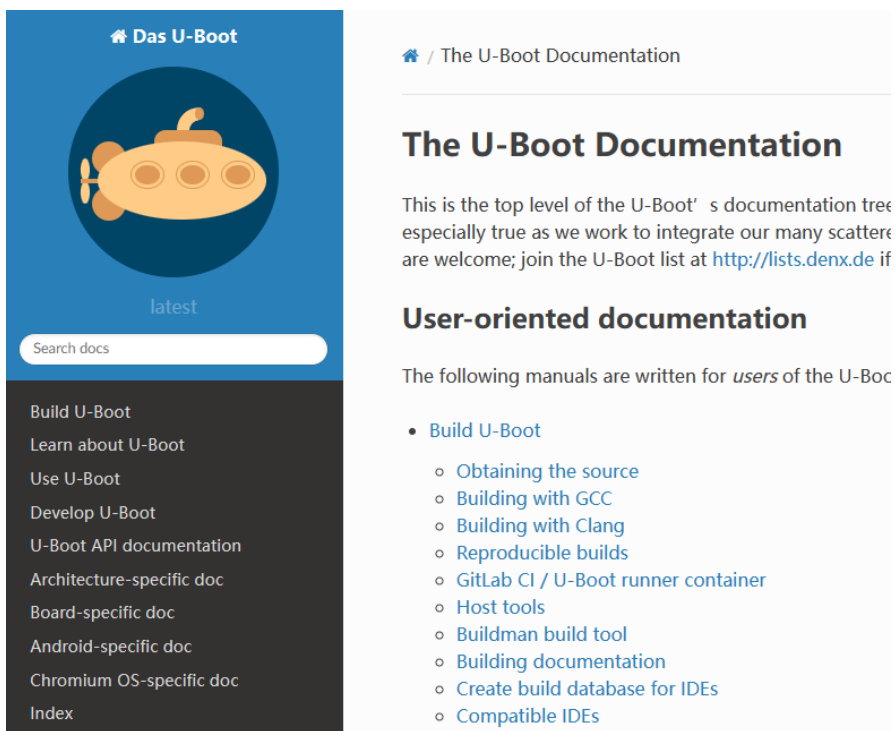
Docs ~ Tools

类似项目实践

1. Gitee + Markdown

2. Read the Docs + reStructuredText

流行的 bootloader U-Boot



🏠 / The U-Boot Documentation

The U-Boot Documentation

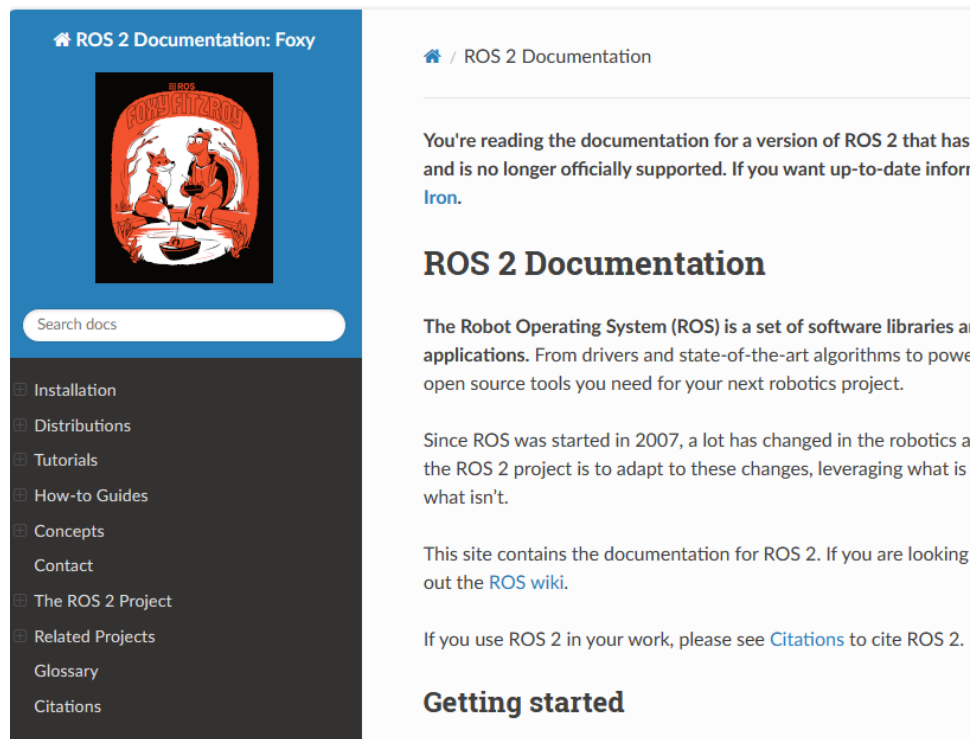
This is the top level of the U-Boot's documentation tree especially true as we work to integrate our many scattered documents; if you are welcome; join the U-Boot list at <http://lists.denx.de> if

User-oriented documentation

The following manuals are written for *users* of the U-Boot:

- Build U-Boot
 - Obtaining the source
 - Building with GCC
 - Building with Clang
 - Reproducible builds
 - GitLab CI / U-Boot runner container
 - Host tools
 - Buildman build tool
 - Building documentation
 - Create build database for IDEs
 - Compatible IDEs

机器人应用框架 ROS2



🏠 / ROS 2 Documentation

You're reading the documentation for a version of ROS 2 that has been deprecated and is no longer officially supported. If you want up-to-date information, please see the [ROS 2 wiki](#).

ROS 2 Documentation

The Robot Operating System (ROS) is a set of software libraries and tools that enable the development of robotics applications. From drivers and state-of-the-art algorithms to powerful open source tools you need for your next robotics project.

Since ROS was started in 2007, a lot has changed in the robotics community. The ROS 2 project is to adapt to these changes, leveraging what is new and discarding what isn't.

This site contains the documentation for ROS 2. If you are looking for more information, please see the [ROS 2 wiki](#).

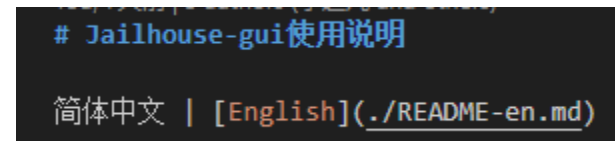
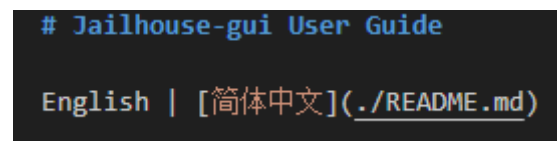
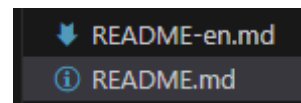
If you use ROS 2 in your work, please see [Citations](#) to cite ROS 2.

Getting started

G11N & Jailhouse

1. README -> English

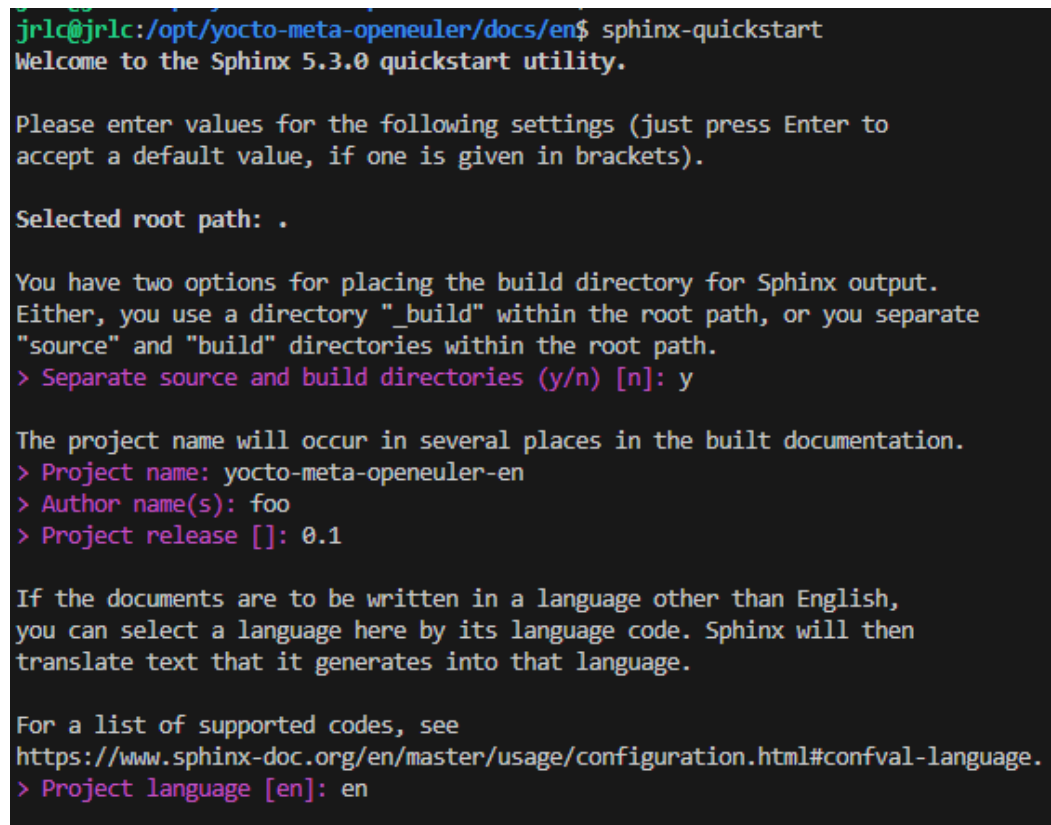
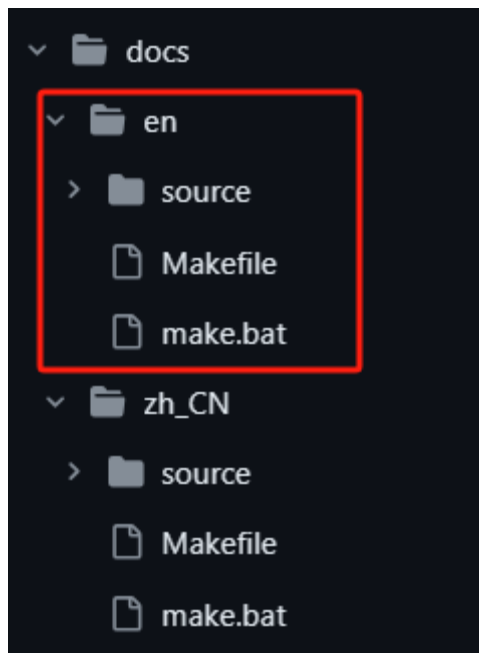
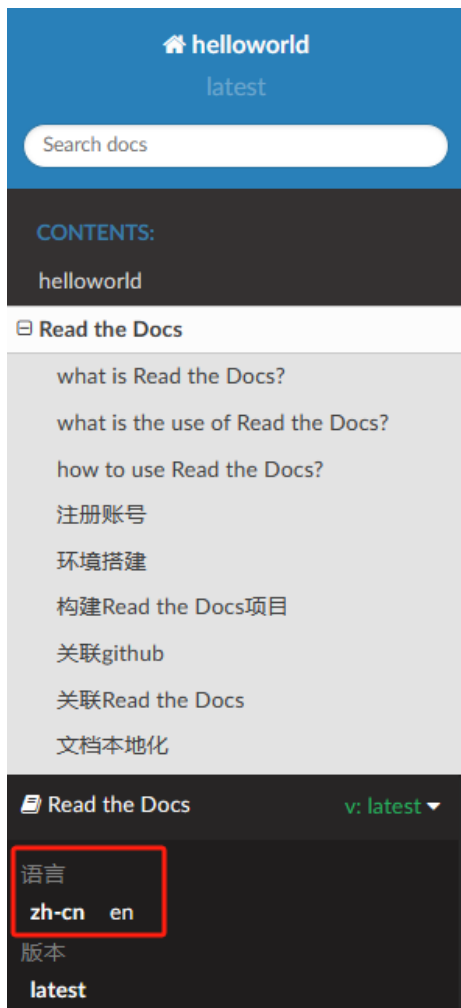
使用社区模板



确保文档路径，进行跳转

G11N & Jailhouse

2. Read the Docs -> English (maybe in the future)



THANKS