

Debian Docker开发指南

文档标识: RK-KF-YF-334

发布版本: V1.2.1

日期: 2023-04-13

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自拥有者所有。

版权所有© 2023 瑞芯微电子股份有限公司

超越合理使用范畴, 非经本公司书面许可, 任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部, 并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

本文档介绍如何搭建Docker环境，在Ubuntu系统PC上编译libmali、gststreamer、mpp、xserver、libdrm等第三仓库编译成deb包，然后安装到Debian系统中。

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

各芯片系统支持状态

仓库	版本	系统
glmark2	2021.02	Debian
gst-plugins-base	1.14.4/1.18.5	Debian
gststreamer-rockchip	1.14.4	Debian
libdrm	2.4.97/2.4.104	Debian
libdrm-cursor	1.4.0	Debian
libmali	1.9.0	Debian
mpp	1.5.0	Debian
libv4l-rkmpp	1.5.0	Debian
openbox	3.6.1	Debian
pcmanfm	1.2.5	Debian
rga	2.1.0/2.2.0	Debian
rkisp	2.2.0	Debian
rkaiq	5.0	Debian
xserver	1.20.4/1.20.11	Debian
wifibt	1.0.0	Debian
rktoolkit	1.0.0	Debian

修订记录

日期	版本	作者	修改说明
2019-08-27	V1.0.0	Caesar Wang	初始版本。
2020-02-18	V1.0.1	Caesar Wang	同步到发布版本
2021-03-15	V1.0.2	Ruby Zhang	更新产品版本信息
2021-07-19	V1.1.0	Caesar Wang	新增支持Debian10
2022-09-20	V1.2.0	Caesar Wang	新增支持Debian11
2023-04-13	V1.2.1	Caesar Wang	更新部分debian 仓库

目录

Debian Docker开发指南

1. Rockchip Docker
2. 操作系统要求
 - 2.1 安装 Docker
 - 2.2 构建应用
 - 2.3 修改Docker镜像
3. 其他
4. 实例

1. Rockchip Docker

[Docker](#) 是一个旨在让使用容器更轻松地创建、部署和运行应用程序的工具。Docker 容器允许开发人员将应用程序及其所需的所有部分打包在一起，例如库和其他依赖项，并将其作为一个包发送出去。通过这样做，开发人员可以放心，该应用程序将在任何其他 Linux 设备上运行，而不管该设备可能具有的用于与编写和测试代码的机器不同的任何自定义设置。

在某种程度上，Docker 有点像虚拟机。但与虚拟机不同的是，Docker 不是创建一个完整的虚拟操作系统，而是允许应用程序使用与它们运行的系统相同的 Linux 内核，并且只需要应用程序附带尚未在主机上运行的组件。这显著提高了性能并减小了应用程序的大小。

重要的是，Docker 是 [开源](#) 的。这意味着任何人都可以为 Docker 做出贡献并扩展它以满足自己的需求。

Rockchip Docker 可以参考第三方网站 [docker-rockchip](#)。

2. 操作系统要求

要安装 Docker，您需要以下任一版本的 64 位 Ubuntu 系统：

- Jammy 22.04 (LTS)
- Focal 20.04 (LTS)
- Bionic 18.04 (LTS)
- Xenial 16.04 (LTS)
- Trusty 14.04 (LTS)

注意：dockerfile 默认用于 ARM 64 位芯片。

以下用于 ARM 32 位芯片：

```
cp dockerfile-32 dockerfile
```

2.1 安装 Docker

- 使用此命令安装最新版本的 Docker（在 ubuntu 14.04 中将 docker 替换为 docker.io）：

```
sudo apt-get install docker qemu-user-static binfmt-support
```

- 开始运行 Docker 守护进程：

```
sudo service docker start
```

- 通过 dockerfile 构建 Docker 镜像：

```
sudo docker build -t rockchip .
```

即可得到一个名为“rockchip”的 Docker 镜像，其中包含一个 Debian 多架构交叉编译环境。

2.2 构建应用

- 进入Docker环境:

```
sudo docker run -it -v <package_dir>:/home/rk/packages rockchip /bin/bash
```

- 开始构建:

对于 ARM 32位芯片:

```
cd /home/rk/packages/<pkg_dev>
DEB_BUILD_OPTIONS=nocheck dpkg-buildpackage -rfakeroot -b -d -uc -us -aarmhf
ls ../ | grep *.deb
```

对于 ARM 64位芯片:

```
cd /home/rk/packages/<pkg_dev>
DEB_BUILD_OPTIONS=nocheck dpkg-buildpackage -rfakeroot -b -d -uc -us -aarm64
ls ../ | grep *.deb
```

2.3 修改Docker镜像

如果您想要修改 Docker 镜像, 可以通过下面的命令:

```
sudo docker run -it rockchip /bin/bash
```

从容器退出后, 使用如下的命令来保存你的更改。

```
sudo docker commit <container_id> rockchip
```

3. 其他

要获取有关 docker 的更多信息, 请查看以下链接: <https://docs.docker.com>

4. 实例

- 如何在 [libmali](#) 上生成 libmali-bifrost-g52-g2p0-x11_1.9-1_arm64.deb

```
~/work/docker/docker-rockchip$ sudo service docker start
~/work/docker/docker-rockchip$ sudo docker build -t rockchip .
~/work/docker/docker-rockchip$ sudo docker run -it -v
/home/wxt/work:/home/rk/packages rockchip /bin/bash
rk@2888134f9c12: /$ cd /home/rk/packages/docker/libmali
rk@2888134f9c12:~/packages/docker/libmali$ DEB_BUILD_OPTIONS=nocheck dpkg-
buildpackage -rfakeroot -b -d -uc -us -aarm64
```

上述步骤将获得 ~/packages/docker/ 的 debs

- 可获取一些第三方仓库编译生成 deb 包，比如：

[glmark2](#)

[libmali](#)

[mpp](#)

[rga](#)

[rkwiibt](#)

[gststreamer-rockchip](#)