## 奔2配套vmware镜像更新说明

本vmware镜像是《奔跑吧linux内核》第二版的配套实验虚拟机镜像。我们提供两个版本：

1. VMware Player 15

2. VirtualBox

大家可以根据自己的喜好来选择。虚拟机镜像都是基于Ubuntu 20.04制作而成。使用MAC Book的小伙伴可以选择VirtualBox。

用户名：rlk

密码： 123

工作目录: /home/rlk/rlk

**注意：**

1. **建议给虚拟机分配至少4GB的内存。**
2. **第一次使用时，建议做如下更新：**

**$ cd /home/rlk/rlk/runninglinuxkernel\_5.0**

**$ git pull**

### 镜像特色

1. runninglinux\_5.0：《奔跑吧Linux内核》第二版的配套实验平台。

配套的实验是在 kmodules 目录下面:

* 奔跑吧进阶篇(卷1和卷2)对应的实验：kmodules/rlk\_lab/rlk\_senior
* 奔跑吧入门篇对应的实验：kmodules/rlk\_lab/rlk\_basic

runninglinux\_5.0有两个git tree镜像。

* github：<https://github.com/figozhang/runninglinuxkernel_5.0>
* 腾讯git：

<https://benshushu.coding.net/public/runninglinuxkernel_5.0/runninglinuxkernel_5.0/git/files>

1. 内置了eclipse-cdt ，安装在/home/rlk/tools/eclipse目录下面。
2. 内置了笨叔定制版的QEMU，可以支持树莓派4。
3. 内置了openocd。
4. 内置了DS-5社区版。
5. 内置了笨叔定制版的aarch64-linux-gnu-gdb，支持tui模式显示中文的
6. 内置了vscode。
7. 内置了YCM。
8. gcc 5安装在/home/rlk/tools/目录下。

### V1.5.1版本

1. 新增riscv\_programming\_practice仓库，目录在/home/rlk/rlk
2. 更新arm\_programming\_practice，runninglinuxkernel\_5.0和runninglinuxkernel\_5.15仓库到2023年3月12日。
3. 安装libsdl2-dev libreadline-dev（编译NEMU需要）
4. gcc 7编译NEMU会报错，把x86的gcc切换到9.3

### V1.5.0版本

1. 安装了qemu-system-riscv64-7.0和qemu-system-aarch64-6.2
2. 安装了如下依赖包

sudo apt install libsdl2-2.0-0 gperf gtkwave device-tree-compiler

1. 安装了支持RVV的riscv64-unknown-elf-gcc，安装目录在/opt/riscv
2. 编译安装了v11-branch分支的iverilog
3. 安装4.1版本的verilator
4. 安装了Litmus测试环境

$ sudo apt install opam

$ opam init

$ opam update

$ opam install herdtools7

$ eval $(opam config env)

### V1.4.2版本

1. 新增runninglinuxkernel\_5.15，为了帮助大家更好体验和验证ARMv8v9新特性，例如MTE, PAC, BTI等。这个git repo支持O0编译和kdump。
2. 更新runninglinuxkernel\_5.0代码
3. 新增了6.2版本的qemu-system-aarch64，路径在/usr/local/bin/ qemu-system-aarch64-git

安装了如下依赖包：

sudo apt install librbd-dev libsource-highlight4v5

1. 新增了crash-arm64工具，可以在x86主机上解析arm64的crash dump文件，比在QEMU的arm64 Linux系统里查看crash dump文件要快速和方便的多。

### V1.4.1版本

1. 更新runninglinuxkernel\_5.0代码
2. 安装了如下软件：

sudo apt install clang exuberant-ctags python-is-python3

1. 重新配置YCM。
2. 安装python-is-python3。YCM默认使用python3
3. 重新编译YCM。

cd /home/rlk/.vim/bundle/YouCompleteMe/

python3 install.py --clangd-completer

1. 使用YCM-Generator来为runninglinuxkernel\_5.0目录生成一个.ycm\_extra\_conf.py配置文件，这个配置文件已经上传到git上，大家只要git pull一下runninglinuxkernel\_5.0即可。

如果读者想自己生成.ycm\_extra\_conf.py，可以通过如下步骤。

$ sudo apt install clang exuberant-ctags

$ git clone <https://github.com/rdnetto/YCM-Generator.git>

$ cd YCM-Generator

$ ./config\_gen.py /home/rlk/rlk/runninglinuxkernel\_5.0

1. 测试YCM。打开vim，然后打开mm/memory.c文件，在第370行，输入vma->



### V1.4.0版本

1. 更新runninglinuxkernel\_5.0代码，新增《奔跑吧linux内核 入门篇》第二版配套的实验参考代码，代码目录：/home/rlk/rlk/runninglinuxkernel\_5.0/kmodules/rlk\_lab/rlk\_basic
2. 新增如下软件包：

sudo apt install libelf-dev gcc-7 xterm curl

sudo apt build-dep linux-image-generic

1. 把x86\_64版本的GCC设置为gcc 7
2. 安装docker
3. 安装k8s

### V1.3版本

1. 关闭系统自动更新。

2. 关闭aarch64-linux-gnu-gcc 5 (update-alternatives --remove)

3. 更新runninglinuxkernel\_5.0最新代码。

4. vbox版本的显卡内存大小从16MB改成128MB。

5. 新增aarch64-linux-gnu-gdb，该版本可以在tui模式下显示中文注释。

### V1.2版本

1. 更新了QEMU for 树莓派4，支持树莓派4上的GIC中断控制器。

v1.2版本可能存在的bug：

1. vmware image v1.2版本，系统更新之后，aarch64-linux-gnu-gcc的版本从9变成了5。

2. vmware image v1.2版本，里面的runninglinuxkernel\_5.0运行不起来，请把run\_debian\_arm64.sh里面的“.gic\_version”改成“.gic-version”

### V1.0版本

1. 安装如下软件。

 sudo apt install net-tools libncurses5-dev libssl-dev build-essential openssl qemu-system-arm libncurses5-dev gcc-aarch64-linux-gnu git bison flex bc vim universal-ctags cscope cmake python3-dev gdb-multiarch openjdk-13-jre trace-cmd kernelshark bpfcc-tools cppcheck docker docker.io make libtool pkg-config autoconf automake texinfo

2. 安装： open-vm-tools open-vm-tools-desktop

3. 安装支持rpi4的qemu

4. 安装openocd

5. 安装DS-5社区版

6. 安装vim和最新的YCM

7. 安装最新版本eclipse-cdt

8. 安装vscode

9. 安装了linaro的arm-linux-gnueabi-gcc 5以及aarch64-linux-gnu-gcc 5

### 补充说明

读者若想编译rlk\_4.0中的arm64内核，需要切换到gcc-5。可以使用update-alternatives命令来切换。

关于如何使用update-alternatives可以参考《奔跑吧Linux内核 \* 入门篇>第一版第16页。

注意：编译runninglinux\_5.0请使用GCC 9版本，编译runninglinux\_4.0请使用GCC 5版本。