

Firefox 在 RISC-V 平台上的可用性报告

测试小队 & 丁丑小队

2024/11/14

摘要

报告进行了 Firefox 在 RISC-V 平台，特别是在 Milk-V Pioneer Box 和 Sipeed LicheePi 4A 上的可用性测试。测试涵盖了多个 Linux 发行版和 Firefox 版本，通过手动和自动测试评估其在 RISC-V 平台上的可用性。手动测试验证了日常使用场景，自动测试利用 web-platform-tests 对 Firefox 进行了全面的兼容性检验。结果显示，Firefox 能够在测试涉及的 RISC-V 设备上运行，但由于硬件性能和软件优化受限，存在无法使用硬件解码播放视频等问题。总体而言，Firefox 在 RISC-V 平台上已经基本可用，但仍需进一步优化和支持。

目录

1 简介	3
1.1 软件说明	3
1.2 测试概述	3
2 环境说明	3
2.1 硬件环境	3
2.2 软件环境	4
2.3 环境搭建	4
2.3.1 安装系统	4
2.3.2 手动测试环境	6
2.3.3 Firefox 自动测试环境	6
2.3.4 web-platform-tests 官方自动测试	8
2.3.5 性能测试	8
3 测试内容	9
3.1 手动测试	9
3.2 web-platform-tests 自动测试	9
3.3 性能测试	9
4 测试结果	9
4.1 手动测试	9
4.2 wpt 自动测试	10
4.3 性能测试	10
5 总结	11

1 简介

由于

1.1 软件说明

Firefox 是一个免费和开源的浏览器，全世界 10% 的人使用 Firefox 作为他们的主要浏览器。Mozilla 是 Firefox 浏览器的开发商，主要提供专注于开放网络的产品。Firefox 是谷歌 Chrome 浏览器的替代品。

1.2 测试概述

本次测试在 RISC-V 设备 Milk-V Pioneer Box 和 Sipeed LicheePi 4A 的多个 Linux 发行版上对多个版本的 Firefox 进行了手动或自动的测试，对其目前在 RISC-V 上的可用性进行了较为全面的测试并得出了相应的结论。

测试过程中采用了大量使用的 web-platform-test 测试框架在部分发行版上对浏览器进行了测试，对其进行了较为全面且完整的验证。在部分系统下还使用了 Speedometer3 和 Basemark 来测试浏览器浏览网页以及图形性能。

手动测试采用贴近真实用户日常使用场景的测试用例，模拟一般用户的操作方式对 Firefox 在测试环境上的日常浏览体验进行了评估。

2 环境说明

2.1 硬件环境

本次测试主要在 Milk-V Pioneer Box 和 Sipeed LicheePi 4A 上进行，机器硬件配置为：
Milk-V Pioneer Box:

- CPU: SG2042 64 Core C920@2.0GHz
- RAM: 4 channel 3200Hz 128GB DDR4 SODIMM (32GB * 4)
- SSD: PCIe 3.0 x 4 1TB
- GPU: AMD R5 230

Sipeed LicheePi 4A:

- CPU: TH1520, RISC-V 2.0G C910 x4
- RAM: 16 GB 64bit LPDDR4X-3733
- Storage: TF Card, 128 GB eMMC

2.2 软件环境

本次测试涵盖的系统版本和 Firefox 版本如下：

- Fedora 38（仅测试 Pioneer Box）
 - 软件仓库自带 Firefox（手动测试）
 - Nightly 版本（wpt 自动测试）
- openKylin 2.0
 - 软件仓库自带 Firefox（手动测试）
 - Nightly 版本（wpt 自动测试）
- RevyOS
 - 软件仓库自带 Firefox（手动测试）

其余未测试系统及原因如下：

- openEuler：截止测试时，仅有 CLI 版本
- openCloudOS：截止测试时，未找到可用版本
- Ubuntu：截止测试时，未找到可用版本
- Debian：RevyOS 可被视作 Debian
- Fedora：在 LicheePi 4A 上无法使用网口

2.3 环境搭建

2.3.1 安装系统

Sipeed LicheePi 4A

LicheePi 4A 各系统在 支持矩阵¹上详细记载了安装过程，可作为参考。

- openKylin 从 openKylin 官方²下载对应镜像并解压后，使用 fastboot 刷入 emmc。

按住 boot 键后，上电/Reset 进入刷写模式。

uboot 分区镜像需要根据 RAM 大小选择版本：

```
tar -xvf openKylin-Embedded-V2.0-Release-licheepi4a-riscv64.tar.xz
cd openKylin-Embedded-V2.0-Release-licheepi4a-riscv64/
sudo fastboot flash ram u-boot-with-spl-lpi4a(-16g).bin
sudo fastboot reboot
sudo fastboot flash uboot u-boot-with-spl-lpi4a(-16g).bin
sudo fastboot flash boot boot-lpi4a-20240720_171951.ext4
sudo fastboot flash root openKylin-2.0-licheepi4a-riscv64.img
```

默认账号密码为：openkylin:openkylin

¹<https://github.com/ruyisdk/support-matrix/tree/main/LicheePi4A>

²<https://www.openkylin.top/downloads/index-cn.html>

- RevyOS 从 ISCAS 镜像³ 下载并解压：

```
wget https://mirror.iscas.ac.cn/revyos/extra/images/lpi4a/20240720/boot-lpi4a-20240720_171951.ext4.zst
wget https://mirror.iscas.ac.cn/revyos/extra/images/lpi4a/20240720/u-boot-with-spl-lpi4a.bin
wget https://mirror.iscas.ac.cn/revyos/extra/images/lpi4a/20240720/root-lpi4a-20240720_171951.ext4.zst
zstd -d boot-lpi4a-20240720_171951.ext4.zst
zstd -d root-lpi4a-20240720_171951.ext4.zst
```

按住 boot 键后，上电/Reset 进入刷写模式。

刷写系统：

```
sudo fastboot devices
sudo fastboot flash ram u-boot-with-spl-lpi4a.bin
sudo fastboot reboot
sudo fastboot flash uboot u-boot-with-spl-lpi4a.bin
sudo fastboot flash boot boot-lpi4a-20240720_171951.ext4
sudo fastboot flash root root-lpi4a-20240720_171951.ext4
```

默认账号密码为：debian:debian

Milk-V Pioneer Box

- Fedora 从 MilkV 官方⁴ 给出的下载链接下载并解压后，使用 dd 刷入 SD 卡：

```
sudo dd if=path/to/fedora-disk-gnome-workstation.raw of=/dev/sdx bs=1M status=progress
```

将 SD 卡插入 Pioneer Box 后面板的 SD 卡槽中后，启动系统。

下载 mv-rootfs.sh⁵ 并以 sudo 在 Pioneer Box 上执行该脚本：

```
wget https://github.com/milkv-pioneer/scripts/blob/main/mv-rootfs.sh
sudo bash ./mv-rootfs.sh
```

按提示选择硬盘，确认和自动在 NVMe 硬盘上安装。安装完成后移除 SD 卡并重启。

- openKylin 下载 openKylin 2.0⁶ 后，使用 xz 解压，dd 到 NVMe 硬盘中*（需要一个硬盘盒）*：

```
xz -kd openKylin-Embedded-V2.0-Release-milk-v-pioneer-riscv64.img.xz
sudo dd if=path/to/openKylin-Embedded-V2.0-Release-milk-v-pioneer-riscv64.img of=/dev/your/nvme bs=4M status=progress
```

将硬盘插入系统后自动启动。

- RevyOS 下载 RevyOS 20241025⁷ 后，使用 zstd 解压，dd 到 NVMe 硬盘中*（需要一个硬盘盒）*：

³<https://mirror.iscas.ac.cn/revyos/extra/images/lpi4a/>

⁴<https://milkv.io/zh/docs/pioneer/getting-started/download>

⁵<https://github.com/milkv-pioneer/scripts/blob/main/mv-rootfs.sh>

⁶<https://mirrors.hust.edu.cn/openkylin-cdimage/2.0/openKylin-Embedded-V2.0-Release-milk-v-pioneer-riscv64.img.xz>

⁷<https://mirror.iscas.ac.cn/revyos/extra/images/sg2042/20241025/revyos-pioneer-20241025-001347.img.zst>

```
zstd -d revyos-pioneer-20241025-001347.img.zst
sudo dd if=path/to/revyos-pioneer-20241025-001347.img of=/dev/your/nvme bs=4M status=progress
```

将硬盘插入系统后自动启动。

2.3.2 手动测试环境

- Fedora 使用 **dnf** 进行安装：

```
sudo dnf update
sudo dnf install firefox
```

- openKylin 使用 **apt** 进行安装：

```
sudo apt update
sudo apt upgrade
sudo apt install firefox
```

- RevyOS 使用 **apt** 进行安装：

```
sudo apt update
sudo apt upgrade
sudo apt install firefox
```

2.3.3 Firefox 自动测试环境

Firefox 自带的 wpt 自动测试需要构建出 Nightly 环境后进行使用；可参考 官方文档⁸。

- 安装依赖：

```
sudo dnf install python3 python3-pip
# apt based:
# sudo apt-get install curl python3 python3-pip
```

- 创建虚拟环境：

⁸https://firefox-source-docs.mozilla.org/setup/linux_build.html

```
python -m venv ~/venv_wpt
source ~/venv_wpt/bin/activate
```

- 安装 hg 工具：

```
pip install mercurial
hg version
```

- 初始化 Firefox 源码：[该步骤会下载超 10G 的源代码](#)，请确保良好的网络连接

```
curl https://hg.mozilla.org/mozilla-central/raw-file/default/python/mozboot/bin/bootstrap.py -O
python bootstrap.py
```

选择 2. Firefox for Desktop 项目

- 配置依赖 Firefox 依赖于 Rust，需要采用 `rustup.sh` 安装 Rust 工具链：

```
curl --proto 'https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

其还依赖 clang 和 nodejs，根据环境安装：

```
sudo dnf install clang clang-devel nodejs
# sudo apt-get install clang clang-dev nodejs
```

该部分由于环境不同，根据 configure 提示使用 apt/pip/cargo 补全依赖即可：

```
./mach wpt configure
```

待到提示所有依赖成功安装后，可进行下一步构建。

- 构建 Nightly 使用 mach 脚本构建即可：

```
./mach build
```

该过程可能耗时 2-3h 不等。

- 运行 wpt **注意** ⚠ 该测试完全运行极其缓慢，可能耗时超过 58 个小时（单 CPU 核心下）。并行测试由于未知原因无法进行，无法加速测试。使用 mach 脚本运行 wpt 测试：

```
./mach wpt run --log-raw wpt_out.txt --log-html wpt_out.html --log-mach wpt_out_readable.txt --log-wptreport  
↪ wpt_out_report.txt --process=8
```

测试完成后，可在当前文件夹下找到测试结果。

2.3.4 web-platform-tests 官方自动测试

直接使用 web-platform-tests⁹ 中提供的测试。

- 安装环境 克隆其 git 仓库：

```
git clone https://github.com/web-platform-tests/wpt.git  
cd wpt
```

安装 Python 依赖：

```
sudo apt-get install python3 python3-pip python3-venv python-is-python3  
# sudo dnf install python3 python3-pip python3-venv python-unversioned-command
```

- 配置 hosts 向 hosts 中添加 wpt 的解析：

```
./wpt make-hosts-file | sudo tee -a /etc/hosts
```

- 运行测试：

```
./wpt --venv /home/openkylin/venv_wpt/ run --log-raw wpt_out.txt --log-html wpt_out.html --log-mach wpt_out_readable.txt  
↪ --log-wptreport wpt_out_report.txt firefox -f --process 16
```

当提示是否安装 geckodriver 时，请选择否（官方无 RISC-V 版本），否则其会下载其它架构版本影响测试。

2.3.5 性能测试

- Speedometer 3

在 Firefox 浏览器中访问：<https://browserbench.org/Speedometer3.0/>

- Basemark

在 Firefox 浏览器中访问：<https://web.basemark.com>

⁹<https://web-platform-tests.org/>

3 测试内容

3.1 手动测试

手动测试中，我们模拟用户日常使用习惯精选了 72 个测试用例进行测试，涵盖了浏览网页、下载、书签、打印、设置、历史等常见操作，并以此基础进行了实际使用体验和步骤截图。

3.2 web-platform-tests 自动测试

web-platform-tests (wpt) 自动测试套件是一个跨浏览器的 Web 平台测试项目。其根据 Web 标准实现了一系列的测试用例，来确保浏览器可被认为是与 Web 兼容的。

整个测试套件涵盖了浏览器的几乎全部方面，从 DOM 节点、CSS 绘制、Cookies、交互等底层内容，到各个 API、外设使用、浏览器核心等均有涉及，组成了 57640 个测试，总计实现了超过两百万个子测试用例，可以认为是浏览器最权威的兼容性测试之一。

wpt 官方维护着一个 x86 架构的测试看板¹⁰，可以在上面找到单 wpt 的最新结果。

由于版权和证书问题，Firefox 自带的 wpt 套件中还包含了其自己转为 wpt 编写的专用用例，以 `_mozilla` 的路径显示，因此 Firefox 自身的报告测试数量会高于 wpt 官方测试的数目，为正常情况。

3.3 性能测试

- Speedometer 3

Speedometer 3 是 Web 浏览器的基准测试，它通过对各种工作负载上的模拟用户交互进行计时来测量 Web 应用程序响应能力。

- Basemark

Basemark Web 3.0 是一个全面的 Web 浏览器性能基准测试，可测试移动或桌面系统使用基于 Web 的应用程序的能力。该基准测试包括使用网络建议和功能的各种系统和图形测试。

4 测试结果

详细测试数据可见 Github 仓库¹¹

4.1 手动测试

手动测试 Firefox 可以正常使用。除解码可能由于没有外设支持导致无法硬解外，其余工作正常。

包括 Bing、Baidu、CSDN 和知乎等网站均能正常访问，账号功能正常可以登录和同步，书签、历史记录、下载等常用功能运行良好。

由于只有软解且 CPU 性能不足，视频播放遇到卡顿，在预期之中。

¹⁰<https://wpt.fyi/results/?label=master&label=experimental&aligned>

¹¹https://github.com/wycklw/firefox_test/tree/main

在 LicheePi 4A 上的 openKylin 系统上使用 Firefox 时会出现稳定性问题，以及无法解码播放在线视频（例如 Bilibili）。

4.2 wpt 自动测试

本节所述 Firefox 均为 Nightly 版本

- Fedora 作为 Pioneer Box 官方支持的发行版，其兼容性较好。共 61486 个测试，通过 21487 个，跳过 216 个，预期错误 3409 个，已知问题 56 个；216 个失败，183 个错误，31 个预期无法通过的被通过。其余测试项无法在该平台上运行，或超时。

除 Firefox 在 CSS 上可能的固有差距外，报错大部分集中于 webgpu、webnn、webnfc 等需要与外设协同工作的组件。

- openKylin

其有一定兼容性，但和 Fedora 相比有一定差距。共 61486 个测试，通过 19663 个，跳过 479 个，预期错误 3414 个，已知问题 62 个；2036 个失败，142 个错误，32 个预期无法通过的被通过。其余测试项无法在该平台上运行，或超时。

除上述 Fedora 上出现的问题外，openKylin 上在 HTML Canvas、DOM 绘制、渲染方面也出现了一些问题。这可能与系统外设驱动、渲染驱动不同，或字体渲染错误等原因造成。

根据测试截图，差异一般较为细微，如绘图偏移、加粗、字体排版差异等，我们认为一般不影响日常使用。

4.3 性能测试

- Speedometer 3

Pioneer Box 得分为 0.747 分，LicheePi 4A 得分为 0.3249 分。作为参考，现代 x86 计算机得分一般为 10-20 分。

- Basemark

Pioneer Box 最终得分为 39.02 分。作为参考，配备了高性能独立显卡（AMD RX 6600）的现代 x86 计算机的得分为 2000 分左右。

LicheePi 4A 不支持测试所需图形渲染。

Pioneer Box 的 Basemark Web 测试兼容性数据：

- – CSS Capabilities : 59.18%
- – HTML5 Capabilities : 91.35%
- – Page Load and Responsiveness Capabilities : 91.59%
- – Resize Capabilities : 75.86%

5 总结

综上所述，我们针对 Firefox 在 RISC-V 平台上的可用性进行了多项手动和自动测试。截止到 2024 年 11 月，Firefox 在 Milk-V Pioneer Box 和 Sipeed Lichee Pi 4A 上运行较为良好。受限于 SG2042 和 TH1520 的单核性能，并不能做到非常流畅的运行。也有诸如无法使用硬件解码播放视频等问题。Firefox 在 RISC-V 平台上的支持仍然任重而道远。

附录

参考文献