

Table of Contents

Authors.....	2
Disclaimer.....	2
Introduction.....	2
Selection of the Raspberry Pi OS distribution package.....	3
How to prepare and install the original Raspberry Pi OS image to run in read-only mode.....	4
What is under the hood.....	5
Conclusions.....	6
License.....	7
References.....	7

How to configure Raspberry Pi OS to run in read-only mode.

Authors.

Copyright (C) 2021, Vladimir Zlatov, Evgueni Tzvetanov, USA
Email us at support@altomaxtech.com.

Disclaimer

The authors of this article give no warranty whatsoever and accept no responsibility if you damage your equipment. Under no circumstances should the authors be held responsible or liable for any claims, damages, loses, expenses, costs or liabilities whatsoever (including, without limitation any direct or indirect damages for loss of profit, business interruption or loss of information) resulting or arising directly or indirectly from your use of or inability to use this article or any articles referenced here.

The authors cannot guarantee the validity of all the information contained in this article. While we use reasonable efforts to include accurate and up to date information, we make no warranties as to the accuracy of the content and assume no liability or responsibility for an error or omission in the content.

Introduction.

The goal of this article is to introduce a method of running Raspberry Pi OS in read-only mode that among other benefits helps

- to achieve better performance,
- to extend the life of Raspberry Pi's SD card,
- and to prevent card corruption due to improper system shutdown.

Indeed several methods have been already introduced/discussed (for example, [5]-[8],[13]-[14], [25], [36]) but those methods may either:

- not provide a complete solution,
- use obsolete tools and practices,
- require the removal of existing OS components,
- or require installation of new software packages.

The proposed method allows running Raspberry Pi's OS in read-only mode by using tools already available in the Raspberry Pi OS. The method is not specific just for the Raspberry Pi OS, it is also compatible with many other modern Linux distributions using 'systemd' and 'tmpfs' components. The method is particularly suitable for existing systems as all changes are reversible. The system can be switched easily between read-only and read-write mode to allow updates and other write operations.

The described method was tested with a read-only system consisting of a Raspberry Pi 3 Model B board [2] (and above), a HiFiBerry sound board [18], and Raspberry Pi OS with Desktop [3] to run Music Player Daemon (MPD) [19] as a server-side application for playing music.

The default behavior of Raspberry Pi OS is to automatically expand the root partition to the whole SD card during installation. However, this behavior does not suit some applications [29].

In the article we show how to modify this default system behavior and how to limit the root partition to a reasonable size. The rest of the card can be partitioned and formatted independently for any other needs.

For example, in our case we formatted this partition as r/w ext4 file system and configured the partition to be mounted automatically on startup. We will use it to store “onboard” music.

This article only discusses how to configure a Raspberry Pi board to run basic Raspberry Pi OS in read-only mode. Other aspects of usage and automation scripts will be provided at a later date.

Note: For information on HiFiBerry sound boards configuration consult the manufacturer’s site [18]. Answers to questions related to MPD configuration can be found in many sources on the Web, for example, [18], [19]-[21], [32]-[33]. Questions about automatic mount of USB sticks/disks are answered by many other articles available on the Web.

At the current implementation stage the method discussed in this article has a limitation. While we are using a system, the default run-time mode of which is GUI, after converting the file system through our method Raspberry Pi will boot to command-line interface (CLI). The authors have plans and hope to overcome this limitation in the nearest future and enhance the method to be able to run the Raspberry Pi OS with Desktop in full power booting to GUI mode as well.

A complete set of Linux scripts is provided to automate most of tedious configuration operations. The scripts can be downloaded from <https://altomaxtech.com/rpi-ro/>.

Selection of the Raspberry Pi OS distribution package.

The method described here is applicable to any of the available packages at the moment of writing the article [3]:

- Raspberry Pi OS with desktop and recommended software (full)
- Raspberry Pi OS with desktop
- Raspberry Pi OS Lite

Though, the “Raspberry Pi OS with desktop” has been chosen as the optimal for our goals because it supports GUI and doesn’t include unnecessary software.

NOTE. In testing we successfully used several versions of “Raspberry Pi OS with desktop” and “Raspberry Pi OS Lite” images released in 2021 so far.

The scripts automatically recognize Raspberry Pi OS files using the current naming scheme “YYYY-MM-DD-raspbios-*-armhf-*.zip”. Partition types and offsets are also automatically calculated using the command partx. It is assumed that a Raspberry Pi OS image contains two main partitions: the first of type “0xc” (vfat) as “boot” and the second of type “0x83” (ext4) as “root”.

If these assumptions are incorrect, the scripts need to be modified accordingly.

How to prepare and install the original Raspberry Pi OS image to run in read-only mode.

This can be done in several ways:

- - Case #1 - The most automatic way applicable for new installs. Automatically unpacks and edits the original Raspberry Pi OS image:
 - The assumption is that on your workstation machine you are running one of the latest Ubuntu or Debian distributions and your shell is bash. If this is not the case, you will need to modify the scripts to match your distribution.
 - Download [automatic-preparation.zip](#)
 - Unpack the package and copy script files to a local folder. You can place them in the same directory where the Raspberry Pi OS zip file is located.
 - Read the instructions provided with the package and verify that your Linux workstation configuration meets the requirements.
 - Run the main script to unpack and prepare your Raspberry Pi system image.
 - Verify that the image was unpacked and modified successfully. Make sure it is located in the /tmp directory.
 - Burn the image onto a SD card (recommended SD card size is 8 GB+).
 - Install the modified Raspberry Pi OS image following the original distribution instructions [3].
 - Note that in this case the installed system will occupy about 50% of the root partition. The rest of the card will be available for partitioning and formatting.
 - Also note that all necessary configuration scripts and files will be automatically copied to appropriate locations of the modified system image and will be ready for immediate use.
 - Run your Raspberry Pi with the modified system on the SD card. System should boot to the default target which is “graphical.target” for the OS with Desktop.
 - To configure read-only mode run each “conf-0-?-...” script following the provided instructions.
 - The installation should take about 40-60 minutes in total, including SD card burning (about 7 minutes) and full system update (about 20 minutes).
 -
 - Case #2 - by copying manually provided configuration scripts onto the SD card with installed original Raspberry Pi OS image:
 - Burn and install the original Raspberry Pi OS image following the original distribution instructions [3].
 - Note that in this case the system will be installed using the whole SD card by default.
 - Mount the installed Raspberry Pi OS image (on the SD card) on another (Linux) machine (specific partition offsets for mount command can be found by the method described in [26]).
 - Download [manual-preparation.zip](#)
 - Unpack the scripts package and copy files to the mounted Raspberry Pi OS system following the provided instructions.

- Unmount the Raspberry Pi OS image to save changes.
- Run your Raspberry Pi with the modified system on the SD card booting to default target (graphical.target for the OS with Desktop, or multi-user.target if the system is already in the CLI mode).
- To configure read-only mode run each “conf-0-?-...” script following the provided instructions.
-
- Case #3 - by copying manually the provided configuration scripts from USB stick to your Raspberry Pi with existing system on it:
 - Download [manual-preparation.zip](#)
 - Unpack the script package and copy files to a USB stick.
 - Mount the USB stick on your Raspberry Pi.
 - Copy files from the USB stick to the system following the provided instructions.
 - To configure read-only mode run each “conf-0-?-...” script following the provided instructions.

What is under the hood.

The main goal of making Raspberry Pi OS to run in read-only mode is achieved by mounting several essential for system services directories as tmpfs [5,6,11,13] and by disabling and masking the swap system service [5,6].

The following directories are mounted as tmpfs:

- /tmp
- /var/tmp
- /var/log
- /var/lib/dhcp
- /var/lib/dhcpd5
- /var/lib/systemd
- /media
- /mnt

The last two directories are included to the list to support auto and manual mounts of external drives while the system is in the read-only mode. The others are needed for system services.

Necessity of a directory to have write access and other additional rights was determined by analyzing system logs (for example, by the command “journalctl -b -p 0..[3|4]”) signaling about new errors (and warnings) after switching the system to the read-only mode. The same method can be used in case of installing some new software by comparing list of errors “before” and “after”.

Note that you may experience some errors in the system installed from the original distribution package. While those errors might be quite important for the system stability and security, they should not be really taken into account during read-only mode tuning up.

To setup the directories from the list above as tmpfs the systemd [12] units were used. The original Raspberry Pi OS system (and other Linux distributions) includes the “tmp.mount” file located in the “/usr/share/systemd” directory. In the Raspberry Pi OS it is located in the directory “/usr/share/systemd”.

For each listed above directory we copy the template to “/etc/systemd/system” , rename it accordingly, and modify a few internal parameters relevant to the unit name and required mode. When a unit is in its place we can issue a command like “sudo systemctl enable <unit-name>” to enable the unit to execute on startup.

System SWAP in Raspberry Pi OS is implemented as a systemd service with the name “dphys-swapfile”. To completely disable the SWAP mechanism the service should be stopped, and then masked to make sure the file “/var/swap” is also removed.

All the modifications made to the system to run in read-only mode are reversible and if necessary can be restored temporarily or permanently to the original state.

Limiting the size of the root partition of the system [29] not to take the whole SD card space can be achieved by editing the file “/usr/lib/raspi-config/init_resize.sh” in the original system image before the first run.

To automate this operation we provide the optional script “custom-additions-01-limit-system-partition-resize.sh” that automatically runs during preparation if located in the same directory as the main script “prepare-system-image-for-read-only-mode.sh”. The resize limit is set to double the initial original root partition size. It can be changed to any desired value by editing the function “modify_init_resize_sh” in the “custom-additions-01-limit-system-partition-resize.sh” script.

Conclusions.

This article shown how to safely configure Raspberry Pi OS (with Desktop) [3] in read-only mode in order to prevent the system SD card from wearing too soon or being corrupted by unexpected power loss.

The method uses only standard commands which makes it applicable to many modern Linux distributions. It can also be applied (in its entirety, or partially) to Linux systems using SSD drives in order to prolong their lifetime.

A set of Linux scripts is developed to automate the configuration process. The scripts are available for download at <https://altomaxtech.com/rpi-ro/>.

Hope you enjoyed the article and found it helpful.

Come often and check out our other publications @ <https://altomaxtech.com/>” including enhancements for the discussed method.

You also may be interested in our other great free products, such as the network packet capture software Altomax TrafficPeeker ® for Windows that can be found there along with the complementary scripts for this article.

We would highly appreciate your opinion and feedback @ <https://altomaxtech.com/>".
Email us at support@altomaxtech.com.

License.

This article and the complementary library are licensed under LGPL v2.1+ [37] and can be found at <https://altomaxtech.com/rpi-ro/>

Copyright (C) 2021, Vladimir Zlatov, Evgueni Tzvetanov, USA
Email us at support@altomaxtech.com.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

References.

- [1] Raspberry Pi Products. <https://www.raspberrypi.org/products/>
- [2] Raspberry Pi 3 Model B. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [3] Raspberry Pi Operating System Images. <https://www.raspberrypi.org/software/operating-systems/>
- [4] Multiple partitions on Raspberry Pi memory card. <https://raspberrypi.stackexchange.com/questions/28950/multiple-partitions-on-raspberry-pi-memory-card>
- [5] How can I extend the life of my SD card? <https://raspberrypi.stackexchange.com/questions/169/how-can-i-extend-the-life-of-my-sd-card>
- [6] Extending the life of your Raspberry PI SD Card. <https://domoticproject.com/extending-life-raspberry-pi-sd-card/>
- [7] How can microSD flash memory wear be minimized? <https://www.thethingsnetwork.org/forum/t/how-can-microsd-flash-memory-wear-be-minimized/15896>
- [8] Extend The Lifespan of Your Raspberry Pi's SD Card with log2ram. <https://levelup.gitconnected.com/extend-the-lifespan-of-your-raspberry-pis-sd-card-with-log2ram-5929bf3316f2>
- [9] Documentation > Configuration > raspi-config. <https://www.raspberrypi.org/documentation/configuration/raspi-config.md>

- [10] Documentation > Configuration > Security.
<https://www.raspberrypi.org/documentation/configuration/security.md>
- [11] tmpfs(5) — Linux manual page. <https://man7.org/linux/man-pages/man5/tmpfs.5.html>
- [12] systemd.mount — Mount unit configuration.
<https://www.freedesktop.org/software/systemd/man/systemd.mount.html>
- [13] Make your Raspberry Pi file system read-only (Raspbian Buster), by Andreas Schallwig, Sep 24, 2019 <https://medium.com/swlh/make-your-raspberry-pi-file-system-read-only-raspbian-buster-c558694de79>
- [14] How to make your Raspberry Pi file system read-only (Raspbian Stretch), by Andreas Schallwig, Jun 10, 2019 <https://medium.com/@andreas.schallwig/how-to-make-your-raspberry-pi-file-system-read-only-raspbian-stretch-80c0f7be7353>
- [15] specify mount option in /etc/fstab (mode=1777)
<https://unix.stackexchange.com/questions/171726/specify-mount-option-in-etc-fstab-mode-1777>
- [16] systemd-timesyncd.service: Failed to run 'start' task: Read-only file system.
<https://bugs.launchpad.net/canonical-devices-system-image/+bug/1625446>
- [17] Systemd mount fails. Where= setting doesn't match unit name.
<https://unix.stackexchange.com/questions/283442/systemd-mount-fails-where-setting-doesnt-match-unit-name>
- [18] HiFiBerry boards. Configuring Linux 4.x or higher.
<https://www.hifiberry.com/docs/software/configuring-linux-3-18-x/>
- [19] Music Player Daemon. <https://www.musicpd.org/>
- [20] Music Player Daemon 0.22.7 documentation » User's Manual.
<https://www.musicpd.org/doc/html/user.html>
- [21] ArchLinux Wiki. Music Player Daemon.
https://wiki.archlinux.org/index.php/Music_Player_Daemon
- [22] Ubuntu Wiki. LightDM. <https://wiki.ubuntu.com/LightDM>
- [23] How To Change Username On Ubuntu, Debian, Linux Mint Or Fedora.
<https://www.linuxuprising.com/2019/04/how-to-change-username-on-ubuntu-debian.html>
- [24] How to modify a Linux username. <https://www.techrepublic.com/article/how-to-modify-a-linux-username/>
- [25] Prevent SD card from being corrupted (improper shutdown)
<https://www.raspberrypi.org/forums/viewtopic.php?t=117287>
- [26] “How do i open/edit a raspberry image?” <https://www.raspberrypi.org/forums/viewtopic.php?t=28860#p254639>
- [27] Modify a disk image to create a Raspberry Pi-based homelab
<https://opensource.com/article/20/5/disk-image-raspberry-pi>
- [28] How to Modify a Raw Disk Image of Your Custom Linux Distro
<https://www.linux.com/news/how-modify-raw-disk-image-your-custom-linux-distro/>
- [29] Temporarily disable expand filesystem during first boot.
<https://raspberrypi.stackexchange.com/questions/56621/temporarily-disable-expand-filesystem-during-first-boot>
- [30] Documentation > Configuration > config-txt
<https://www.raspberrypi.org/documentation/configuration/config-txt/README.md>
- [31] Uncomplicated Firewall. <https://help.ubuntu.com/community/UFW>
- [32] My ultimate guide to the Raspberry pi audio server I wanted — MPD. Part 4. By Mathieu Requillart (Apr 21, 2020) <https://mathieu-requillart.medium.com/my-ultimate-guide-to-the-raspberry-pi-audio-server-i-wanted-mpd-7ec04aabf91f>
- [33] My ultimate guide to the Raspberry pi audio server I wanted — DLNA. Part 4. By Mathieu Requillart (Apr 21, 2020) <https://mathieu-requillart.medium.com/my-ultimate-guide-to-the-raspberry-pi-audio-server-i-wanted-dlna-5b9dcd3c912e>

[34] My ultimate guide to the Raspberry pi audio server I wanted — USB sticks. Part 5. By Mathieu Requillart (Apr 21, 2020) <https://mathieu-requillart.medium.com/my-ultimate-guide-to-the-raspberry-pi-audio-server-i-wanted-usb-sticks-c5d7d76e236a>

[35] My ultimate guide to the Raspberry pi audio server I wanted — Bluetooth. Part 1. By Mathieu Requillart (Apr 21, 2020) <https://mathieu-requillart.medium.com/my-ultimate-guide-to-the-raspberry-pi-audio-server-i-wanted-bluetooth-64c347ee0d22>

[36] Raspberry Pi: Extending the life of the SD card. <https://www.zdnet.com/article/raspberry-pi-extending-the-life-of-the-sd-card/>

[37] GNU Lesser General Public License, version 2.1
<https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>

[...] ... and many-many other helpful links not listed here due to limited space... Many thanks to everyone!