

# 2020 REXxLA International Rexx Language Symposium Proceedings

René Vincent Jansen (ed.)

THE REXX LANGUAGE ASSOCIATION  
REXXLA Symposium Proceedings Series  
ISSN 1534-8954

## Publication Data

©Copyright The Rexx Language Association, 2023

All original material in this publication is published under the Creative Commons - Share Alike 3.0 License as stated at <https://creativecommons.org/licenses/by-nc-sa/3.0/us/legalcode>.

A publication of **RexxLA Press**

The responsible publisher of this edition is identified as *IBizz IT Services and Consultancy*, Amsteldijk 14, 1074 HR Amsterdam, a registered company governed by the laws of the Kingdom of The Netherlands.

The RexxLA Symposium Series is registered under ISSN 1534-8954

The 2020 edition is registered under ISBN 978-94-648-5246-2



2023-05-31 First printing

---

## Introduction

### History of the International REXX Language Symposium

In 1990, Cathie Dager of SLAC<sup>1</sup> convened the organizing committee for the first independent REXX<sup>2</sup> Symposium for Developers and Users. SLAC continued to organize this annual event until the middle of the 1990's when the REXXLA took over that responsibility. Symposia have been held annually since 1990.

### About REXXLA

During the 1993 Symposium in La Jolla, California, plans for a REXX User Group materialized. The REXX Language Association (REXXLA), as it was called, is an independent, non-profit organization dedicated to promoting the use and understanding of the REXX programming language. REXXLA manages several open source implementations of REXX.

### The selection procedure

Presentation proposals are solicited yearly using a CFP<sup>3</sup> procedure, after which the REXXLA symposium committee reviews them and votes which presentations are selected for the symposium. The presentations are peer reviewed before being presented. Presenters are not compensated for their presentations.

### Location

The 2020 symposium was held Online from 29 Sep 2020 to 2 Oct 2020.

---

<sup>1</sup>Stanford Linear Accelerator Center, since 2008 SLAC National Accelerator Laboratory

<sup>2</sup>Cowlshaw, M. F., **The REXX Language** (second edition), ISBN 0-13-780651-5, Prentice-Hall, 1990.

<sup>3</sup>Call For Papers.

---

# Contents

1	Picture Processing in Rexx – Walter Pachtl	1
2	Implementing NetRexx and BSF4ooRexx Solutions for a Raspberry Pi 4 Linux Desktop – Tony Dycks	7
3	ooRexx and Web Servers – Rony G. Flatscher	37
4	Fixing BRexx on VM/370 – Adrian Sutherland	63
5	The VM/370 interface to Rexx – Bob Bolch	73
6	TSO Rexx API for PCRE – Ze’ev Atlas	95
7	Creating Freud 2.0 with REXX – Michael Beer	121
8	NetRexx and BSF4ooRexx Code Examples for Redis (Using the Jedis Java Client) – Tony Dycks	141
9	Producing the ooRexx documents – Gil Barmwater	167
10	From Rexx to ooRexx – Rony G. Flatscher	171
11	CMS/NetRexx Pipelines: what’s new (and what’s old) – Jeff Hennick	187
12	ArcaOs (OS/2) ooRexx update – Roderick Klein	202
13	Running Rexx from a USB drive – Rony G. Flatscher	207
14	DBM Key/Value Data Store APIs for NetRexx and BSF4ooRexx – Tony Dycks	215

# Picture Processing in Rexx – Walter Pachtl

## Date and Time

30 Sep 2020, 16:00:00 CET

## Presenter

Walter Pachtl

## Presenter Details

Walter worked for 25 years in the IBM Laboratory Vienna. He worked on several program products such as DITTO, SDF, and a PL/I subset compiler for DPPX. The highlight of his career were the test activities for the REXX Compiler. A fully automated test environment allowed for continuous regression testing and ensured a high quality product. When "freed" from IBM, Walter worked at an Austrian bank where he started to convert many of the (VM/CMS) REXX tools that we developed and enjoyed in the Lab to TSO/ISPF. Real work was, however, based on PL/I and DB2. You can find many entries he made on [rosettacode.org](http://rosettacode.org) under (Classic) REXX and some under ooREXX, netREXX, and PL/I.

## Picture Processing Using REXX

The 2020 International Rexx SymposiumOnline ("Covid-19")

September 29th – October 1st 2020

Walter Pachi

Many years ago I learned how to generate BMP files using Rexx.

An algorithm computed the walls of a rectangular maze of an arbitrary size with a single entry and exit. The walls had then to be turned into Xs to be printed on our (IBM's) 1403 printer. Modern technology made me change the Xs to black blocks and the optionally printed path to a red line.

An example of such a picture you can see on

[https://austria-forum.org/af/Infos\\_zum\\_AF/Editorial\\_Board/Pachi%2C\\_Walter%2C\\_Dipl.-Ing./Pachi%2C\\_Walter\\_english](https://austria-forum.org/af/Infos_zum_AF/Editorial_Board/Pachi%2C_Walter%2C_Dipl.-Ing./Pachi%2C_Walter_english)

Recently I found a challenge to manipulate a nice picture of our granddaughter which shows her with two kinds of fabric that don't quite fit together



First I transformed the given jpeg file into bmp format.

The structure of a bmp file is described in Wikipedia

[https://de.wikipedia.org/wiki/Windows\\_Bitmap](https://de.wikipedia.org/wiki/Windows_Bitmap)

It comprises a header of 54 bytes followed by the picture contents. The header contains the width  $w$  and height  $h$  of the picture encoded as little endian numbers.

A little function converts these to numbers as used in REXX.

```
dnel: Procedure
/*****
* compute the number from its representation (little endian)
*****/
Parse Arg s
sr=reverse(s)
res=c2d(sr)
say 'dnel:' c2x(s) '=>' c2x(sr) '=>' c2d(sr)
Return res
```

The picture content follows the header in  $h$  lines, each of which contains  $w*3$  bytes, i.e.,  $w$  pixels. The order of these lines is bottom up; the first line contains the lowest line of the picture.

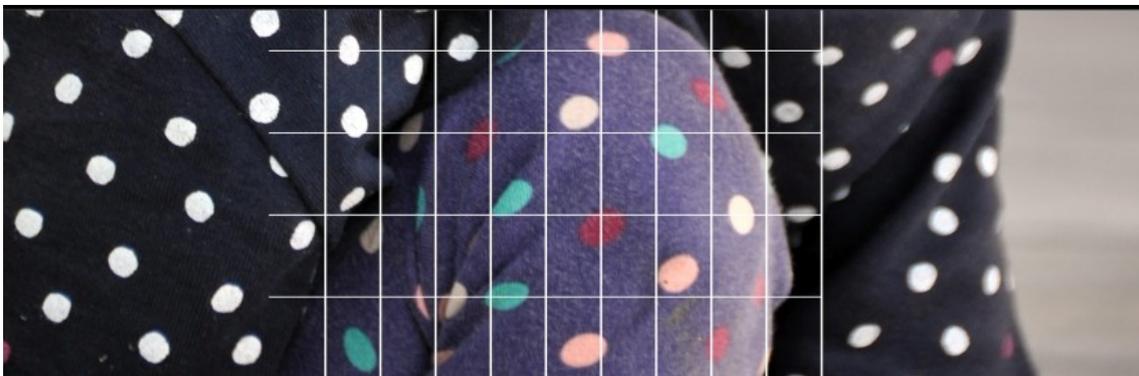
My program stores this data in an array  $l.i$  with  $i$  running from 1 to  $h$ .

The next step is to identify the area that needs to be replaced.

I do this by working on two polygons, describing the left and right border of the area, respectively.

The vertices along the line are specified as  $i$  (the line we are in) and  $x$ , in terms of bytes, from the start of the line. This takes a little trial and error.

The program can display a grid highlighting specific lines and columns.



In order to speed up the process, only the first 900 lines are used in these steps.

Another feature is to draw white edges according to the specified polygons. The  $x$  values for every line are computed using the segment specifications of the neighboring vertices.

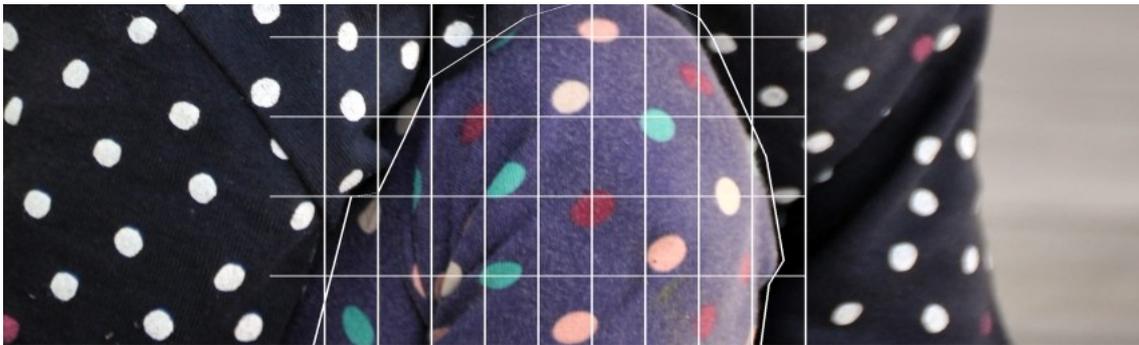
The start point of a replacement must now be adjusted to a pixel boundary, i.e., it must be of the form  $1+3*x$ .

Similarly, the length of a replacement must be a multiple of 3 (bytes).

This picture shows the boundaries.



Combining these two (grid and border) gives this:



Now we can put white pixels into all points of the replacement area thereby showing which "bad" parts of the picture will be replaced.



### Replaced by what?

I identified an area of the picture with data suitable for replacement. Replacement strings are copied from these lines into an array r.i.

Finally, we put these r.i strings into the corresponding l.i lines and build the output file by appending the picture data to the original header (remember? t)



## Some time is left?

Some years ago I created my home page [www.wpachl.at](http://www.wpachl.at) which has links that I can use from anywhere:

<a href="#">Quiz: Capitals of USA States</a> <input type="text" value="Google"/> <input type="text" value="jSparrow de Bono words"/>		<b>Email an Walter SENDEN Bitte testen&gt;</b>
<b>My internet links</b>		
<a href="#">Rexx Programs</a> (14.06.2015)		
<a href="#">Mike Cowlishaw's Rexx Programs</a> (courtesy mfc) <a href="#">Ruurd Idenburg's Rexx Programs</a> (courtesy Ruurd) <b>rosettacode.org</b> : Hundreds of Algorithms in many Languages (including REXX, ooRexx, and netrex) <b>PDSCOPY</b> - courtesy Phil Sevetson <b>NetRexx Examples</b> :...under construction (11.06.2015).		
<input type="text" value="webmail"/>		

On [http://www.wpachl.at/Rexx\\_Programs.html](http://www.wpachl.at/Rexx_Programs.html) you can find some of my programs that I considered worth “publishing”.

<b>Name</b>	<b>Type</b>	<b>Description</b>
<a href="#">first.rex</a>	pgm	First Entry.
<a href="#">72.rex</a>	pgm	Convert a long text file to line length 72.
<a href="#">compaxx.rex</a>	pgm	Compare two (text) files line by line.
<a href="#">csv2txt.rex</a>	pgm	Convert a csv file to a text file (columns aligned).
<a href="#">decrypt.rex</a>	pgm	Decode an encoded file using the key phrase used.
<a href="#">encrypt.rex</a>	pgm	Encode a file using a key phrase.
<a href="#">exists.rex</a>	fun	Check if a specified file exists.
<a href="#">fn.rex</a>	fun	Return file name.
<a href="#">fore.rex</a>	fun	Determine if a program runs in the foreground.
<a href="#">host.rex</a>	fun	Determine if a program runs on the host.
<a href="#">ipod2lst.rex</a>	pgm	List the contents of an IPOD.
<a href="#">mp3md.rex</a>	pgm	Extract meta data from an mp3 file.
<a href="#">MusicList.rex</a>	pgm	List mp3 files with metadata contained in a folder.
<a href="#">pryn.rex</a>	fun	Prompt for Y or N.
<a href="#">safecrea.rex</a>	fun	Safe file creation tool.
<a href="#">safecrea2.rex</a>	fun	Safe file creation tool with append option.
<a href="#">wordsort.rex</a>	fun	Sort a list of words.

# Implementing NetRexx and BSF4ooRexx Solutions for a Raspberry Pi 4 Linux Desktop – Tony Dycks

## Date and Time

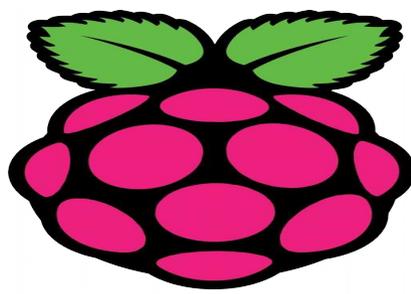
30 Sep 2020, 17:00:00 CET

## Presenter

Tony Dycks

## Presenter Details

Web Development and Database Consultant for Clients in the Healthcare and Entertainment Industry. Initial Use of REXX started in the mid-1990s for IBM OS/2 DB/2 Applications. I have used Object REXX, Regina REXX, NetREXX and BSF4ooREXX on a number of Database Management Systems including IBM DB2 and Apache Derby, Microsoft Access and SQL Server, MySQL, HSQLDB, H2 DBMS and SQLite Versions 2 and 3. Other programming language experiences include Python, Perl 5, PHP, Java, C#, Visual Basic, VBScript, Java Script and Lua.



# Raspberry Pi

## Rexx LA Virtual Symposium 2020 A Survey of Linux Distros for the RPi4

Documentation of Installation Experiences, System Use Findings and Recommendations for Implementing a Rexx Desktop Development Environment on the Raspberry Pi 4 Model B Single Board PC.

by Tony Dycks

**Last Revised:** September 30, 2020

## Overview of Presentation

- **Objectives for R Pi4 Desktop Environment**
- **Hardware Utilized with Findings and Recommendations**
- **Linux Distros Selected with Installation Details, Findings, References and Recommendations**
- **Linux Packages Installed as Rexx Pre-Requisites**
- **Open Object Rexx Installation Details for Distros**
- **Net Rexx Installation Details & Findings for Distros**
- **BSF4ooRexx Installation Details for Distros**
- **Checking the R Pi's Temperature**
- **Backing Up the R Pi's SD Card Image on Another Linux PC**



# Linux Distros Selected - I

## Linux Distro Selection Criteria

Easily Implemented from a Linux or Windows Base Workstation

No Additional Hardware Required Other Than The Raspberry Pi 4 Board, Keyboard, Mouse, Display, Power Supply, microSD Card, Cat 5 Internet Connection and a Base Linux Workstation for setup and backup storage

A Distro that has a good base of Installation Software Packages (Debian, Ubuntu, Arch Linux, Mint, Manjaro, CentOS, Fedora and openSUSE to name the possible candidates)

Stability in the Distro

Linux Based with a reliable GUI Desktop and stable File Manager

3

## Linux Distros Considered and Implemented

### Desktop Linux Distros Implemented

Debian Family

**Raspberry Pi OS**

**Sparky Linux (Stripped down Debian implementation for reduced Resource demands)**

Ubuntu Family

**Xubuntu based on Ubuntu Server v18.04 LTS (Xfce Desktop)**

**Ubuntu Mate based on Ubuntu Server v20.04 (Mate Desktop)**

Arch Linux Family

**Manjaro Xfce Updated Desktop**

**Manjaro KDE Plasma Desktop**

**Arch Linux Xfce Legacy Desktop (Similar to Debian Buster v10 Xfce Desktop)**

4

# Objectives for Rpi4 Desktop Environment

**Implement a GUI Based Desktop Workstation to Develop ooRexx, NetRexx and BSF4ooRexx code**

**Create a Workstation that is faster than a Windows 10 Implementation**

**Workstation Image will be Easier and Faster to Backup than my Windows 10 Workstation**

**Complete Backup of System Image not Just The Files**

**Workstation will have enough Storage Space to Implement a Good Number of Development Software Products (Web Servers, Libre Office Suite, Editors, IDEs, Several Programming Languages in addition to REXX)**

5

## Hardware Utilized with Findings and Recommendation - I

### Raspberry Pi Board

Version 4 Model B with 4GB RAM Memory

### Keyboard and Mouse

Official Raspberry Pi Keyboard and Hub

Official Raspberry Pi Mouse

### Cases with Cooling Fans

Canakit Acrylic Premium Case – Available in Black, White or Clear (Fits together without use of any Screws)

Vilros Aluminum Case (4 Screws Used To Put Together Halves of Case Shell) Available Colors: Black and Silver

I Uniker Pi Enclosure for Pi 4 (Colors: White, Black and Clear; 4 Screws used to Fit R Pi Board to bottom half of Case)

6

## Hardware Utilized with Findings and Recommendation - II

### Fan Installation in Case

All of the 3 Cases provided Documentation for the GPIO pins to use for Connecting the Fan Wires.

I Uniker Pi Enclosure Case had options for Faster or Slower Fan Speed Installation

Canakit Kit Case Fan Snaps in to the Top of The Case without need for Screws

I Uniker Pi Enclosure and Vilros Aluminum Allow Cases provided 4 Screws for Fan Installation

The Vilros Case and I Uniker cases provided a Screwdriver for Installing the Very Small Screws

7

## Hardware Utilized with Findings and Recommendation - III

### Heat Sinks for the Raspberry Pi 4 Board

The I Uniker Pi Enclosure Case had 4 Aluminum Heat Sinks

The Canakit Premium Case had 3 Heat Sinks

The Vilros Case had 2 Heat Sinks (CPU and RAM Chips)

### Heat Dissipation

All of the 3 Cases had either 4 installed or installable Rubber Feet for the bottom of the case

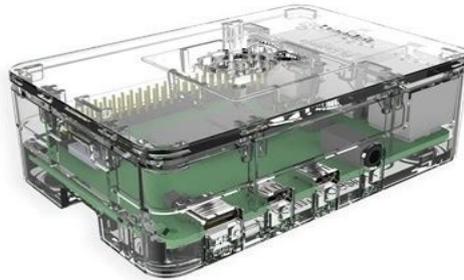
The Vilros Aluminum Case was coolest to the touch after extended use

All R Pi Cases performed well in a 86 degree Fahrenheit room

8

# Hardware Utilized with Findings and Recommendation - IV

## Canakit Premium Acrylic Case



9

# Hardware Utilized with Findings and Recommendation - V

## Canakit Fan and Heat Sinks



# Hardware Utilized with Findings and Recommendation - VI

## Vilros Heavy Duty Aluminum Case



11

# Hardware Utilized with Findings and Recommendation - VII

## I Uniker Pi Enclosure Acrylic Case



12

# Web Site References for Case Screenshots

## Canakit Premim Acrylic Case

<https://www.canakit.com/raspberry-pi-4-case-clear.html>

## Vilros Heavy Duty Aluminum Alloy Case

<https://vilros.com/products/vilros-raspberry-pi-4-basic-starter-kit-with-fan-cooled-heavy-duty-aluminum-alloy-case?variant=31425736310878>

## I Uniker Pi Enclosure Acrylic Case

<https://www.iunikerpi.com/raspberrypi-4-case-iuniker-raspberrypi-4-fan-abs-case-with-cooling-fan-raspberrypi-4-heatsink-simple-removable-top-cover-for-i-4-model-b-4b-p0058.html>

13

## Basic Starter Kit Recommendation

### Vilros Aluminum Alloy Case Basic Starter Kit (\$84.99 USD on Amazon)

#### Pros:

- Case has better heat dissipation over the Acrylic Cases
- Very Good Strong Fan that is fairly quiet compared to my Intel CPU PCs
- Getting Started Documentation is well done
- Screwdriver Provided was Great for Fan and Case Installation
- Power Supply Has On/Off Switch; Canakit Power Supply did not
- A lot of other Aluminum Alloy cases do not provide Heat Syncs or Fans

#### Cons:

- Very Small Screws; use the Screwdriver Provided with the Kit if you don't have a good quality small Phillips head screwdriver
- You have to Install the Fan and the Heat Syncs; not that Big of a Challenge
- Exercise caution when Installing the SD Card; it can slip into the Case
- Only 2 Heat Sinks for CPU and RAM Storage Chips

14

## Other Recommendations - I

### SD Card Storage

If you are planning to store a great deal of info on the Card; consider getting a 64GB or larger card. Most R Pi 4 kits currently have 32 or 64 GB cards

After installing nearly 2500 Debian Packages several of most SDXC Cards have utilized nearly 40% of the Storage space on a 64 GB Card.

Opt for a high quality micro SDXC UHS-I card for improved storage, durability and performance. R Pi 4 may not boot if a poor quality micro-SD card is used

Personally I used the Samsung EVO Plus or Select 64Gb cards for my storage

[Maker.Pro - What Micro SD Card is Best for a Raspberry Pi 4?](#)

15

## Other Recommendations - II

### Raspberry Pi 4 Board

Use the 4 GB or 8 GB Versions of the R Pi Board for an effective Desktop PC so that your memory intensive applications run well

Software Examples that require Memory

**Libre Office**

**Apache Web Server with Web Apps**

**Web Browsers such as Firefox and Chromium**

**MongoDB Server**

Raspberry Pi 4 8 GB Board Does Generate a bit more heat  
Fan cooling Recommended

16

## Other Recommendations - III

### Monitor

Personal taste and budget truly influence your choice here  
I decided to go with the Evciv Monitor as it was the right size (7 Inches Diagonal) and configuration for my limited desktop space (Currently about \$77 USD on Amazon)

### Amazon Link:

<https://www.amazon.com/dp/B07L6WT77H?tag=duckduckgo-d-20&linkCode=osi&th=1&psc=1>

What may influence your choice is your need for the size of your displayed visual information

17

## Other Recommendations - IV

### Monitor Recommendation Screenshot



18

# Other Recommendations - V

## Monitors to Evaluate:

### [Top 10 Best Raspberry Pi Monitors 2020 Review](#)

## Cabling:

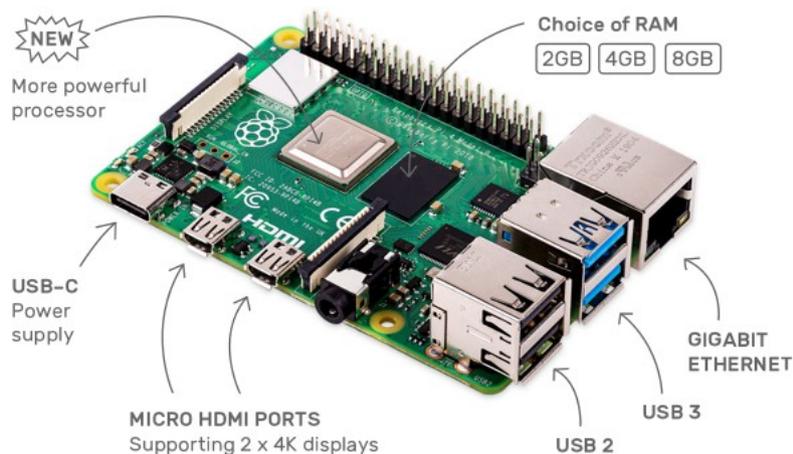
Use the leftmost smaller HDMI port immediate to the right of the Power Supply port to connect the Display HDMI cable

The rightmost smaller port did not yield any video output when I attempted to use that HDMI port for the EVICIV 7 Display

19

# Raspberry Pi 4 Model B Board

## Screenshot Source ([raspberrypi.org](https://www.raspberrypi.org))



20

# Linux Distros Selected and Rejected

## Criteria for Distro Selection

Stable Distro with a large Package Base; Easy to Install and Configure

Candidates Evaluated

**Recommended Choice for Distro Family Group ...**

**Raspberry Pi OS (formerly Raspbian) - Implemented**

**Sparky Linux - Implemented**

**Xubuntu for Raspberry Pi 4 - Implemented**

**Ubuntu Mate for Raspberry Pi 4 - Implemented**

**Manjaro Xfce Desktop - Implemented**

**Manjaro KDE Plasma Desktop - Implemented**

**Arch Linux Xfce Desktop - Implemented**

**Balena OS - Rejected**

**CentOS - Rejected**

**Alpine Linux - Rejected**

21

## Reasons for Linux Distros Rejected

### Balena OS

Concept for OS is based upon Docker Containers; currently my knowledge of Docker is too limited to consider a Docker container based OS solution. I do love the Balena Etcher Card Flash utility. More later.

### Alpine Linux

Difficult to implement a GUI Desktop Manager solution on the R Pi 4.

After efforts to create both a writeable /usr and /home directory, I could never accomplish both; just one or the other.

### CentOS

Their development of their existing Raspberry Pi 2/3 Distros has not evolved enough for a easy to install Raspberry Pi 4 solution. In the case of CentOS, only version 7 is available for the R Pi 4 with a limited base of outdated GPG signed packages. Their development repository does have a fair number of unsigned packages for installation.

22

## Linux Distros Selected - II

### Debian Family

Raspberry Pi OS (formerly Raspbian) – based on Debian Buster Distro

Sparky Linux – Debian based distro with Openbox Desktop Manager

### Ubuntu Family

Xubuntu – Updated Xfce Desktop based on Ubuntu Server 18.04 LTS

Ubuntu Mate – Mate Desktop based on Ubuntu Server 20.04 LTS

23

## Linux Distros Selected - III

### Arch Linux Family

Manjaro Linux Xfce Desktop – Based on Arch Linux v 20.05 – Updated Xfce Deskto

Manjaro KDE Plasma Desktop – Based on Arch Linux v 20.05 – KDE Plasma Desktop (a bit slow on a 4 GB R Pi4 board; more later)

Arch Linux – Xfce Legacy Desktop – Similar to Debian Buster v10.4 Xfce Desktop

24

# Debian Family Installs - Raspberry Pi OS

## Raspberry PI OS (Raspbian)

If you purchased a Raspberry Pi 4 Kit chances are you received a NOOBS SD Card (likely 32GB; go \$Bigger or go \$HOME)

NOOBS can also be downloaded using the Raspberry Pi Imager from the following link:

<https://www.raspberrypi.org/downloads/>

This page also contains links for copying images to an SD card

Currently, Raspberry Pi OS is armv71 32 bit architecture

Work is underway for a 64 bit architecture ([Beta available for download](#))

Desktop is one of the more stable implementations and response time on this distro is quite good

This is the easiest Linux distro to implement; good GUI response

25

# Debian Family Installs -- Sparky Linux

## Sparky Linux

V5.12 is the latest implementation based upon Debian Buster

armv71 architecture (32 Bit)

Openbox Desktop GUI is less polished than the Raspberry OS distro Desktop

Very fast and quite stable; good response on a 4 GB Raspberry Pi 4 board

As of this writing, the older Java OpenJDK v8 is still available for installation from the Sparky/Debian package repository

The Raspbian Version of the ooRexx .deb Package can be used to install ooRexx

This version requires a bit more work to implement vs. Raspberry Pi OS

Vast Library of Installable Packages for the Debian based distros

Fastest response time of all the implementations surveyed

26

# Arch Linux Family Installs - Manjaro Xfce

## Manjaro Linux Xfce Desktop

Manjaro Installs are 64 bit aarch Architecture

Manjaro Images are the easiest to Install Images

**Download the image file**

**Use Balena Etcher to burn the downloaded image to the SD card**

Link to Download Manjaro Xfce

<https://www.manjaro.org/downloads/arm/raspberry-pi-4/arm8-raspberry-pi-4-xfce/>

Balena Etcher can be used to transfer a downloaded image to a SD card

Website link for Balena Etcher (Downloads for Linux, Windows and Mac OS)

<https://www.balena.io/etcher/>

Current Version: 20.06 (July 2020)

Xfce Desktop works well on a 4GB Raspberry Pi 4B Board

27

# Arch Linux Family Installs - Manjaro KDE Plasma

## Manjaro Linux KDE Plasma Desktop

Manjaro Installs are 64 bit aarch Architecture

KDE Plasma is the Slickest Desktop IMHO; it is slower than the Xfce Version of Manjaro

Recommend getting the 8 GB R Pi4 for this distro

Manjaro Images are the easiest to Install Images

**Download the image file**

**Use Balena Etcher to burn the downloaded image to the SD card**

Link to Download Manjaro KDE Plasma

<https://www.manjaro.org/downloads/arm/raspberry-pi-4/arm8-raspberry-pi-4-kde-plasma/>

Balena Etcher can be used to transfer a downloaded image to a SD card

Website link for Balena Etcher

<https://www.balena.io/etcher/>

Current Version: 20.06 (July 2020)

KDE Desktop is sluggish on a 4GB Raspberry Pi 4B Board; Dolphin file Manager is buggy

28