2024 Rexx Language Association Symposium
Brisbane, Australia
By Tony Dycks
Last Update: March 2, 2024
Overview

- Objective of Presentation
- SBC Hardware for Build Server
- Linux OS Selection and Installation
- Setup of Basic Headless Environment on Win 10
- Linux OS First Time Remote Access Configuration
- Rexx Implementations for Build Server
- Linux SW Required to Build ooRexx Beta
- Linux SW Required to Build Regina
- Linux SW Required to Build BSF4ooRexx
- Mixing ooRexx / BSF4ooRexx Install with Regina Install
- Attempt to Create a Cross Architecture Build Environment
- Summary of Findings
- Future To Dos
- List of Web Based Resources
- Acknowledgments of Rexx Language Association Members
Objective of Presentation

- Create a Headless Single Purpose Server Environment to Build Rexx Software for Both 32 and Possibly 64 Bit Arm Linux Environments
- Use SBC Hardware that will Restrict Cost to $100 or Less
- Use Existing Windows 10 Workstation for Remote SSH Access to SBC Server
- Document Headless Linux OS Setup
- Document Headless Server Access on Windows 10
- Document How to for Native 32 Bit (armv7l) Build Processes
- Test to See if Different Bitness Executables can be Built on the Nano Pi NEO
Friendly Nano Pi NEO

- **CPU**: Allwinner H3, Quad-core Cortex-A7 Up to 1.2GHz
- **DDR3 RAM**: 512MB
- **Connectivity**: 10/100M Ethernet
- **USB Host**: Type-A x 1, 2.54 mm pin x 2
- **MicroSD Slot**: x 1
- **MicroUSB**: OTG, for power input
- **Debug Serial Port**: 4 Pin, 2.54 mm pitch pin header
- **Audio input/output Port**: 5 Pin, 2.0mm pitch pin header
- **GPIO**: 2.54mm pitch 36pin. It includes UART, SPI, I2C, IO etc
- **Power Supply**: DC 5V/2A
- **PCB Dimension**: 40 x 40 mm
- **Working Temperature**: -20°C to 70°C
- **Weight**: 14g (WITHOUT Pin-headers)
SBC Hardware for Build Server

Nano Pi NEO SBC

- $32.99 USD Amazon (Jan. 2024)
Nano Pi NEO SBC Aluminum Case with Heatsink Base

- $12.99 USD Amazon (Jan. 2024)
Additional Hardware with Amazon Prices:

- **Power Supply:**
  - CanaKit 5V 2.5A Raspberry Pi 3 B+ Power Supply/Adapter $9.95 USD

- **Ethernet 10/100 Mbps Connectivity:**
  - 15 Foot Amazon Basics RJ45 CAT-6 LAN Cable $7.19 USD

- **Disk Storage:**
  - Samsung EVO Select 64GB Micro SD Card $11.99 USD
Amazon.com Total Cost (under $100 USD):

<table>
<thead>
<tr>
<th>Hardware Product</th>
<th>$ Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nano Pi NEO 512Mb SBC</td>
<td>$32.99</td>
</tr>
<tr>
<td>Nano Pi NEO Metal Case</td>
<td>$12.99</td>
</tr>
<tr>
<td>Canakit 5 Volt 2.5 Amp Power</td>
<td>$9.95</td>
</tr>
<tr>
<td>Amazon Basics CAT-6 RJ45 Cable</td>
<td>$7.19</td>
</tr>
<tr>
<td>Samsung EVO Select 64GB MicroSD</td>
<td>$11.99</td>
</tr>
</tbody>
</table>

Sub-Total                                               | $75.11  |
Sales Tax                                               | $7.70   |
Total Cost                                               | $82.81  |
Linux OS Selection and Installation

- **Armbian Linux for Nano Pi NEO**
- Jammy 23.02 (Ubuntu) CLI
- Older Release due to OS Boot Issues with Linux Kernel v6
- Community Build; Not Officially Supported
- Image File to Download:
  - Armbian_23.02.2_Nanopineo_jammy_current_5.15.93.img.xz
- Why Ubuntu? OpenJDK Version 8 Debian Install Package still Available
Linux OS Selection and Installation

- **Burn Downloaded Image File to MicroSDXC**
- Use Bit Accurate Imaging Utility to Flash OS to MicroSD Card
- **Recommended Tool**: Balena Etcher for Windows or Linux Intel
- Armbian Image Contains Base SSH Image to facilitate Remote Access
- No Need to Mess with GPIO pin or USB Port to Connect Serial Display for Data Entry
- Use Remote Access Tools such as PuTTY or Terra Term on Windows
Setup of Basic Headless Environment on Win 10

- Local Area Network IP Connectivity Detection:
  - Use Tool such as Angry IP Scanner
  - Cross Platform Friendly: Java Based
  - https://angryip.org

- Remote SSH Access Tool for Windows 10-11:
  - PuTTY – https://putty.org
  - Debian Packages Also Available for Linux Distros
  - Download Link: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
  - Latest Release: v0.80
Setup of Basic Headless Environment on Win 10

- PuTTY Screenshot
Linux OS First Time Remote Access Configuration

- Initial Image **root** User Password: 1234
- System Will Prompt for The Following on First Login:
  - Change of **root** User Password
  - Creation of **New User Id** with **sudo** Privileges
  - Entry of Formal **Name** (Doesn’t need to be too formal)
  - Choice of use of **bash** or **zsh** Command Shell
  - Choice of System **Locale** and **Data Encoding**
  - Choice of Timezone
Following the Initial Configuration Additional Desired Configuration:

- Change and Verify the Computer Hostname:
  - # hostnamectl set-hostname MyHostName
  - # hostnamectl

- Apply Any System Updates (Kernel Updates are Disabled for Stability):
  - # apt update
  - # apt upgrade

- Install Security Related Debian Packages for Firewall and Anti-virus
  - # apt install ufw clamav

- For CLI Environment apt installs Will Be Used to Install all .deb Packages
Rexx Implementations for Build Server

- **Rexx Software for Builds:**
  - ooRexx v5.1 Beta from Subversion Checkout
  - BSF4ooRexx850 from Subversion Checkout
  - Regina Rexx v3.9.6 from Downloaded Source Tarball
  - Rexx/CURL v2.1.0 from Downloaded Source Tarball

- **Linux Software Package Pre-Requisites for All The Products**
  - buildessential (Already installed for Ubuntu)
  - The gcc Compiler Suite will be used for Compilation of All Rexx Products
Rexx Implementations for Build Server

- Linux Software Package Pre-Requisites for ooRexx
  - $ sudo apt install libncurses-dev
  - $ sudo apt install subversion
  - $ sudo apt install cmake
- Linux Software Package Pre-Requisites for BSF4ooRexx850
  - A Java Runtime or JDK Version 8 or Later
    - Used for this Presentation: java-8-openjdk (still available in Ubuntu 22.*)
  - subversion
  - 32 Bit ARM ooRexx build or Debian package install
Rexx Implementations for Build Server

- **Linux Software Package Pre-Requisites for Regina Rexx**
  - $ sudo apt install pkgconfig
  - $ sudo apt install fakeroot (If Building .deb files)

- **Linux Software Package Pre-Requisites for Rexx/CURL**
  - A Regina Installation or The following Debian Packages
    - $ sudo apt install regina-rexx
    - $ sudo apt install libregina
    - $ sudo apt install libregina-dev
  - $ sudo apt install libcurl4-ssl-dev
  - $ sudo apt install pkgconfig
  - $ sudo apt install fakeroot (If Building .deb files)
  - $ sudo apt install debhelper (If Building .deb files for Rexx / CURL)
Regina and ooRexx can co-exist on the same Linux Environment by Separating the Binary and Library Environments

My Strategy for Sequence and Configuration Steps and Options:

1) Build ooRexx 5.1 Beta Source to Reside in `/usr/local`
2) Install Java OpenJDK 8 Debian Package
3) Install BSF4ooRexx850 (Defaults Install to Linux Base Directory: `/opt/BSF4ooRexx850`)
4) Configure and Build Regina Source to Reside in `--prefix=/usr`
5) Configure and Build Rexx/CURL Source to Reside in `--prefix=/usr`
Mixing ooRexx / BSF4ooRexx Install with Regina Install

- Sample Regina Rexx Source Build How Tos
  - Download Regina Rexx v3.9.5 Source Gzipped Tarball
    - Extract tarball to /usr/local/ Directory
    - $ cd /usr/local/regina-rexx-3.9.5
    - $ ./configure --prefix=/usr
    - Check End Result of Configure to Confirm SW File Locations
    - $ make
    - $ make deb (Optional: If Creating .deb Package)
    - $ make install (Or …)
    - Install the libregina and regina-rexx Debian Binary Packages
Mixing ooRexx / BSF4ooRexx Install with Regina Install

- **Sample Rexx / CURL Source Build How Tos**
  - Download Rexx / CURL v2.1.0 Source Gzipped Tarball
    - Extract tarball to `/usr/local/` Directory
    - `$ cd /usr/local/rexxcurl-2.1.0`
    - `$ sudo ./configure –prefix=/usr --with-rexx=regina`
    - Check End Result of Configure to Confirm SW File Locations
    - `$ sudo make`
    - `$ sudo make deb` *(Optional: If Creating .deb Package)*
    - `$ sudo make install` *(Or …)*
    - Install the 3 Debian Binary Packages
Attempt to Create a Cross Architecture Rexx Build Environment

- GCC Compiler Tools for Alternate ARM SBC CPU Bitness
  - For the Nano Pi NEO in this presentation the preceding slides have detailed the 32-Bit (armv7l) GCC and G++ Tools (The build-essential Debian Package)
  - For 64 Bit (aarch64) One Would Install the Following Debian Packages:
    - `sudo apt update`
    - `sudo apt install gcc-arm-none-eabi` (bare metal binary)
    - `sudo apt install gcc-arm-linux-gnueabihf` (not available for the 32 Bit Nano Pi NEO Armbian Ubuntu distro)
    - `sudo apt install gcc-aarch64-linux-gnu` (not available for the 32 Bit Nano Pi NEO Armbian Ubuntu distro)
Attempt to Create a Cross Architecture Rexx Build Environment

- Changes to gcc Compiler Steps – A Simple C Program Example
  - To Build a helloworld.c Program for the Same Architecture ...
    - $ gcc -o helloworld helloworld.c
  - To Build helloworld.c for a 64-Bit ARM Architecture ...
    - $ aarch64-linux-gnu-gcc -o helloworld helloworld.c
  - To Verify the Architecture Bitness use the Command ...
    - $ file <executable-path>
  - Some Examples:
    - $ file ./helloworld
    - $ file /usr/bin/regina
    - $ file /usr/local/bin/rexx
Attempt to Create a Cross Architecture Rexx Build Environment

- 64-Bit Natively Built Examples:
  
  - OoRexx 5.0.0 r12523 Native build for aarch64 Architecture:
  - tonyd@LedZeppelin:~$ file /usr/local/bin/rexx
  - /usr/local/bin/rexx: ELF 64-bit LSB pie executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=c7ee09c0ef3b72052a937d2ebfd066f8cd0d435b, for GNU/Linux 3.7.0, not stripped
  - Regina 3.9.5 Debian Binary Package Installation for aarch64 Architecture:
  - tonyd@LedZeppelin:~$ file /usr/bin/regina
  - /usr/bin/regina: ELF 64-bit LSB pie executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=78ad3abf8e247064d456f87e6de877617db2bba6, for GNU/Linux 3.7.0, stripped
Attempt to Create a Cross Architecture Rexx Build Environment

- 32-Bit Natively Built Examples on Nano Pi NEO:
  - ooRexx Native build for armv7l Architecture:
  - tonyd@Easter:~/gcc/source$ file /usr/local/bin/rexx
  - /usr/local/bin/rexx: ELF 32-bit LSB pie executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf.so.3, BuildID[sha1]=ea5ba2cf83b5fd3baa228a20107cacb66cafb354, for GNU/Linux 3.2.0, not stripped
  - Regina 3.9.5 Source Built Debian Binary Package Installation for armv7l Architecture:
  - tonyd@Easter:~/gcc/source$ file /usr/bin/regina
  - /usr/bin/regina: ELF 32-bit LSB pie executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf.so.3, BuildID[sha1]=ebe6adf8f98d2713e4f293ba3501ee957f7d63ac, for GNU/Linux 3.2.0, stripped
Attempt to Create a Cross Architecture Rexx Build Environment

- Hello World Program Built Examples on Nano Pi NEO:
  - Migrated Object from aarch64 Linux Environment will not run due to Architecture Differences:
    - tonyd@Easter:~/gcc/source$ file ./helloworld
      ./helloworld: ELF 64-bit LSB pie executable, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=513e8c87c59c8051a8f9ef065c89af49ab1b759f, for GNU/Linux 3.7.0, not stripped
    - tonyd@Easter:~/gcc/source$ ./helloworld
      -bash: ./helloworld: cannot execute binary file: Exec format error
  - Native Compilation for armv7l Architecture runs Okay:
    - tonyd@Easter:~/gcc/source$ file ./helloworld
      ./helloworld: ELF 32-bit LSB pie executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf.so.3, BuildID[sha1]=b1cf4551a387171e0ff0a30d8b103d64d285e540, for GNU/Linux 3.2.0, not stripped
Hello World Program Built on Nano Pi NEO:

- Executable Copied to USB Thumb Drive and Copied to Ubuntu 64 Bit Environment
- Make Sure the Migrated Binary has Executable Privileges
  - $ chmod +x ./helloworld
- Verify The Architecture is ELF 32-Bit:
  - $ file ./helloworld
- Attempt to Run Binary Results:
  - No such file or directory Error Message
Summary of Findings

- The **Nano Pi NEO** Runs Exceptionally Cool when Installed in the Metal Case with the Heatsink; No Need for a Fan
- If **Operating Temperature Reaches Over 70C** Then Additional Cooling will be Necessary
- Since the SBC **Does Not Have a Graphics Processing Unit** (GPU); No Windows Based Programs will Run Successfully
- As an Alternative Use a **Curses Based Programs** or Shell Based Programs can be Used
- **Examples:** Shell Dialogs, Ncurses Based Programs, Nano Text Editor, Midnight Commander, CLI Shell Programs
- Other Linux Distros that Will Run on the Nano Pi NEO:
  - Armbian Linux Debian 11 Buster CLI (Linux Kernel v5)
  - Debian Bookworm 12 CLI
  - **Diet Pi for Nano Pi NEO** (Debian Bookwork armv7l)
- Avoid the Red Hat Family of Linux Distros due to Boot Issues (Fedora, Oracle, Alma Linux)
- Migrated Natively Built aarch64 and armv7l Architecture Programs Will Not Run on the Other Bitness Linux Distros
Summary of Findings

Sample PuTTY Console Screenshot – Completed Login Message
Summary of Findings

Sample PuTTY Console Screenshot – neofetch Utility Output
Summary of Findings

Sample PuTTY Console Screenshot – mc Utility Output
Summary of Findings

- For Transfer of Files from Other Storage:
  - Utilize a USB Thumb Storage Drive on the One USB Port
  - Setup Example from Command Prompt:
    - `$ cd /mnt`
    - `$ sudo mkdir usb1`
    - `$ cd $HOME`
    - `$ sudo mount /dev/sda1 /mnt/usb1`
    - `$ cd /mnt/usb1`
    - `$ mc (Use Midnight Commander to Transfer Files from Left Pane to Right Pane)`
  - Attempted to Use FileZilla to Transfer Files from Windows 10 PC; Unable to Connect via SFTP (Possibly due to Firewall Settings)
Future To Dos

- For Repeated Builds and File Transfers of Rexx Product Files to and from this SBC Server:
  - Implement a Secured FTP Server Connection to SBC
  - Implement an Automated Build Process with Linux SW Components such as:
    - Subversion
    - Pkgconfig
    - Make and CMake
    - GCC
    - Rexx / CURL
    - Bash Shell Scripting
    - Kron
  - Implement All Necessary Software to Cross Compile ARM 64 Bit Rexx Object Files
    - **Challenges**: Architecture Bitness of Dependent Binary Packages; Differences in USR Library Path Conventions
  - Implement Debian Packaging to Create Binary Installation Package
List of Web Based Resources

- Friendly Elec Nano Pi NEO Wiki:

- Armbian OS CLI Alternative Image Download:
  - http://xogium.performanceservers.nl/archive/nanopineo/archive/

- Angry IP Scanner Download for Windows 10:
  - https://angryip.org/download/#windows

- PuTTY SSH Remote Access Software: https://putty.org

- Diet Pi Linux Distro for Nano Pi NEO (based on Debian 12 Bookworm):

- Uncomplicated Firewall Configuration:
Acknowledgements of Rexx Language Association Members

- Dr. Rony Flatscher for his Continued Support of the ARM Library Maintenance for BSF4ooRexx
- Mark Hessling for his Continued Support and Maintenance of Regina Rexx and Rexx / CURL
- P.O. Jonsson for his Continued Support and Maintenance of the ooRexx ARM SBC Builds
- Rene` Jansen for his Instruction on How to Build ooRexx from Subversion Source
- Howard Fosdick for his Wrox Press Rexx Programmer’s Reference Book