Installing Slackware on the Raspberry Pi

Platform	AArch64/ARM64
Hardware Model	Raspberry Pi 4/400
Document Version	1.07,Apr 2025
Author	Stuart Winter <mozes@slackware>
Contributors	Brenton Earl <el0226@slackware> (R&D for the RPi4 Hardware Model)

Supporting the Slackware ARM Project

If you like what we're doing here, please consider becoming a patron.

Supported Raspberry Pi versions

At the moment, only the Raspberry Pi 4 is officially supported.

The Raspberry Pi 400 has been validated as working by the user community.

Video Tutorial

This tutorial is also available in video form.

The video tutorial demonstrates the original installation approach where the Slackware installation media was separate. A single Slackware Installer image is provided that contains all of the media, so it's easier than shown in the video.

Help / Support

Please post questions to the Slackware ARM forum.

Caveats

Fragility due to upstream support

The support within the proper Linux Kernel is fragile due to the development model of the Raspberry Pi company. You may need to switch to the RPi Kernel fork for the best experience. Instructions on how to do this are included in the post installation section of this document.

Installation Lifecycle

The Installation consists of a number of distinct stages:

1. Acquiring all required hardware
2. Setup of the Raspberry Pi hardware
3. Downloading and Verifying the Slackware assets
4. Writing the Slackware Installer to the Micro SD card
5. Booting the Slackware Installer
6. Installing Slackware
7. Completing the installation
8. Booting the Slackware OS
9. Post installation configuration and tweaks

Requirements

Hardware Setup

If you haven't yet assembled your Raspberry Pi 4, you may want to review the hardware setup guide. The configuration outlined there was used as the basis for developing this installation guide.

Notes on storage setup

It is possible to create a 4th partition on the Micro SD card ('MMC') which can house the Slackware OS. However, this is not recommended due to the longevity of this type of storage - particularly if it has high I/O. If you want to install the OS to the MMC card, it's preferable to use the "bare" Slackware Installer image rather than the standard 'AiO' (All in One) Installer (which is what this Installation guide covers). Using the `bare` Installer means you will need to install over the network or from some locally attached storage.

Computing / Network Environment

Item	Specification	Notes
Host Computer: an Internet-connected computer running an existing Linux distribution	The Host Computer needs approximately **5GB free storage** to download the required software assets. **You must be able to obtain *root* access to this Host computer**.	This will be used to download the Slackware distribution from the Internet and to write the Slackware Installation media to the Micro SD card.

Downloading the Slackware assets

In this section, we'll prepare the Linux Host Computer to receive and download the Slackware assets required for the installation.

1. Downloading the Slackware Linux AArch64 Distribution and Installation Assets

The '$' prefixes in the commands indicates the shell prompt - it's not to be typed/copied

Open a shell on the Linux Host Computer.

Prepare a directory to hold and serve the Slackware Distribution

We'll download the Slackware Linux distribution into a directory named 'slackware'.

```
$ cd ## this returns to the root of your home directory
$ mkdir slackware
$ cd slackware
```

Determine where you are within the Host Computer's Filesystem

```
$ pwd
/home/mozes/slackware
```

Note the directory location returned - you'll need this later

Installing the Slackware ARM GPG key

The Slackware ARM GPG key will be used to verify your downloads.

```
$ curl -sSL https://www.slackware.com/infra/keys/arm/GPG-KEY | gpg --import
-
```

Set the model of Raspberry Pi

For the Raspberry Pi 4:

```
$ HWM=bcm2711_rpi4
```

Set the version of Slackware AArch64 to download

At the time of writing, the only version available is 'current'.

```
$ SLKVER=current
```

Set the Internet media distribution server

If you are using a mirror server rather than the master Slackware ARM server, set it here. The format is: <hostname>::<rsync module name>

```
$ SLKSRV=ftp.arm.slackware.com::slackwarearm
```

Download the Slackware Linux Installer

```
$ rsync -PavL $SLKSRV/platform/aarch64/bootware/installer-
aio/slackwareaarch64-${SLKVER}/${HWM}-aio-slackwareaarch64-
${SLKVER}.img.xz.asc slkaio.img.xz.asc
$ rsync -PavL $SLKSRV/platform/aarch64/bootware/installer-
aio/slackwareaarch64-${SLKVER}/${HWM}-aio-slackwareaarch64-${SLKVER}.img.xz
slkaio.img.xz
```

The Slackware Installer images are approximately 5 GBytes in size.

Verify the Slackware Installer image

Verify the digital signature of the Slackware Installer:

```
$ gpg --verify slkaio.img.xz.asc
```

As the images are large, verification may take a minute or two.

The output will be similar to this. You are looking for 'Good signature from Slackware ARM…'

```
gpg: assuming signed data in `slkaio.img.xz'
gpg: Signature made Wed 24 Nov 2021 06:07:44 PM BST
gpg:                using RSA key F7ABB8691623FC33
gpg: Good signature from "Slackware ARM (Slackware ARM Linux Project)
<mozes@slackware.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the
owner.
Primary key fingerprint: 36D3 7609 2F12 9B6B 3D59  A517 F7AB B869 1623 FC33
```

If you see 'BAD signature' you should re-download as it may have become corrupt. If this doesn't help, drop a note to the Slackware ARM forum

Writing the Slackware assets to the Micro SD

Elevate yourself to root

On your Host Computer, obtain root:

The # prefix indicates that you're using the **root** user - it's not to be typed in!

```
$ su -   ## Note the hyphen - it's required
```

Write the Slackware Installer to the Micro SD card

Check what block devices are present

Prior to inserting the Micro SD Card into the USB adapter, we need to see what's already present within the OS so that we can easily locate our Micro SD card:

```
# lsblk -d
NAME MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda    8:0    0 465.8G  0 disk
```

As you can see, this Host Computer there is a single storage device - *sda*.

Insert the Micro SD card into your USB Card Reader and connect the adapter to a free USB port on the Host Computer

Run lsblk again:

```
# lsblk -d
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda      8:0    0 465.8G  0 disk
sdc      8:32   1    58G  0 disk
sdd      8:48   1     0B  0 disk
```

As you can see, *sdc* is 58 GBytes in size. This is the Micro SD card (in this example, it's labeled as '64GB' on the exterior of Micro SD card).

If your Micro SD card has existing partitions, you will not see them surfaced in this list - use *lsblk -b* to view them.

You'll also observe the presence of *sdd* - often the USB adapter itself obtains a block device. You can ignore this as it's 0Bytes.

Write the Slackware Installer to the Micro SD card

Enter the directory into which the Slackware assets were downloaded previously:

```
# cd ~mozes/slackware ## Substitute with the path you noted earlier
# xzcat slkaio.img.xz | dd status=progress bs=4M iflag=fullblock of=/dev/XXX
## Substitute/dev/XXX with the correct block device
# sync
```

Remove the MicroSD card from the Host Computer

You may now disconnect the USB adapter/remove the Micro SD card from the Host Computer.

Logout from the root user

We no longer need to use the Host Computer, so you can logout of the root shell.

```
# logout
```

Installing Slackware

To proceed, you must have:

- Connected the storage to the Raspberry Pi 4
- Connected the HDMI monitor
- Connected the keyboard (and optionally, mouse)
- Optionally connected the Ethernet cable to set the date via NTP from the Internet
- Inserted the Micro SD card containing the Slackware Installer

Encrypted storage

If you'd like to encrypt your storage, check the Disk Encryption Guide.

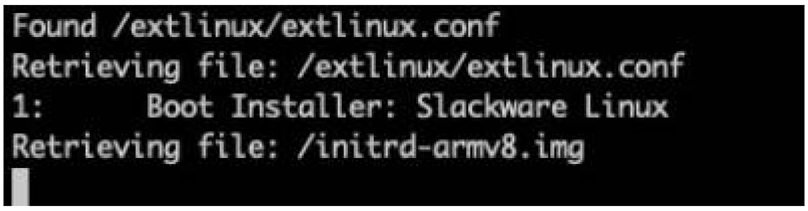
Begin installation

Disconnect any USB storage devices that aren't required for the OS installation

Due to bugs in the upstream Linux Kernel, the screen may blank every few seconds. This is normal and can be resolved by switching to the Raspberry PI Kernel fork (detailed later in this guide).

Power on the Raspberry Pi 4

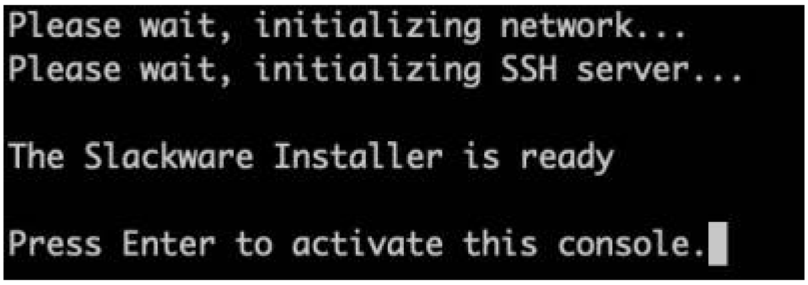
Apply power to the Raspberry Pi and after a few seconds, you will see the following on screen:



```
Found /extlinux/extlinux.conf
Retrieving file: /extlinux/extlinux.conf
1:      Boot Installer: Slackware Linux
Retrieving file: /initrd-armv8.img
```

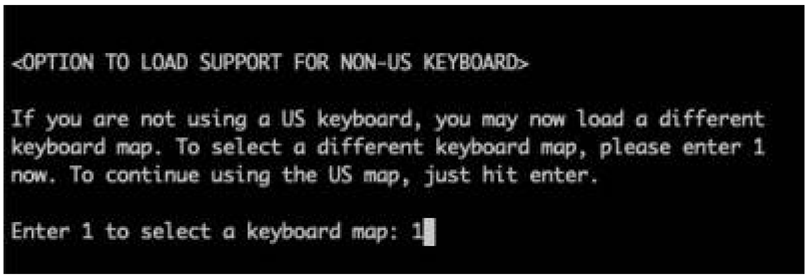
It takes several seconds to load and boot the installer, and it may take several seconds more for any further output to appear on the HDMI monitor. The process of obtaining an IP address via DHCP can also delay the ability to interact with the Installer.

Once an IP address has been obtained, you will be presented with a prompt. Press **ENTER**

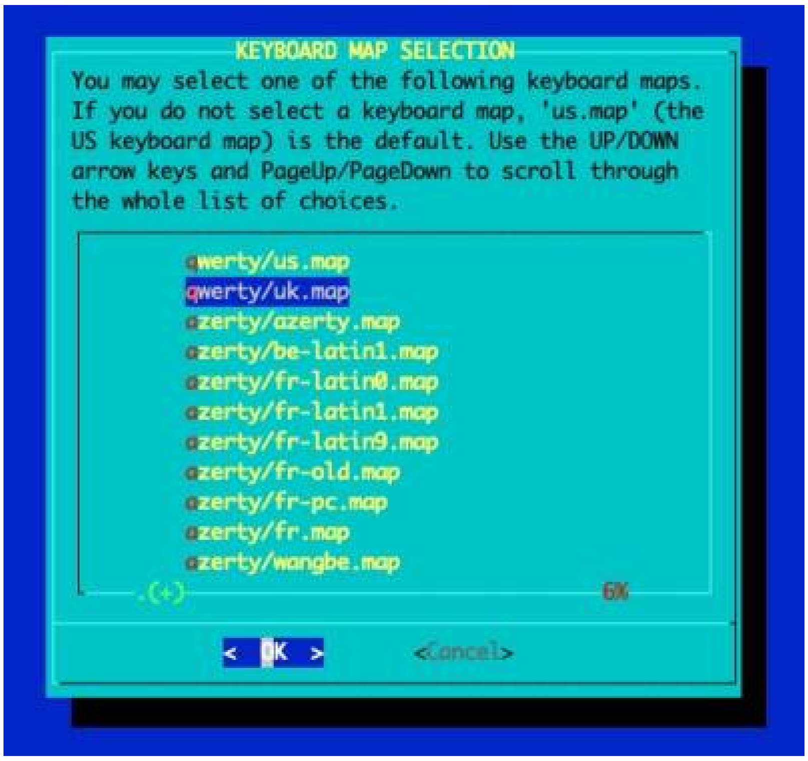


```
Please wait, initializing network...
Please wait, initializing SSH server...

The Slackware Installer is ready

Press Enter to activate this console.
```

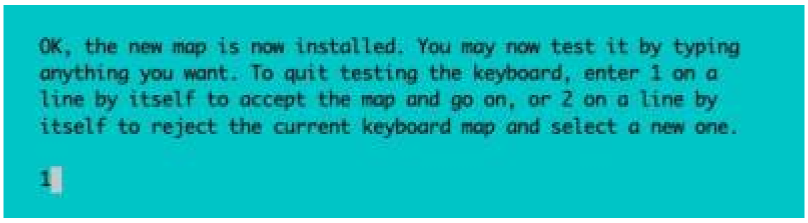
Set the keymap



```
<OPTION TO LOAD SUPPORT FOR NON-US KEYBOARD>

If you are not using a US keyboard, you may now load a different
keyboard map. To select a different keyboard map, please enter 1
now. To continue using the US map, just hit enter.

Enter 1 to select a keyboard map: 1
```



```
KEYBOARD MAP SELECTION
You may select one of the following keyboard maps.
If you do not select a keyboard map, 'us.map' (the
US keyboard map) is the default. Use the UP/DOWN
arrow keys and PageUp/PageDown to scroll through
the whole list of choices.

        qwerty/us.map
        qwerty/uk.map
        azerty/azerty.map
        azerty/be-latin1.map
        azerty/fr-latin0.map
        azerty/fr-latin1.map
        azerty/fr-latin9.map
        azerty/fr-old.map
        azerty/fr-pc.map
        azerty/fr.map
        azerty/wangbe.map
  .(+)                                  6%

          <  OK  >        <Cancel>
```



```
OK, the new map is now installed. You may now test it by typing
anything you want. To quit testing the keyboard, enter 1 on a
line by itself to accept the map and go on, or 2 on a line by
itself to reject the current keyboard map and select a new one.

1
```

Font size

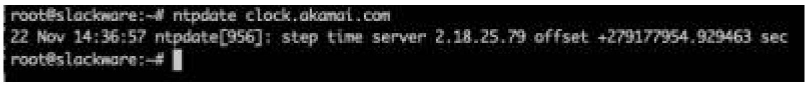
If you're using a smaller monitor, such as one with a screen size less than 20 inches, it may be necessary to adjust the console font size to ensure that menus and other interface elements fit correctly on the screen. If so, type this into the shell prompt:

```
setfont ter-v18n
```

Set the date/time

Even if you have a battery pack for the RTC (Real Time Clock), the date on your system may be incorrect. We will sync the date from a highly-available NTP server:

```
ntpdate clock.akamai.com
hwclock -w
```



```
root@slackware:~# ntpdate clock.akamai.com
22 Nov 14:36:57 ntpdate[956]: step time server 2.18.25.79 offset +279177954.929463 sec
root@slackware:~#
```

Setup disk partitions

For this installation a basic partitioning scheme will be created.

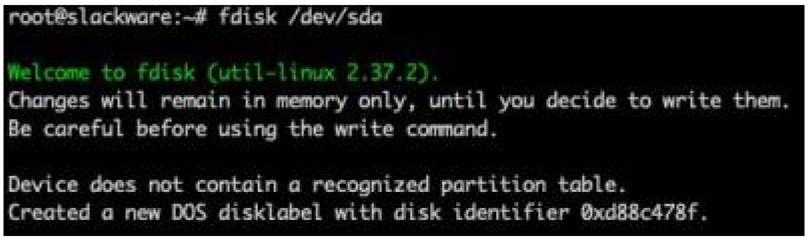
Partition

Partition number	Device name	Size	Purpose
1	/dev/sda1	4GB	Swap
2	/dev/sda2	Rest of storage	OS root ('/') partition

/boot will reside on the Micro SD card and is automatically configured by the Slackware Installer

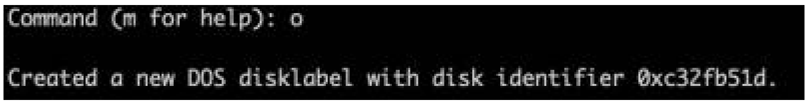
Open fdisk against the /dev/sda block device. In this guide, /dev/sda will be your primary storage, and in this guide is the SSD connected to the USB adapter.

```
fdisk /dev/sda
```



```
root@slackware:~# fdisk /dev/sda

Welcome to fdisk (util-linux 2.37.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xd88c478f.
```

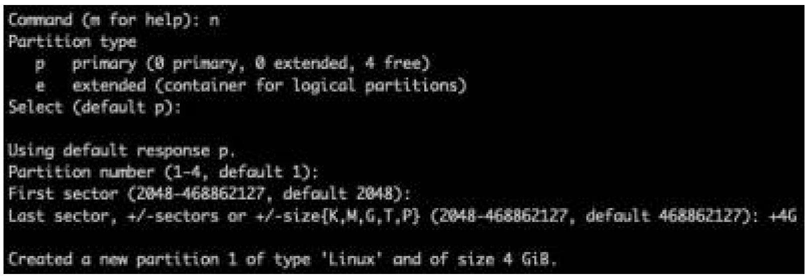
Clear an existing partition table: Press 'o' to clear the partition table



```
Command (m for help): o

Created a new DOS disklabel with disk identifier 0xc32fb51d.
```

Create the Swap partition:

Type 'n' for new partition:



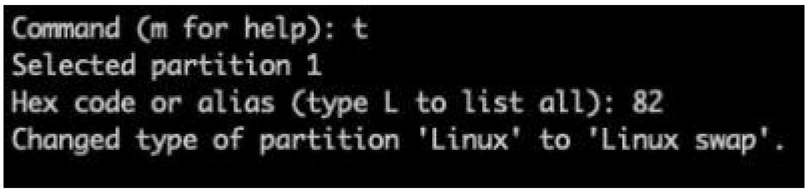
```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-468862127, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-468862127, default 468862127): +4G

Created a new partition 1 of type 'Linux' and of size 4 GiB.
```

Type 'p' for primary partition type:

Press ENTER for the 'First sector'

Type '+4G' for the 'Last Sector'/size:

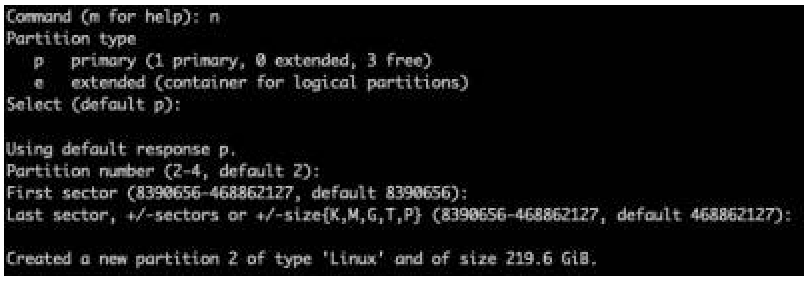
Change the partition type to 'Swap'. Type 't' then hex code '82':



```
Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 82
Changed type of partition 'Linux' to 'Linux swap'.
```

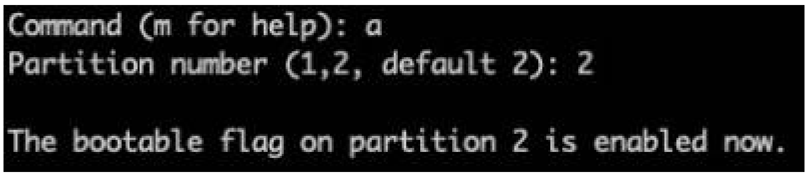
Create the partition for the root filesystem ('/'):

Type 'n' for new partition. Press ENTER to accept the defaults - this will create partition 2 as the maximum size available.



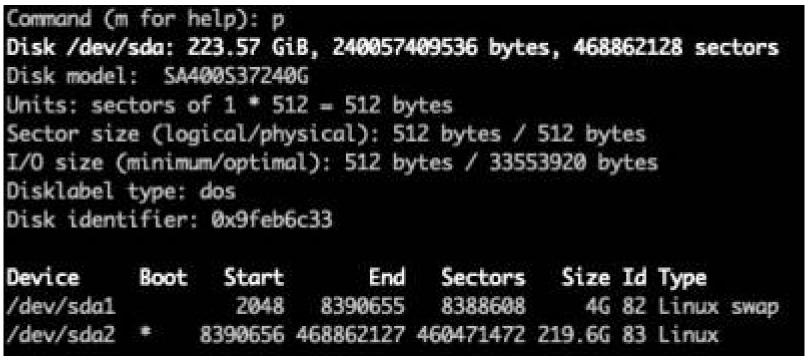
```
Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (2-4, default 2):
First sector (8390656-468862127, default 8390656):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (8390656-468862127, default 468862127):

Created a new partition 2 of type 'Linux' and of size 219.6 GiB.
```

Type 'a' to mark the root partition (number 2) as bootable Type '2' to select partition 2.



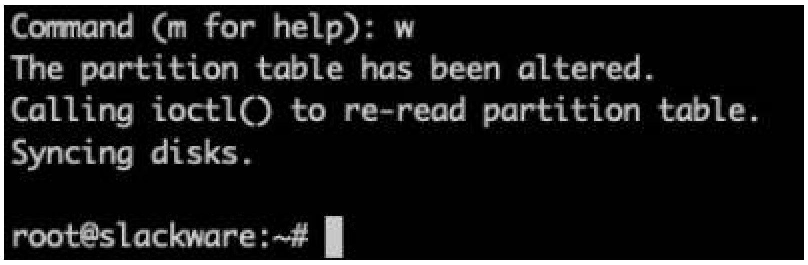
```
Command (m for help): a
Partition number (1,2, default 2): 2

The bootable flag on partition 2 is enabled now.
```

Type 'p' to print to view the partition table.



```
Command (m for help): p
Disk /dev/sda: 223.57 GiB, 240057409536 bytes, 468862128 sectors
Disk model:  SA400S37240G
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 33553920 bytes
Disklabel type: dos
Disk identifier: 0x9feb6c33

Device     Boot   Start       End   Sectors   Size Id Type
/dev/sda1          2048   8390655   8388608     4G 82 Linux swap
/dev/sda2  *    8390656 468862127 460471472 219.6G 83 Linux
```

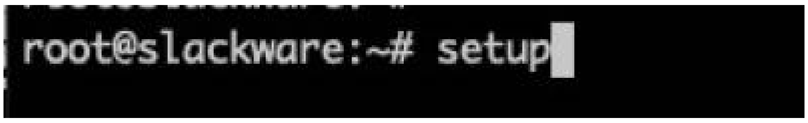
Type 'w' to write the partition table:



```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@slackware:~# 
```

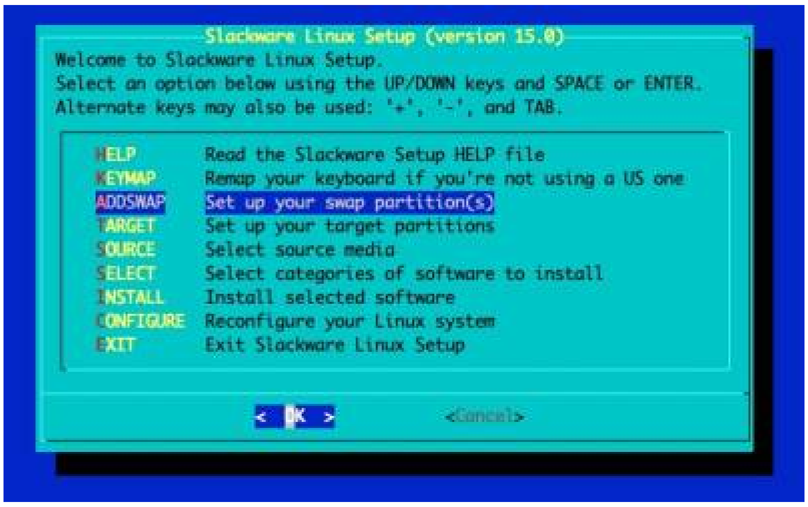
fdisk will now exit.

Load the Setup menu

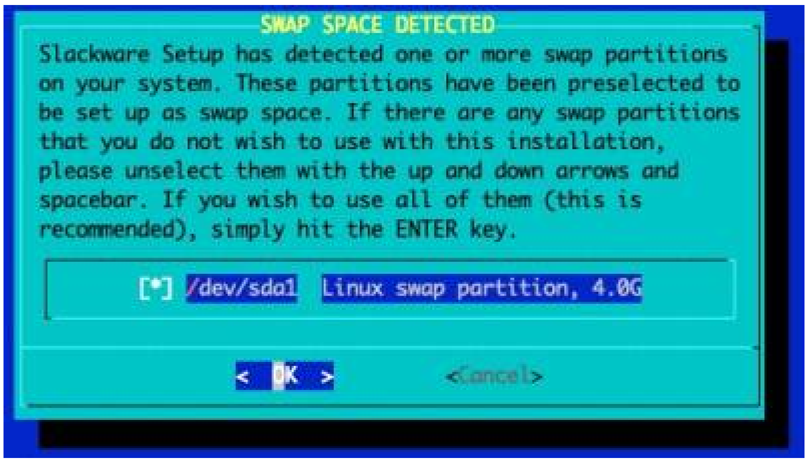


```
root@slackware:~# setup
```

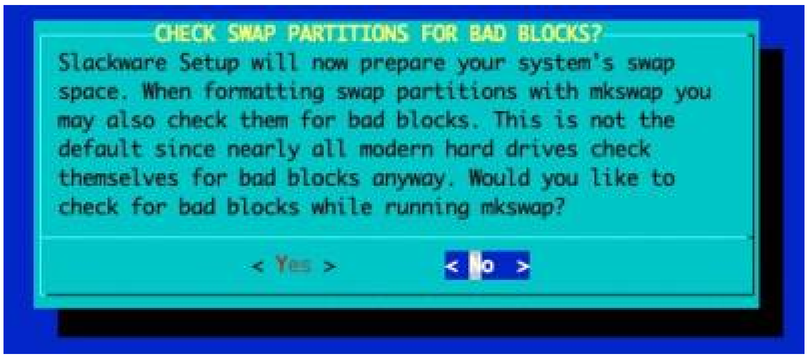
Setup Swap partition



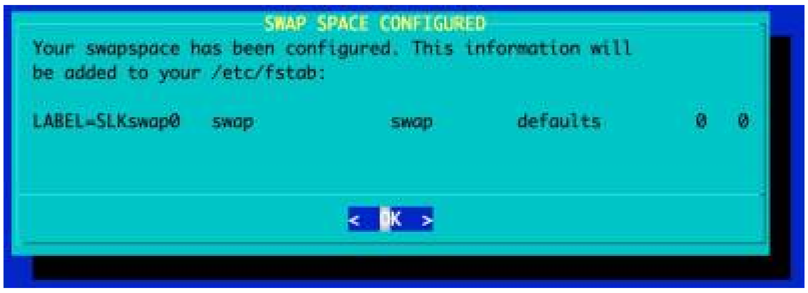
```
Slackware Linux Setup (version 15.0)
Welcome to Slackware Linux Setup.
Select an option below using the UP/DOWN keys and SPACE or ENTER.
Alternate keys may also be used: '+', '-', and TAB.

HELP        Read the Slackware Setup HELP file
KEYMAP      Remap your keyboard if you're not using a US one
ADDSWAP     Set up your swap partition(s)
TARGET      Set up your target partitions
SOURCE      Select source media
SELECT      Select categories of software to install
INSTALL     Install selected software
CONFIGURE   Reconfigure your Linux system
EXIT        Exit Slackware Linux Setup

        < OK >          <Cancel>
```



```
SWAP SPACE DETECTED
Slackware Setup has detected one or more swap partitions
on your system. These partitions have been preselected to
be set up as swap space. If there are any swap partitions
that you do not wish to use with this installation,
please unselect them with the up and down arrows and
spacebar. If you wish to use all of them (this is
recommended), simply hit the ENTER key.

    [*] /dev/sda1  Linux swap partition, 4.0G

        < OK >          <Cancel>
```

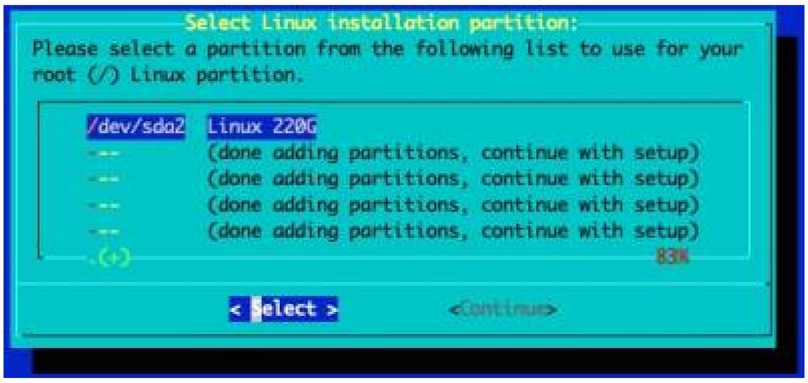


```
CHECK SWAP PARTITIONS FOR BAD BLOCKS?
Slackware Setup will now prepare your system's swap
space. When formatting swap partitions with mkswap you
may also check them for bad blocks. This is not the
default since nearly all modern hard drives check
themselves for bad blocks anyway. Would you like to
check for bad blocks while running mkswap?

        < Yes >         < No >
```

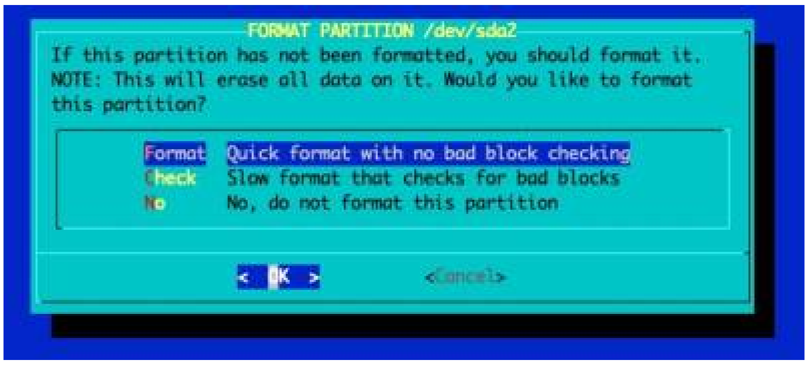


```
SWAP SPACE CONFIGURED
Your swapspace has been configured. This information will
be added to your /etc/fstab:

LABEL=SLKswap0   swap           swap        defaults        0   0

                         < OK >
```

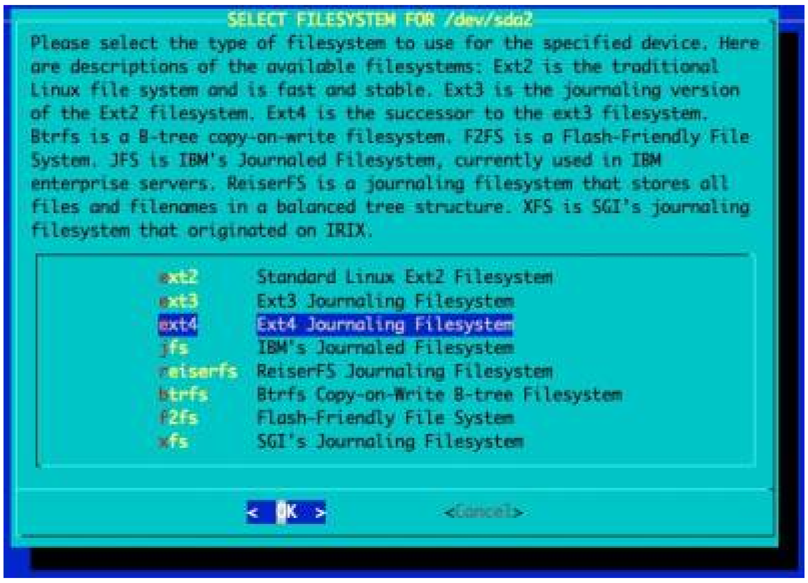
Select and format the partition for the OS' root file system



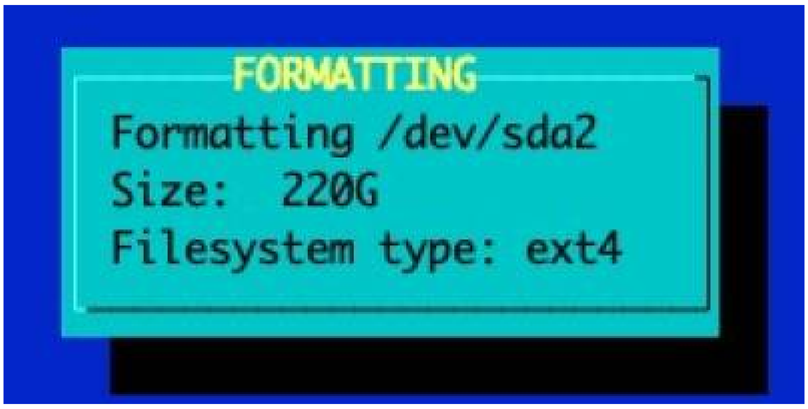
```
Select Linux installation partition:
Please select a partition from the following list to use for your
root (/) Linux partition.

    /dev/sda2  Linux 220G
    ---        (done adding partitions, continue with setup)
    ---        (done adding partitions, continue with setup)
    ---        (done adding partitions, continue with setup)
    ---        (done adding partitions, continue with setup)
    (+)                                                83%

           < Select >          <Continue>
```



```
FORMAT PARTITION /dev/sda2
If this partition has not been formatted, you should format it.
NOTE: This will erase all data on it. Would you like to format
this partition?

        Format  Quick format with no bad block checking
        Check   Slow format that checks for bad blocks
        No      No, do not format this partition

             < OK >          <Cancel>
```



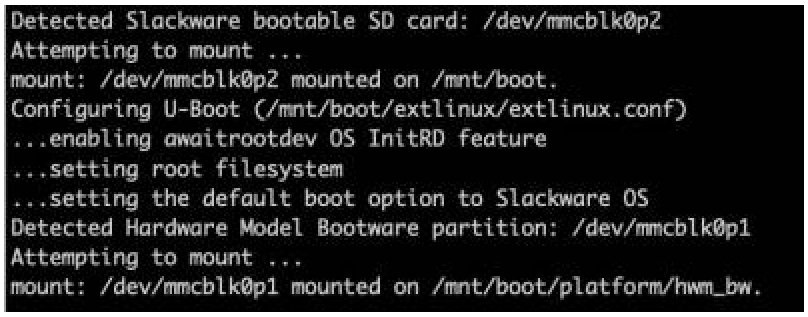
```
SELECT FILESYSTEM FOR /dev/sda2
Please select the type of filesystem to use for the specified device. Here
are descriptions of the available filesystems: Ext2 is the traditional
Linux file system and is fast and stable. Ext3 is the journaling version
of the Ext2 filesystem. Ext4 is the successor to the ext3 filesystem.
Btrfs is a B-tree copy-on-write filesystem. F2FS is a Flash-Friendly File
System. JFS is IBM's Journaled Filesystem, currently used in IBM
enterprise servers. ReiserFS is a journaling filesystem that stores all
files and filenames in a balanced tree structure. XFS is SGI's journaling
filesystem that originated on IRIX.

            ext2      Standard Linux Ext2 Filesystem
            ext3      Ext3 Journaling Filesystem
            ext4      Ext4 Journaling Filesystem
            jfs       IBM's Journaled Filesystem
            reiserfs  ReiserFS Journaling Filesystem
            btrfs     Btrfs Copy-on-Write B-tree Filesystem
            f2fs      Flash-Friendly File System
            xfs       SGI's Journaling Filesystem

             < OK >          <Cancel>
```



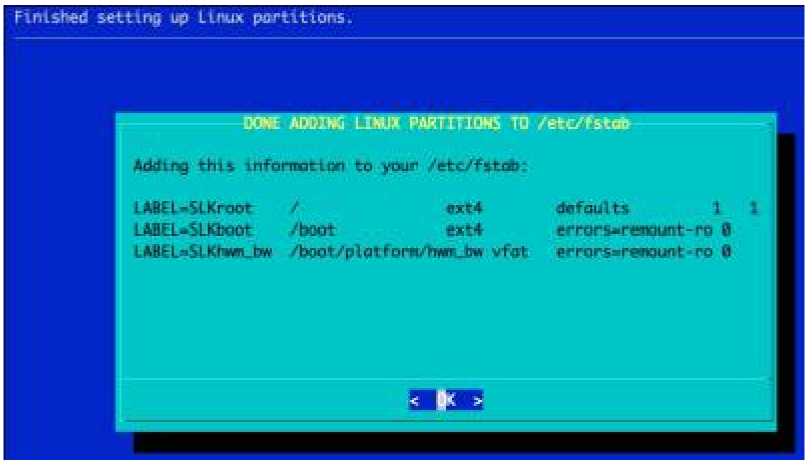
```
FORMATTING
Formatting /dev/sda2
Size:  220G
Filesystem type: ext4
```

Boot Loader Configuration

The Installer will configure the Boot Loader and the OS' /etc/fstab automatically:



```
Detected Slackware bootable SD card: /dev/mmcblk0p2
Attempting to mount ...
mount: /dev/mmcblk0p2 mounted on /mnt/boot.
Configuring U-Boot (/mnt/boot/extlinux/extlinux.conf)
...enabling awaitrootdev OS InitRD feature
...setting root filesystem
...setting the default boot option to Slackware OS
Detected Hardware Model Bootware partition: /dev/mmcblk0p1
Attempting to mount ...
mount: /dev/mmcblk0p1 mounted on /mnt/boot/platform/hwm_bw.
```



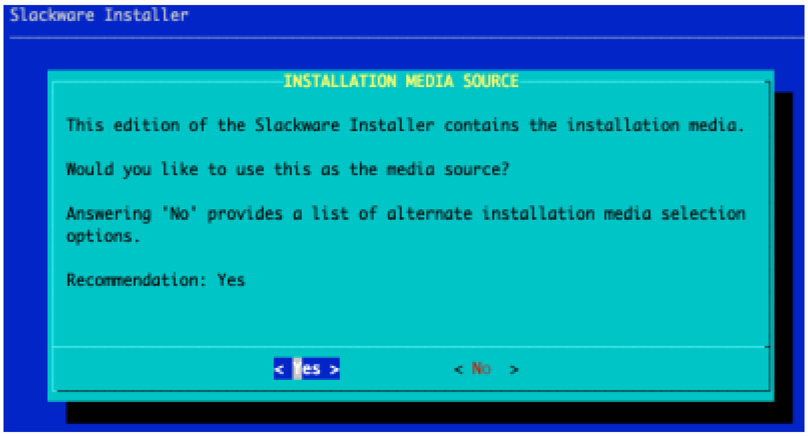
```
Finished setting up Linux partitions.

DONE ADDING LINUX PARTITIONS TO /etc/fstab

Adding this information to your /etc/fstab:

LABEL=SLKroot    /                       ext4      defaults          1   1
LABEL=SLKboot    /boot                   ext4      errors=remount-ro 0
LABEL=SLKhwm_bw  /boot/platform/hwm_bw vfat    errors=remount-ro 0

< OK >
```

On the ARM platform, the Swap and root file systems are addressed by labels (*see above:* '*LABEL=*') where as on x86/64 it's addressed by a direct reference to the block device (e.g. /dev/sda). The Slackware Installer will label swap partitions as '*SLKswap<x>*', and the root file system '*SLKroot*'. Other labels in use are '*SLKboot*' for the OS boot partition (/boot), and '*SLKhwm_bw*' for the Hardware Model Bootware partition (/boot/platform/hwm_bw). These labels are pre-labeled on the SD card image that you deployed and won't be presented during the installation process.

The rationale behind this divergence is that on x86 the root file system is typically on a storage bus (SCSI, SATA, ATA), where the physical configuration (which port the storage is connected to) of the storage rarely changes. This can be the case on ARM, but it's generally to a lesser extent and the root file system may be connected to a hot-plug bus such as USB. This lends itself to the risk of device re-ordering across boot cycles (e.g. /dev/sda becomes /dev/sdb), causing boot failure.

Please be aware that the Slackware Installer *only* labels the swap and root file system. Therefore you are advised to manually label the file systems and modify the OS /etc/fstab accordingly. If you have only a single storage device and don't plan on adding more, you can use the settings that the Slackware Installer configures.

Select Source Media



```
Slackware Installer

INSTALLATION MEDIA SOURCE

This edition of the Slackware Installer contains the installation media.

Would you like to use this as the media source?

Answering 'No' provides a list of alternate installation media selection
options.

Recommendation: Yes

            < Yes >            < No  >
```

Press ENTER to say 'Yes'.

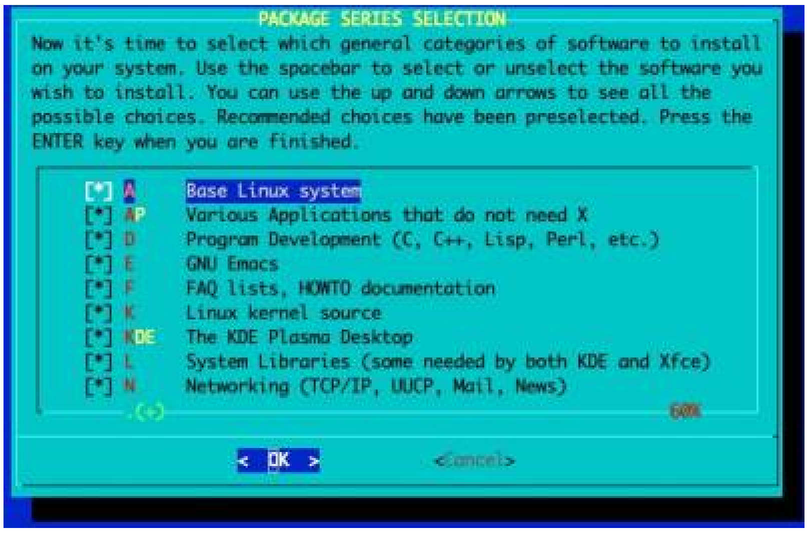
If you would like to install from an alternate media source, pick 'No' and you will be presented with options to install over NFS, USB and HTTP amongst others.

Ordinarily you should always say 'Yes' unless you've been directed to do otherwise.

Package Series Selection

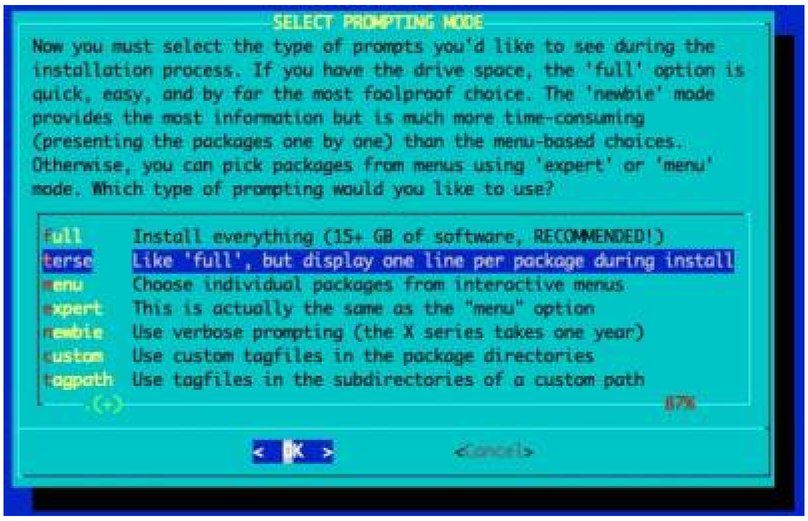
You can now choose the package sets to install. The recommendation is to install everything. A full Slackware installation will occupy approximately 15GB.

If you do not plan to use the graphical window manager such as KDE, you should de-select it.



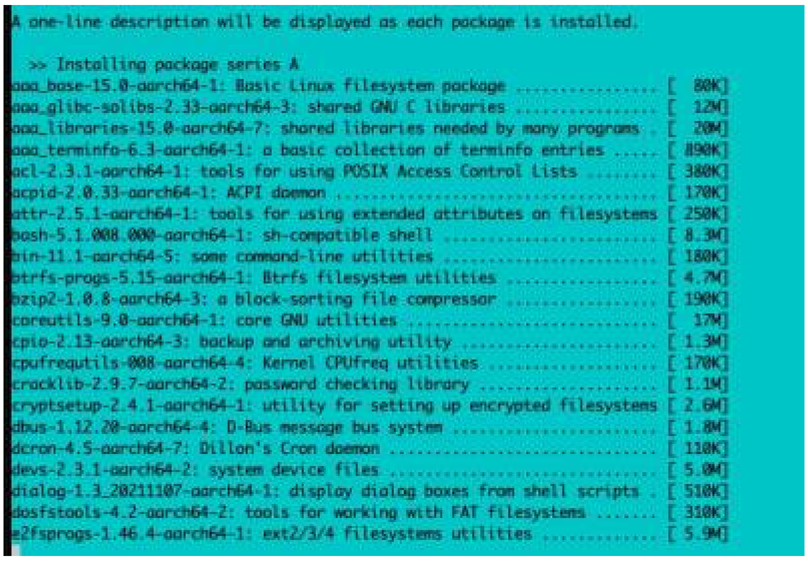
```
PACKAGE SERIES SELECTION
Now it's time to select which general categories of software to install
on your system. Use the spacebar to select or unselect the software you
wish to install. You can use the up and down arrows to see all the
possible choices. Recommended choices have been preselected. Press the
ENTER key when you are finished.

      [*] A      Base Linux system
      [*] AP     Various Applications that do not need X
      [*] D      Program Development (C, C++, Lisp, Perl, etc.)
      [*] E      GNU Emacs
      [*] F      FAQ lists, HOWTO documentation
      [*] K      Linux kernel source
      [*] KDE    The KDE Plasma Desktop
      [*] L      System Libraries (some needed by both KDE and Xfce)
      [*] N      Networking (TCP/IP, UUCP, Mail, News)
         .(+)                                              60%

              <  OK  >              <Cancel>
```

Pick the 'terse' option:



```
SELECT PROMPTING MODE
Now you must select the type of prompts you'd like to see during the
installation process. If you have the drive space, the 'full' option is
quick, easy, and by far the most foolproof choice. The 'newbie' mode
provides the most information but is much more time-consuming
(presenting the packages one by one) than the menu-based choices.
Otherwise, you can pick packages from menus using 'expert' or 'menu'
mode. Which type of prompting would you like to use?

 full     Install everything (15+ GB of software, RECOMMENDED!)
 terse    Like 'full', but display one line per package during install
 menu     Choose individual packages from interactive menus
 expert   This is actually the same as the "menu" option
 newbie   Use verbose prompting (the X series takes one year)
 custom   Use custom tagfiles in the package directories
 tagpath  Use tagfiles in the subdirectories of a custom path
     .(+)                                              87%

              <  OK  >              <Cancel>
```

The packages will begin installing:

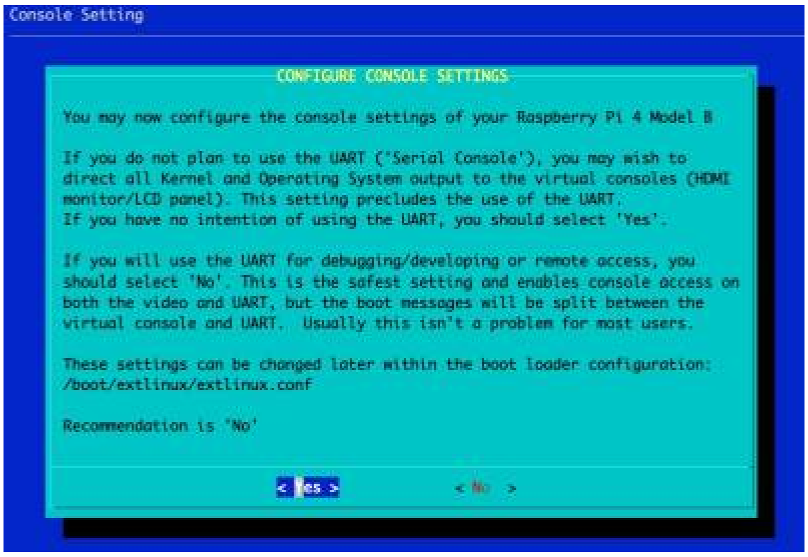


```
A one-line description will be displayed as each package is installed.

  >> Installing package series A
aaa_base-15.0-aarch64-1: Basic Linux filesystem package ................. [  80K]
aaa_glibc-solibs-2.33-aarch64-3: shared GNU C libraries ................. [  12M]
aaa_libraries-15.0-aarch64-7: shared libraries needed by many programs . [  20M]
aaa_terminfo-6.3-aarch64-1: a basic collection of terminfo entries ..... [ 890K]
acl-2.3.1-aarch64-1: tools for using POSIX Access Control Lists ........ [ 380K]
acpid-2.0.33-aarch64-1: ACPI daemon .................................... [ 170K]
attr-2.5.1-aarch64-1: tools for using extended attributes on filesystems [ 250K]
bash-5.1.008.000-aarch64-1: sh-compatible shell ........................ [ 8.3M]
bin-11.1-aarch64-5: some command-line utilities ........................ [ 180K]
btrfs-progs-5.15-aarch64-1: Btrfs filesystem utilities ................. [ 4.7M]
bzip2-1.0.8-aarch64-3: a block-sorting file compressor ................. [ 190K]
coreutils-9.0-aarch64-1: core GNU utilities ............................ [  17M]
cpio-2.13-aarch64-3: backup and archiving utility ...................... [ 1.3M]
cpufrequtils-008-aarch64-4: Kernel CPUfreq utilities ................... [ 170K]
cracklib-2.9.7-aarch64-2: password checking library .................... [ 1.1M]
cryptsetup-2.4.1-aarch64-1: utility for setting up encrypted filesystems [ 2.6M]
dbus-1.12.20-aarch64-4: D-Bus message bus system ....................... [ 1.8M]
dcron-4.5-aarch64-7: Dillon's Cron daemon .............................. [ 110K]
devs-2.3.1-aarch64-2: system device files .............................. [ 5.0M]
dialog-1.3_20211107-aarch64-1: display dialog boxes from shell scripts . [ 510K]
dosfstools-4.2-aarch64-2: tools for working with FAT filesystems ....... [ 310K]
e2fsprogs-1.46.4-aarch64-1: ext2/3/4 filesystems utilities ............. [ 5.9M]
```

Configure the Console Settings

If you plan on using the UART/'Serial' console, you should select 'No' here. If you plan on exclusively using an HDMI monitor, you should pick 'Yes'.

This setting can be reset to the default by editing /boot/extlinux/extlinux.conf and removing the 'console=' setting once the OS has booted.

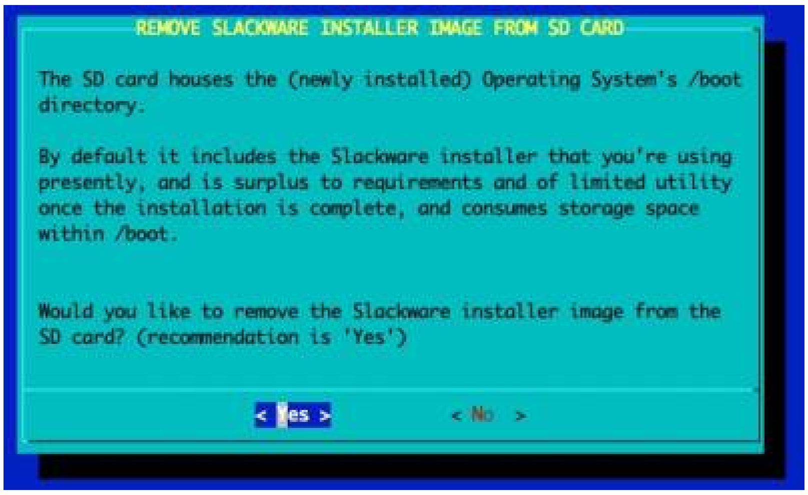


```
Console Setting

CONFIGURE CONSOLE SETTINGS

You may now configure the console settings of your Raspberry Pi 4 Model B

If you do not plan to use the UART ('Serial Console'), you may wish to
direct all Kernel and Operating System output to the virtual consoles (HDMI
monitor/LCD panel). This setting precludes the use of the UART.
If you have no intention of using the UART, you should select 'Yes'.

If you will use the UART for debugging/developing or remote access, you
should select 'No'. This is the safest setting and enables console access on
both the video and UART, but the boot messages will be split between the
virtual console and UART.  Usually this isn't a problem for most users.

These settings can be changed later within the boot loader configuration:
/boot/extlinux/extlinux.conf

Recommendation is 'No'

                    < Yes >          < No  >
```

Remove the Slackware Installer from the SD card

The Micro SD card is transformed from being the Slackware Installer into the Slackware OS's /boot partition. At this stage, if the installation has worked for you (at certain points in the Slackware installer you are past the point of no return) you can delete the Installer. However, if something has gone wrong you can reset the Raspberry Pi and reboot the installer without having to re-deploy the Slackware Installer image from your Linux Host Computer.

Generally you should say 'Yes' here.



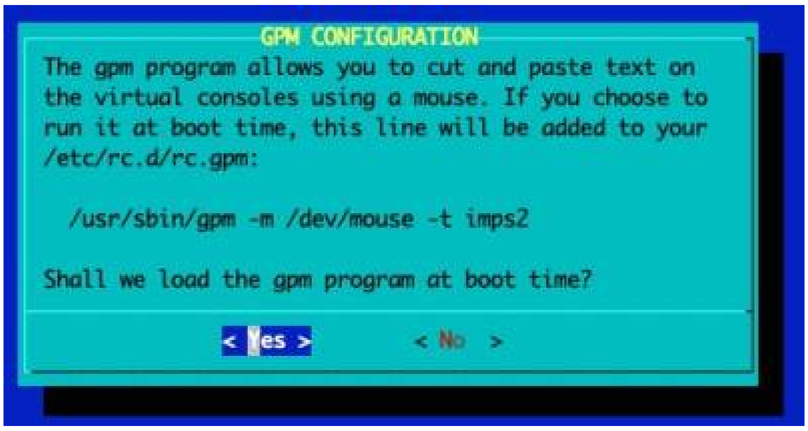
```
REMOVE SLACKWARE INSTALLER IMAGE FROM SD CARD

The SD card houses the (newly installed) Operating System's /boot
directory.

By default it includes the Slackware installer that you're using
presently, and is surplus to requirements and of limited utility
once the installation is complete, and consumes storage space
within /boot.


Would you like to remove the Slackware installer image from the
SD card? (recommendation is 'Yes')

                < Yes >          < No  >
```

You may be tempted to retain the Slackware Installer, but note that the Installer contains Linux Kernel modules for the Kernel that the Installer was originally shipped with. This means that as soon as you upgrade the Slackware Kernel package, the

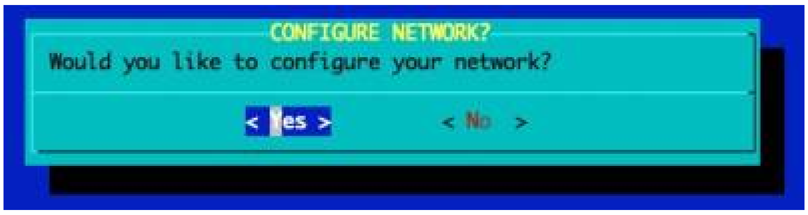
Installer will fail to boot. The option to retain the Installer is present purely because on a number of occasions, this author only realised that the installation was incorrectly performed upon completion, and needed to reinstall. Retaining the Installer avoids the requirement to re-deploy the image to the SD card.

Post Installation Configuration

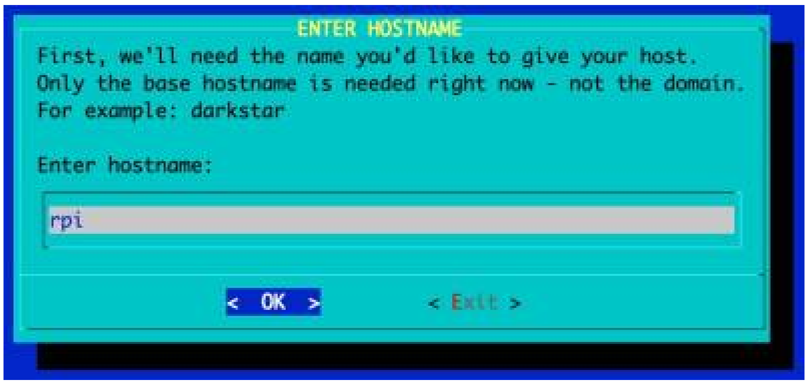
The Slackware Installer will walk you through the standard Slackware setup. The on-screen instructions will suffice.



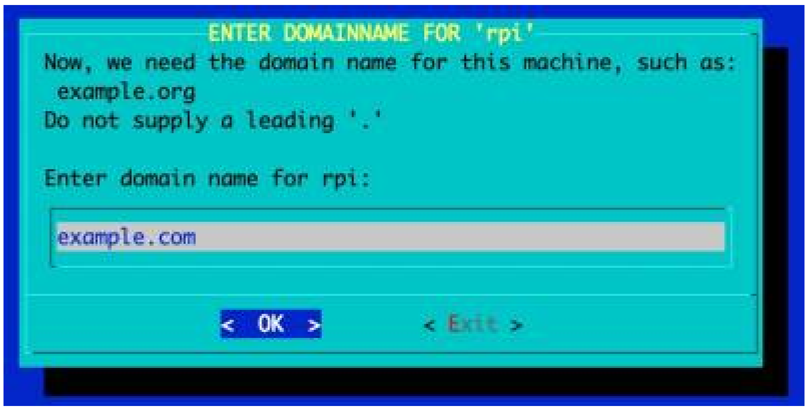
```
GPM CONFIGURATION
The gpm program allows you to cut and paste text on
the virtual consoles using a mouse. If you choose to
run it at boot time, this line will be added to your
/etc/rc.d/rc.gpm:

  /usr/sbin/gpm -m /dev/mouse -t imps2

Shall we load the gpm program at boot time?

        < Yes >        < No  >
```



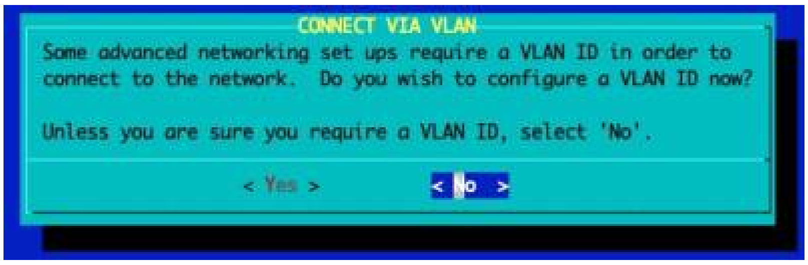
```
CONFIGURE NETWORK?
Would you like to configure your network?

        < Yes >        < No  >
```



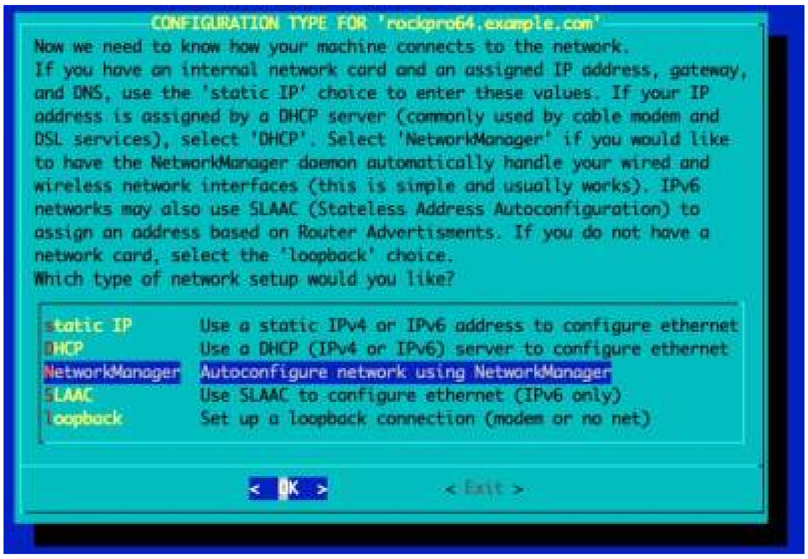
```
ENTER HOSTNAME
First, we'll need the name you'd like to give your host.
Only the base hostname is needed right now - not the domain.
For example: darkstar

Enter hostname:

rpi

        <  OK  >        < Exit >
```



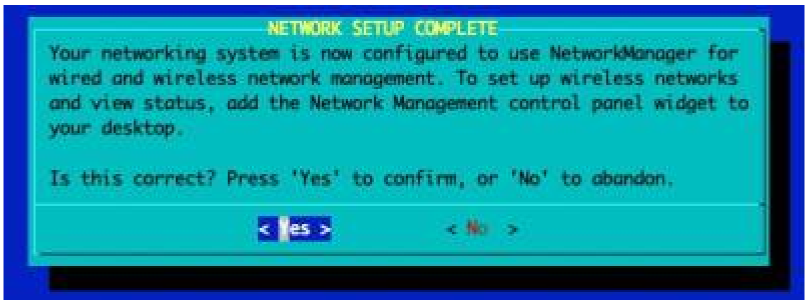
```
ENTER DOMAINNAME FOR 'rpi'
Now, we need the domain name for this machine, such as:
 example.org
Do not supply a leading '.'

Enter domain name for rpi:

example.com

        <  OK  >        < Exit >
```



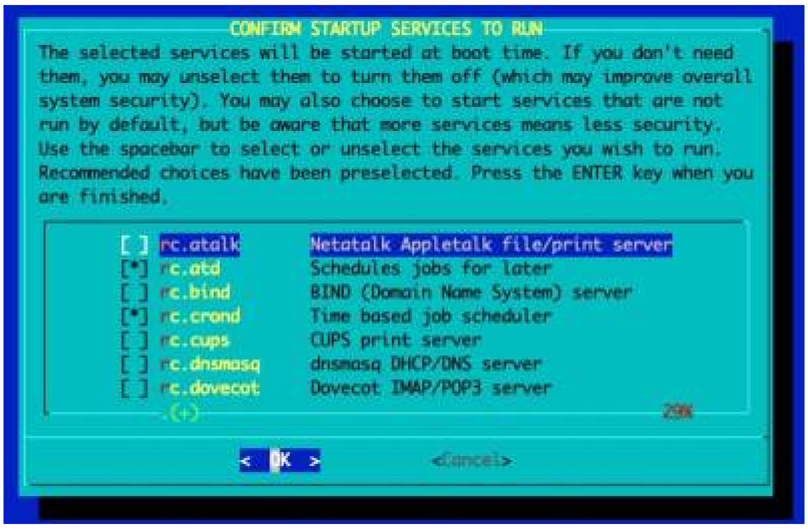
```
CONNECT VIA VLAN
Some advanced networking set ups require a VLAN ID in order to
connect to the network.  Do you wish to configure a VLAN ID now?

Unless you are sure you require a VLAN ID, select 'No'.

            < Yes >          < No  >
```



```
CONFIGURATION TYPE FOR 'rockpro64.example.com'
Now we need to know how your machine connects to the network.
If you have an internal network card and an assigned IP address, gateway,
and DNS, use the 'static IP' choice to enter these values. If your IP
address is assigned by a DHCP server (commonly used by cable modem and
DSL services), select 'DHCP'. Select 'NetworkManager' if you would like
to have the NetworkManager daemon automatically handle your wired and
wireless network interfaces (this is simple and usually works). IPv6
networks may also use SLAAC (Stateless Address Autoconfiguration) to
assign an address based on Router Advertisments. If you do not have a
network card, select the 'loopback' choice.
Which type of network setup would you like?

static IP        Use a static IPv4 or IPv6 address to configure ethernet
DHCP             Use a DHCP (IPv4 or IPv6) server to configure ethernet
NetworkManager   Autoconfigure network using NetworkManager
SLAAC            Use SLAAC to configure ethernet (IPv6 only)
loopback         Set up a loopback connection (modem or no net)

            < OK >          < Exit >
```



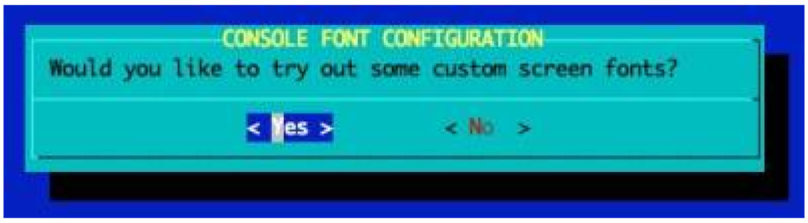
```
NETWORK SETUP COMPLETE
Your networking system is now configured to use NetworkManager for
wired and wireless network management. To set up wireless networks
and view status, add the Network Management control panel widget to
your desktop.

Is this correct? Press 'Yes' to confirm, or 'No' to abandon.

            < Yes >          < No  >
```



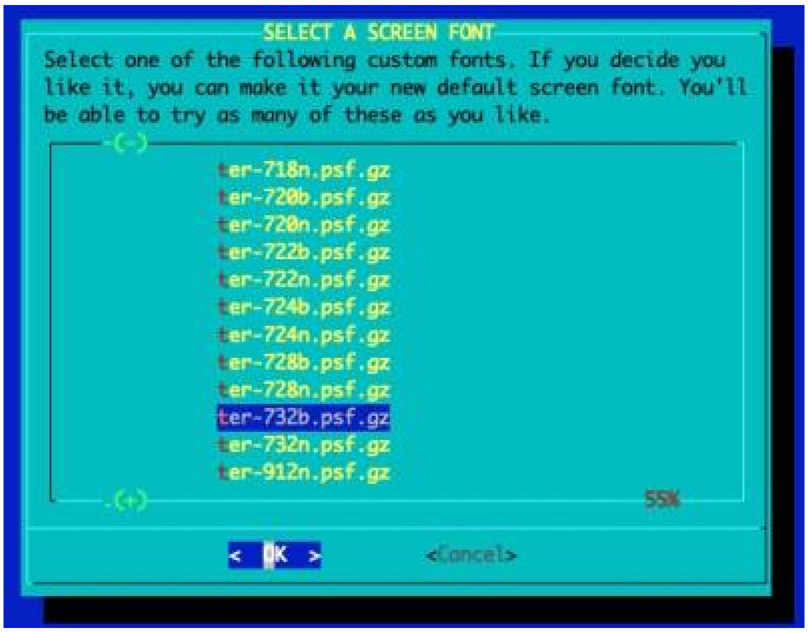
```
CONFIRM STARTUP SERVICES TO RUN
The selected services will be started at boot time. If you don't need
them, you may unselect them to turn them off (which may improve overall
system security). You may also choose to start services that are not
run by default, but be aware that more services means less security.
Use the spacebar to select or unselect the services you wish to run.
Recommended choices have been preselected. Press the ENTER key when you
are finished.

        [ ] rc.atalk        Netatalk Appletalk file/print server
        [*] rc.atd          Schedules jobs for later
        [ ] rc.bind         BIND (Domain Name System) server
        [*] rc.crond        Time based job scheduler
        [ ] rc.cups         CUPS print server
        [ ] rc.dnsmasq      dnsmasq DHCP/DNS server
        [ ] rc.dovecot      Dovecot IMAP/POP3 server
          (+)                                          29%

            < OK >          <Cancel>
```

Select a Console Font

It's recommended for the Raspberry Pi that a larger console font is configured for the virtual console.

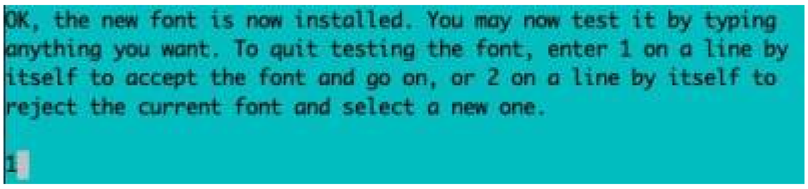


```
CONSOLE FONT CONFIGURATION
Would you like to try out some custom screen fonts?

        < Yes >          < No  >
```



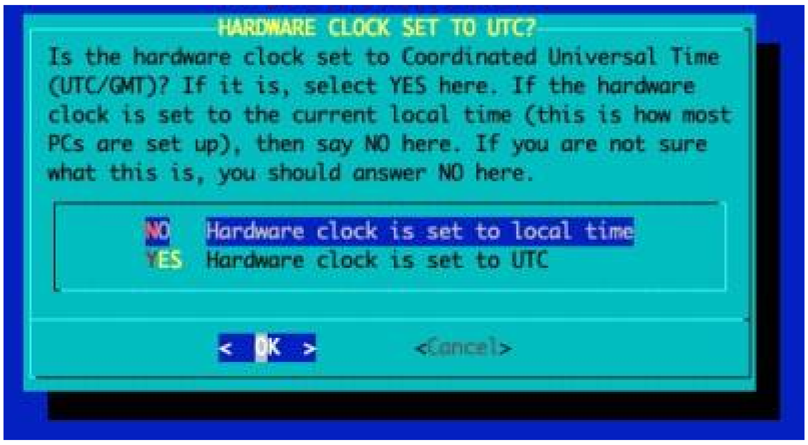
```
SELECT A SCREEN FONT
Select one of the following custom fonts. If you decide you
like it, you can make it your new default screen font. You'll
be able to try as many of these as you like.
   -(-)
          ter-718n.psf.gz
          ter-720b.psf.gz
          ter-720n.psf.gz
          ter-722b.psf.gz
          ter-722n.psf.gz
          ter-724b.psf.gz
          ter-724n.psf.gz
          ter-728b.psf.gz
          ter-728n.psf.gz
          ter-732b.psf.gz
          ter-732n.psf.gz
          ter-912n.psf.gz
   -(+)                                    55%

        <  OK  >         <Cancel>
```

The recommended font is 'ter-732b.psf'. This is the font used within the Installer.

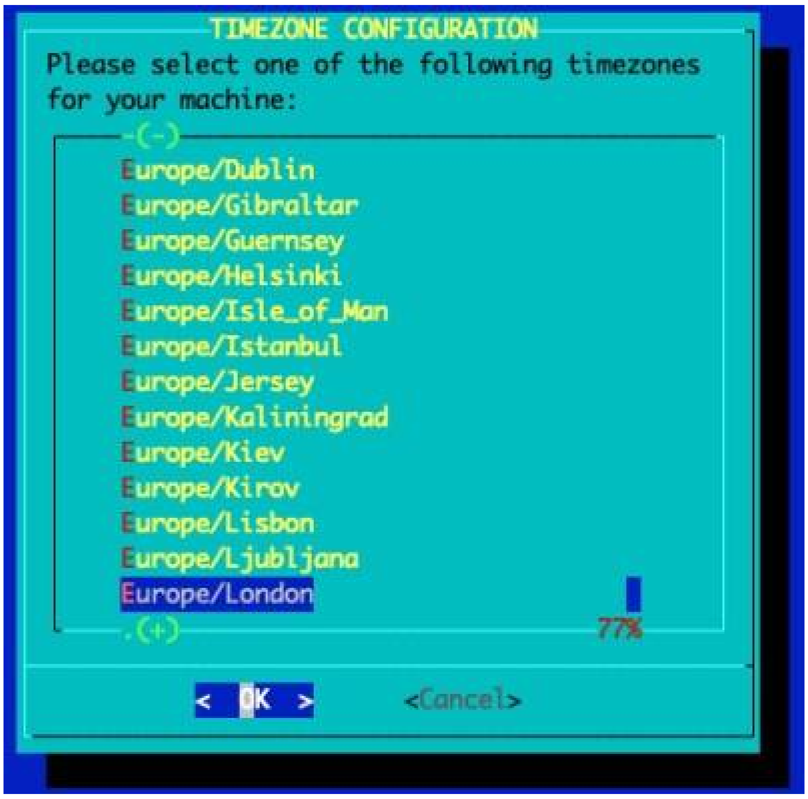


```
OK, the new font is now installed. You may now test it by typing
anything you want. To quit testing the font, enter 1 on a line by
itself to accept the font and go on, or 2 on a line by itself to
reject the current font and select a new one.

1
```

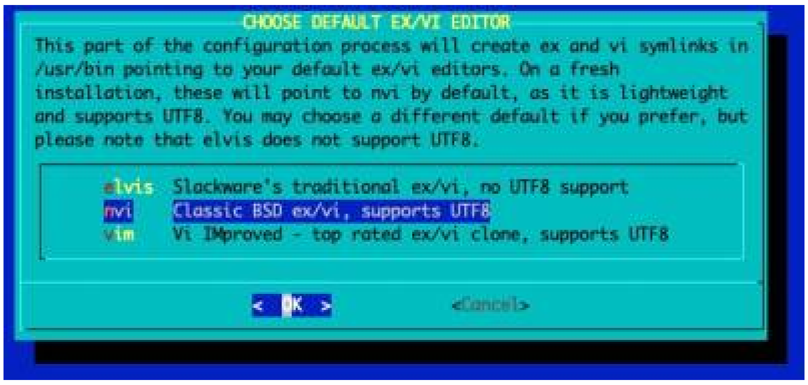
Continue Post Installation Configuration



```
HARDWARE CLOCK SET TO UTC?
Is the hardware clock set to Coordinated Universal Time
(UTC/GMT)? If it is, select YES here. If the hardware
clock is set to the current local time (this is how most
PCs are set up), then say NO here. If you are not sure
what this is, you should answer NO here.

      NO   Hardware clock is set to local time
      YES  Hardware clock is set to UTC

        <  OK  >         <Cancel>
```



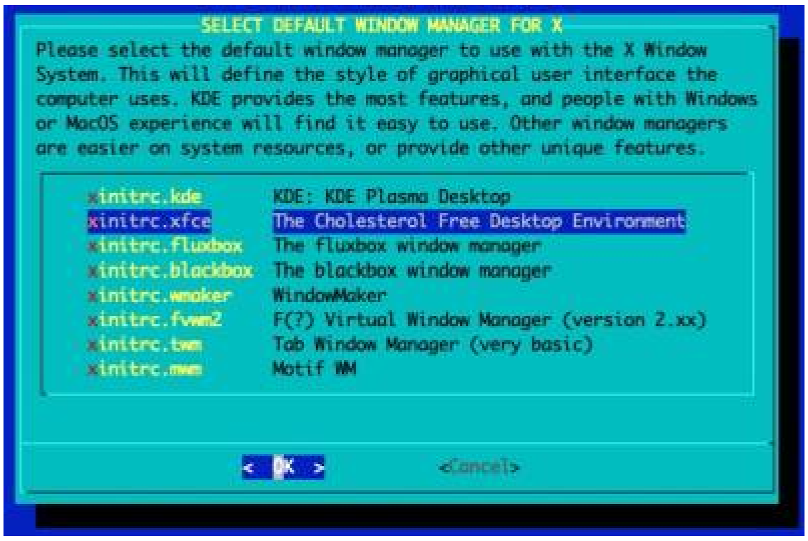
```
TIMEZONE CONFIGURATION
Please select one of the following timezones
for your machine:
-(-)
    Europe/Dublin
    Europe/Gibraltar
    Europe/Guernsey
    Europe/Helsinki
    Europe/Isle_of_Man
    Europe/Istanbul
    Europe/Jersey
    Europe/Kaliningrad
    Europe/Kiev
    Europe/Kirov
    Europe/Lisbon
    Europe/Ljubljana
    Europe/London
.(+)                                  77%

        <  OK  >        <Cancel>
```



```
CHOOSE DEFAULT EX/VI EDITOR
This part of the configuration process will create ex and vi symlinks in
/usr/bin pointing to your default ex/vi editors. On a fresh
installation, these will point to nvi by default, as it is lightweight
and supports UTF8. You may choose a different default if you prefer, but
please note that elvis does not support UTF8.

    elvis  Slackware's traditional ex/vi, no UTF8 support
    nvi    Classic BSD ex/vi, supports UTF8
    vim    Vi IMproved - top rated ex/vi clone, supports UTF8

        <  OK  >        <Cancel>
```

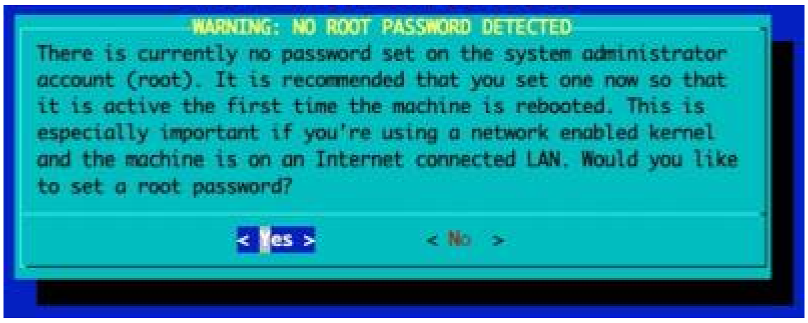
Configure GUI Window Manager

This author recommends using XFCE as it's light weight versus KDE.

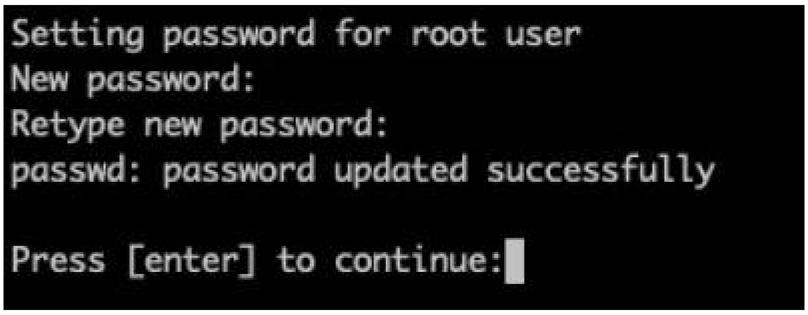


```
SELECT DEFAULT WINDOW MANAGER FOR X
Please select the default window manager to use with the X Window
System. This will define the style of graphical user interface the
computer uses. KDE provides the most features, and people with Windows
or MacOS experience will find it easy to use. Other window managers
are easier on system resources, or provide other unique features.

    xinitrc.kde        KDE: KDE Plasma Desktop
    xinitrc.xfce       The Cholesterol Free Desktop Environment
    xinitrc.fluxbox    The fluxbox window manager
    xinitrc.blackbox   The blackbox window manager
    xinitrc.wmaker     WindowMaker
    xinitrc.fvwm2      F(?) Virtual Window Manager (version 2.xx)
    xinitrc.twm        Tab Window Manager (very basic)
    xinitrc.mwm        Motif WM

        <  OK  >        <Cancel>
```

Continue Post Installation Configuration

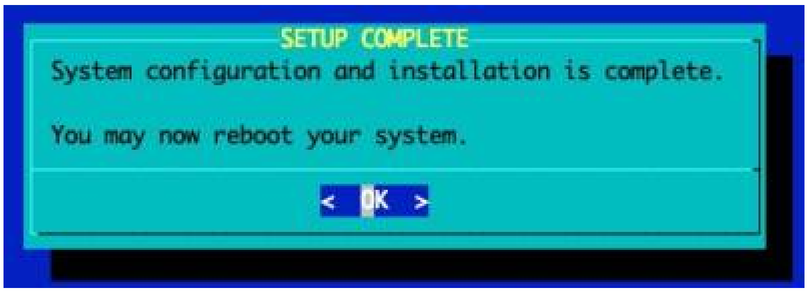


```
WARNING: NO ROOT PASSWORD DETECTED
There is currently no password set on the system administrator
account (root). It is recommended that you set one now so that
it is active the first time the machine is rebooted. This is
especially important if you're using a network enabled kernel
and the machine is on an Internet connected LAN. Would you like
to set a root password?

          < Yes >          < No  >
```

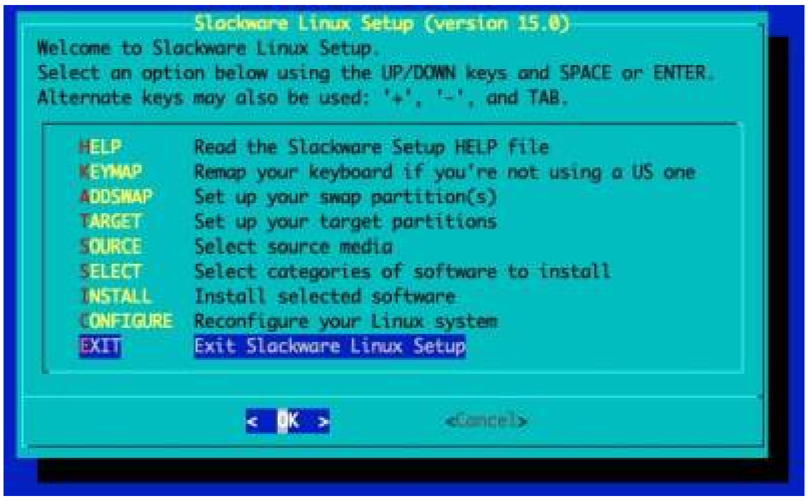


```
Setting password for root user
New password:
Retype new password:
passwd: password updated successfully

Press [enter] to continue:
```

Slackware Setup Complete

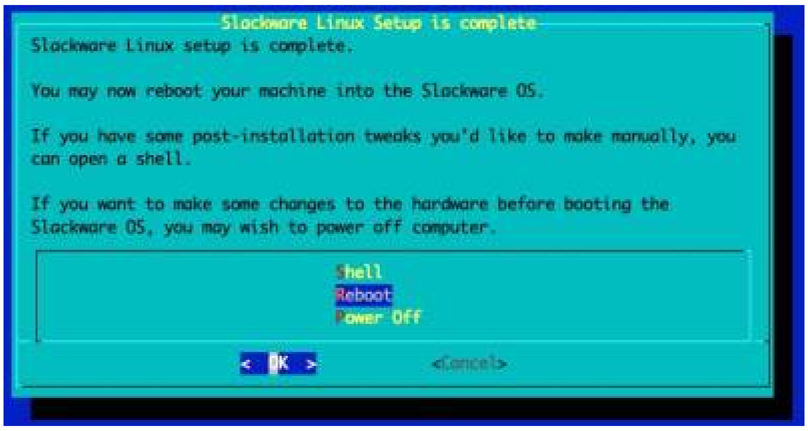


```
SETUP COMPLETE
System configuration and installation is complete.

You may now reboot your system.

              <  OK  >
```



```
Slackware Linux Setup (version 15.0)
Welcome to Slackware Linux Setup.
Select an option below using the UP/DOWN keys and SPACE or ENTER.
Alternate keys may also be used: '+', '-', and TAB.

   HELP       Read the Slackware Setup HELP file
   KEYMAP     Remap your keyboard if you're not using a US one
   ADDSWAP    Set up your swap partition(s)
   TARGET     Set up your target partitions
   SOURCE     Select source media
   SELECT     Select categories of software to install
   INSTALL    Install selected software
   CONFIGURE  Reconfigure your Linux system
   EXIT       Exit Slackware Linux Setup

          < OK >          <Cancel>
```

Reboot into the Slackware OS

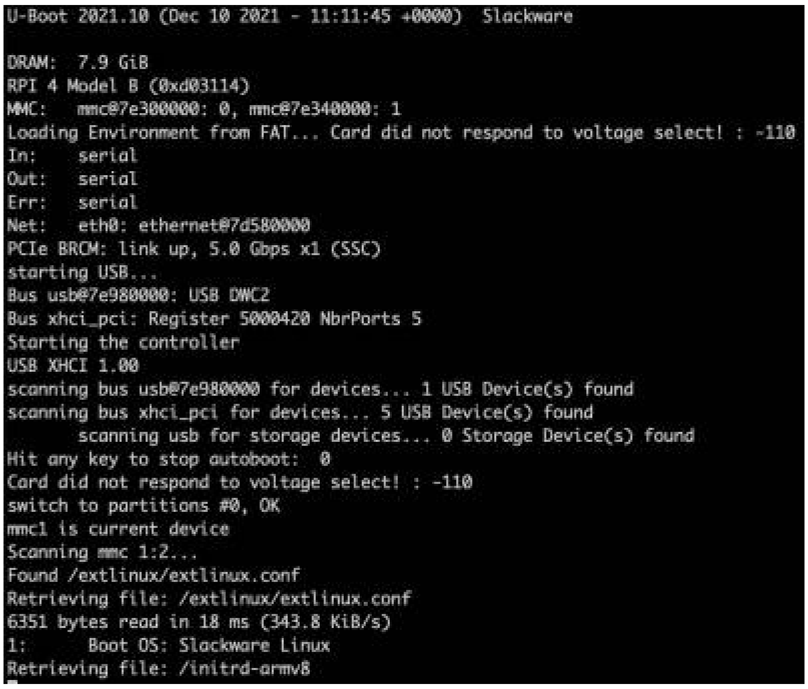

```
Slackware Linux Setup is complete
Slackware Linux setup is complete.

You may now reboot your machine into the Slackware OS.

If you have some post-installation tweaks you'd like to make manually, you
can open a shell.

If you want to make some changes to the hardware before booting the
Slackware OS, you may wish to power off computer.
```

Generally you'll want to reboot into the OS.

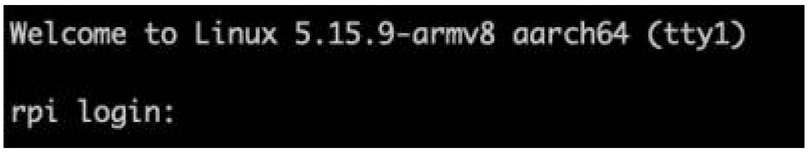
However, if you are planning on setting up RAID or need to customise the Operating System Initial RAM Disk, you should select 'Shell'.

The Slackware OS will be found within '/mnt'. You can use the 'os-initrd-mgr' tool (Video tutorial).

Booting the Slackware OS


```
U-Boot 2021.10 (Dec 10 2021 - 11:11:45 +0000)  Slackware

DRAM:  7.9 GiB
RPI 4 Model B (0xd03114)
MMC:   mmc@7e300000: 0, mmc@7e340000: 1
Loading Environment from FAT... Card did not respond to voltage select! : -110
In:    serial
Out:   serial
Err:   serial
Net:   eth0: ethernet@7d580000
PCIe BRCM: link up, 5.0 Gbps x1 (SSC)
starting USB...
Bus usb@7e980000: USB DWC2
Bus xhci_pci: Register 5000420 NbrPorts 5
Starting the controller
USB XHCI 1.00
scanning bus usb@7e980000 for devices... 1 USB Device(s) found
scanning bus xhci_pci for devices... 5 USB Device(s) found
       scanning usb for storage devices... 0 Storage Device(s) found
Hit any key to stop autoboot:  0
Card did not respond to voltage select! : -110
switch to partitions #0, OK
mmc1 is current device
Scanning mmc 1:2...
Found /extlinux/extlinux.conf
Retrieving file: /extlinux/extlinux.conf
6351 bytes read in 18 ms (343.8 KiB/s)
1:      Boot OS: Slackware Linux
Retrieving file: /initrd-armv8
```

Login to the Slackware OS


```
Welcome to Linux 5.15.9-armv8 aarch64 (tty1)

rpi login:
```

You may now login as 'root', using the password you set within the installer.

Post Installation Configuration

There are a few post-installation configuration tasks to complete.

Initial Time/Date Sync

If Internet access is available to the Raspberry Pi, prior to proceeding with any further setup, you may wish to set the time now as a one-off event:

Elevate yourself to ***root*** and use ***ntpdate***:

```
$ su -
# ntpdate rolex.ripe.net  # or pick your favourite NTP server
# hwclock -w # writes the date to the RTC
# logout
```

NTP (Network Time Protocol) setup

Even if your Raspberry Pi has an RTC (as documented in this guide), you may wish to configure it to set time from an Internet NTP Server. The Raspberry Pi requires continuous Internet access for this to function.

Add a plebeian user

You should add a plebeian (non-root) user using the 'adduser' tool.

This is documented here.

KDE fixups

If you are using KDE, you need to first adjust a setting.

As your plebian user, **prior** to loading KDE, run this command.

This will disable the *Compositing* feature. This is required when using the mainline Linux Kernel (as Slackware does).

```
kwriteconfig5 --file kwinrc --group Compositing --key Enabled false
```

You may now start KDE.

Use a graphical login manager

If you prefer to use a graphical login manager, you can configure the default runlevel as 4:

```
su -
sed -i 's?id:3:?id:4:?g' /etc/inittab
reboot
```

Managing Slackware on the Raspberry Pi

Keeping the Slackware OS up to date

One of the preferred tools to keep your system up to date is slackpkg.

Upgrading the Kernel: In Slackware ARM, you simply upgrade the Kernel packages and reboot - no manual steps are required

Loading Additional Linux Kernel Modules within the OS Proper

Often Kernel modules for discovered hardware will be automatically loaded, but occasionally you will need to manually configure the loading of some modules.

```
/etc/rc.d/rc.modules.local
```

This file is a shell script that is run as one of the last steps before the OS has fully booted. You can enter `modprobe` commands here to load the specific modules you require.

Configuration files within the directory `/lib/modprobe.d/` can be used to configure the parameters of the modules. Existing files within that directory serve as reference examples should you need them.

Loading Additional Linux Kernel Modules early in the boot sequence

There are a number of peripherals that may require Kernel modules loading early on in the boot sequence. An example of this would be RTCs (Real Time Clocks) or storage controllers that are required to access the file systems on which the OS lives.

Usually you won't need to load modules early in the boot sequence. See the previous section about loading modules from within the OS Proper.

To load Kernel modules during the early boot sequence, read:

```
/boot/local/README.txt
```

As root, the easiest way to begin is by renaming the example script:

```
mv /boot/local/load_kernel_modules.post.sample
/boot/local/load_kernel_modules.post
```

Then add the appropriate module loading commands to: `/boot/local/load_kernel_modules.post` You can also add shell code here to initialise a peripheral - writing something to the peripheral's Kernel interface, for example.

Slackware repository partition

The Slackware Installer image contains a type `ext4` partition labeled `SLKins_aio-pkgs` from which the packages are installed.

```
root@slackware:~# mount LABEL=SLKins_aio-pkgs /mnt/zip
root@slackware:~# cat /mnt/zip/README.txt
This file system contains the Slackware repository that is used during the
installation of Slackware.

Once you've booted into your OS you can delete or change this partition if
you
wish, or perhaps you might like to retain it for future reference.

root@slackware:~#
```

Most users simply leave the partition alone, as it causes no issues.

Customising the Slackware Linux Kernel

If you'd like to customise the Linux Kernel, the easiest way is to follow the HOWTO guide and use the Slackware ARM Kernel build script to create new packages.

This document also covers using the Raspberry Pi Kernel fork (although this is not recommended).

Reducing Boot Time

Slackware ARM ships with a generic OS InitRD (Operating System Initial RAM Disk - the environment that prepares the machine to boot the Operating System Proper), so as to support a wide range of Hardware Models.

However, this isn't the optimal setup once the Slackware OS has been installed because the generic OS InitRD typically exceeds 250MB, which in some cases can add several seconds to the boot time whilst it's loaded from the SD card.

The `os-initrd-mgr` (Operating System Initial RAM Disk Manager) tool has an option to synchronize the OS InitRD's Kernel modules with *only* those presently loaded within the Operating System.

To do this:

```
$ su -c 'os-initrd-mgr --sync-loaded-kmods' -  # note the final -
```

This option isn't the default, but you can make it so by following the instructions within `/etc/os-initrd-mgr.conf.sample`

This way when you upgrade the Kernel packages in the order described above, it'll automatically synchronize the modules.

os-initrd-mgr has a safety check to only proceed when the running Kernel and incoming Kernel are at the same major version and patch level.

For example, when running Linux 5.17.1, upgrading to 5.17.2 will work; but an upgrade of Linux 5.17.1 → 5.18.1 will require a reboot then to run os-initrd-mgr again to re-sync.

If at any point you want to revert to the generic OS InitRD, simply reinstall the `a/kernel` package (and unset the setting if you configured it in `/etc/os-initrd-mgr.conf`).

Managing the Raspberry Pi Firmware

Do not use the `rpi-update` script (found in other distributions) - this is unnecessary and is incompatible with Slackware.

The Raspberry Pi firmware is managed by the Slackware packages.

Bootware

The Raspberry Pi boots from a FAT partition on the Micro SD card. This partition contains the closed-source proprietary boot loader, firmware and various other assets it requires. It is mounted within the Slackware OS under `/boot/platform/hwm_bw` (on other distributions these files reside within /boot).

These assets are provided and managed by the Slackware package `a/hwm-bw-raspberrypi`.

Simply upgrading to the latest available Slackware packages will update this firmware.

Firmware for the EEPROM

The Slackware package `a/hwm-bw-raspberrypi` contains the available firmware that can be programmed to the Raspberry Pi's EEPROM.

You also need to have the `a/rpi-userland` package installed.

This firmware update requires manual action - see below

To update the firmware on your RPi's EEPROM; as root:

```
$ rpi-eeprom-update -d
```

This will report if there's newer firmware available.

To update the firmware:

```
$ rpi-eeprom-update -d -a
```

If the firmware was successfully updated, reboot:

```
$ reboot
```

The EEPROM tool configuration file is: `/etc/rpi-eeprom-update`. By default we're using firmware from the “critical” (most stable and well-tested) release channel, but you can change it within that configuration file.

Raspberry Pi SoC/IP Core Firmware

The firmware for the peripherals on the Raspberry Pi Hardware Models is contained within the Slackware `a/kernel-firmware` package.

Simply upgrading to the latest available Slackware packages will update this firmware.

Using Device Tree Overlays

Device Tree Overlays can be configured within the Raspberry Pi's Native Boot Loader configuration file: `/boot/platform/hwm_bw/config.txt`

Note: The Raspberry Pi's Native Boot Loader configuration is outside of the management of the Slackware OS so you need not be concerned with any automatic modifications.

Switching to the Raspberry Pi Kernel fork

The Raspberry Pi's are only intended to run the vendor's own Linux distribution, or to run as an 'Appliance' using the Raspberry Pi Linux Kernel fork.

As a consequence, the upstream mainline Linux Kernel in which support for the Raspberry Pi is fragile and prone to breaking without notice. Slackware uses official upstream repositories/releases and has a no-patch policy. As such, the recommendation of using the vendor's own Kernel fork is far outside the scope of Slackware.

Over time the fixes and new features for the Raspberry Pi will continue to appear in the main stream Kernel. However, if you are facing instability or need some specific feature that isn't yet available in the mainline official Linux Kernel, Slackware provides replacement Kernel packages that are built from the Raspberry Pi's Linux Kernel fork.

Switching to the Raspberry Pi Kernel fork packages is **not** a recommendation, just an option for users who face instability.

Installing extra Software

Slackware comes with a good base of software applications, but there are plenty more available in the Open Source Ecosystem.

The best way to add new software is to use the build scripts from SlackBuilds.org.

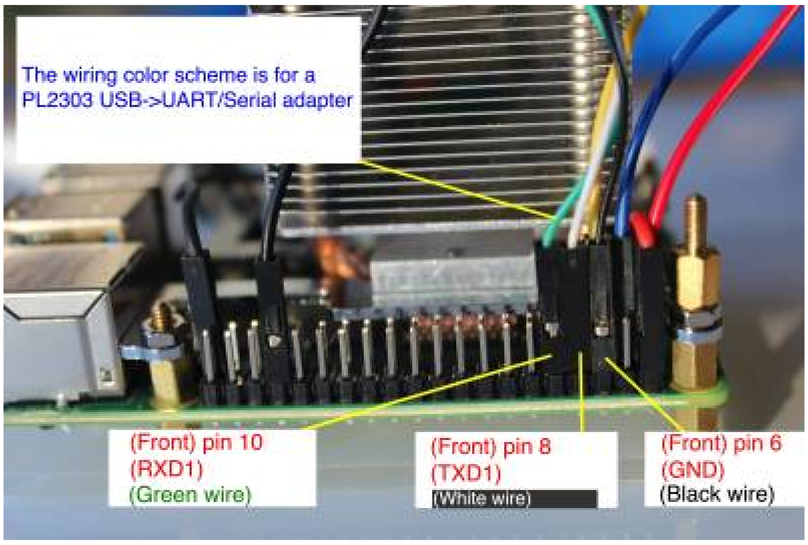
Using the Serial/UART adapter

This documentation discusses using the Raspberry Pi without the UART/Serial console.

If you'd like to use one, there are two that have been tested.

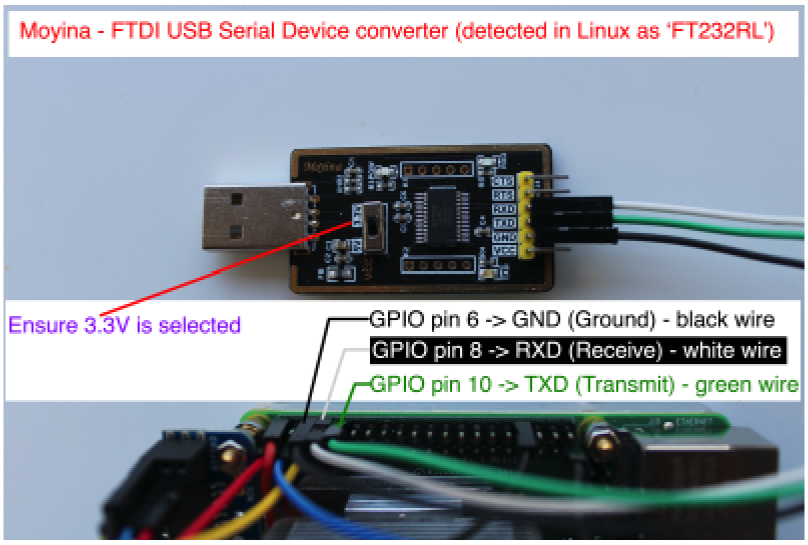
USB Serial Device converter: Prolific Technology Inc / PL2303

This image below shows the PL2302 (the Serial adapter listed in the Hardware table at the head of this document) connected to the Raspberry Pi:



USB Serial Device converter: FTDI / FT232RL

This is the model shown here.


Moyina - FTDI USB Serial Device converter (detected in Linux as 'FT232RL')
Ensure 3.3V is selected
GPIO pin 6 -> GND (Ground) - black wire
GPIO pin 8 -> RXD (Receive) - white wire
GPIO pin 10 -> TXD (Transmit) - green wire

Using the USB Serial Device converter on the Linux Host Computer

Once wired up, connect the USB end of the adapter into your Linux Host Computer, and use the following command.

This assumes that there are no other similar adapters occupying `/dev/ttyUSB0`. If so, you will need to adjust the device name accordingly (e.g. perhaps /dev/ttyUSB1).

```
screen  -T screen-256color /dev/ttyUSB0 115200
```

Known Limitations / Bugs

Issue	Work around	Notes
General fragility/instability	Transition to the Raspberry Pi Kernel fork packages	

Contributing to the Slackware ARM project

There are a plethora of ARM devices on the market which requires initial R&D and continuous testing. If you'd like to help Slackware support more ARM boards, please check out the documentation explaining how to get involved.

References

RPI GPIO exposition

RPi4 GPIO pin layout (credit: Les Pounder

Adafruit RTC wiring documentation