



EBook Gratis

APRENDIZAJE Raspberry Pi

Free unaffiliated eBook created from
Stack Overflow contributors.

#raspberry-
pi

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con Raspberry Pi.....	2
Observaciones.....	2
Examples.....	2
Configuración de la tarjeta SD.....	2
Instalación de Raspberry Pi - Windows.....	2
Raspberry Pi Introducción - Hello World in C.....	5
Pretexto.....	5
Hello World - Mi primer programa.....	5
Iniciar sesión con IPv6.....	6
Configuración Proxy.....	7
Raspberry Pi v2 y v3 Arch Linux Installation --- desde Mac o Linux.....	8
Capítulo 2: Crear una estación de escaneo con scanbd (Raspbian).....	9
Examples.....	9
Introducción e instalación.....	9
Configurar sane para escanear desde la red.....	10
Configure scanbd para sondear el escáner local.....	10
Identificar el escáner local.....	10
Confirme que el escáner local se encuentra por scanbd.....	11
¿Qué pasa si "dispositivo encontrado" no aparece?.....	12
Configure xinetd para hacer que scanbm escuche en la red.....	12
Activar acciones desde los botones del escáner.....	12
Capítulo 3: Habilitar SSH en Raspbian.....	15
Introducción.....	15
Observaciones.....	15
Examples.....	15
Habilitar SSH usando una computadora con Windows.....	15
Usa Raspberry Pi como sistema sin cabeza.....	16
Capítulo 4: Sistemas operativos.....	18
Introducción.....	18

Examples.....	18
Raspbian.....	18
Sistemas operativos basados en Yocto.....	18
Windows 10 IoT Core.....	18
OSMC.....	18
LibreELEC.....	18
Kali.....	19
Creditos.....	20

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [raspberrypi](#)

It is an unofficial and free Raspberry Pi ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Raspberry Pi.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con Raspberry Pi

Observaciones

Esta sección proporciona una descripción general de qué es raspberry-pi {también se puede describir en Stack Exchange con el término abreviado de "RPI"}, y por qué un desarrollador puede querer usarlo.

Tiene su propio sitio dedicado de Stack Exchange aquí en: <http://raspberrypi.stackexchange.com/>

Esta sección proporciona una descripción general de por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de raspberry-pi y vincular a los temas relacionados. Dado que la Documentación para raspberry-pi es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Configuración de la tarjeta SD

Comience por instalar el sistema operativo en la tarjeta MicroSD, ya sea NOOBS o Raspbian, ambos proporcionados por la fundación Raspberry Pi, disponibles en [su sitio](#) . NOOBS, que significa New Out of Box Software, está diseñado para principiantes y es el más fácil de instalar en su tarjeta SD. Puede seguir sus [instrucciones oficiales](#) o ver más abajo.

Formatee la tarjeta SD con una sola partición de FAT32, usando la herramienta de administración de unidades del sistema o una herramienta de terceros, como [SD Formatter 4.0 de la Asociación SD](#) , o [GParted](#) . Descarga el [archivo ZIP de NOOBS](#) y descomprímelo. a continuación, copie los archivos a la partición que acaba de crear. Asegúrate de expulsarlo antes de desconectarlo.

Instalación de Raspberry Pi - Windows

Pretexto:

Estas son instrucciones detalladas sobre cómo configurar una Raspberry Pi con el sistema operativo **Raspbian** .

Estas instrucciones son algo específicas de Windows.

Algunos pasos de instalación también pueden aplicarse a otros sistemas operativos, pero tenga en cuenta el primero.

Contenido

- Requerimientos
- Elegir un sistema operativo
- Instalando el sistema operativo

- Configuración básica
-

Requerimientos

1. Raspberry Pi (versión 1, 2 o 3 - cualquier modelo)
 2. Fuente de alimentación (5V, adaptador usb 2000mAh - recomendado)
 3. Tarjeta MicroSD (Clase 10 recomendada con al menos 8 GB; consulte la lista de tarjetas compatibles aquí: [eLinux](#))
 4. Computadora con lector de tarjetas SD (o un lector de tarjetas SD USB)
 5. Monitor / TV con puerto HDMI
 6. Cable HDMI
 7. Teclado USB
 8. [Win32 Disk Imager](#)
-

Elegir un sistema operativo

- [NOOBS](#)
Una colección de diferentes sistemas operativos para la Raspberry Pi, contiene Raspbian. Además de Raspbian, NOOBS también incluye Pidora, LibreELEC, OSMC, RISK OS, Arch Linux, Windows 10 IOT Core y algunas otras opciones. Es importante tener en cuenta que no todas estas opciones están disponibles en todos los modelos de Raspberry Pi (por ejemplo, Windows IOT Core solo está disponible en Raspberry Pi 2 y más reciente). La instalación de cualquier otro sistema operativo que no sea Raspbian requerirá una conexión Ethernet en su Raspberry Pi.
 - [Raspbian](#)
El sistema operativo oficialmente soportado para la Raspberry Pi, es un puerto del sistema operativo conocido como Debian. Preinstalado con software educativo y con una gran comunidad: este es el sistema operativo recomendado para la Raspberry Pi
 - [Windows 10 IOT Core](#) Una versión simplificada de Windows específicamente diseñada para dispositivos IOT (Internet Of Things). Es importante tener en cuenta que esta no es una versión completa de Windows, y no tiene lo que el usuario podría esperar, como un menú de inicio y la capacidad de ejecutar la mayoría de las aplicaciones de Windows.
-

Instalación del sistema operativo

tarjeta SD

1. Conecte su tarjeta SD a su computadora
2. Descargue el sistema operativo que ha elegido en formato **.img**
(Puede estar comprimido en un archivo **.zip** , que debe extraerse después de la descarga)
3. Abrir Win32DiskImager
4. Seleccione su tarjeta SD
5. Seleccione su archivo **.img de** sistema operativo
6. Haga clic en `Write`

Frambuesa pi

1. Inserte la tarjeta SD en la Raspberry Pi
2. Conecte su teclado USB
3. Conecte su monitor / TV
4. Conecte un cable Ethernet (opcional: ¡pero recomendado!)
5. Enchufe el cable de alimentación USB en la Raspberry Pi

Sistema operativo

Si el sistema operativo se escribió correctamente en la tarjeta SD y el sistema operativo es un sistema operativo ARM válido, debe instalarse automáticamente en la Raspberry Pi, con poca o ninguna interacción.

(Sin embargo, esto depende en gran medida del tipo de sistema operativo que se haya elegido) Si elige NOOBS, aparecerá una interfaz y podrá elegir qué instalar.

Configuración básica

Línea de comando:

El comando `sudo raspi-config` le da acceso a la configuración de Raspberry Pi.

Opciones de configuración:

1 Expand Filesystem	Ensures that all of the SD card storage is available to the OS
2 Change User Password	Change password for the default user (pi)
3 Enable Boot to Desktop/Scratch	Choose whether to boot into a desktop environment, Scratch, or the command line
4 Internationalisation Options	Set up language and regional settings to match your location
5 Enable Camera	Enable this Pi to work with the Raspberry Pi camera
6 Add to Rastrack	Add this Pi to the online Raspberry Pi Map (Rastrack)
7 Overclock	Configure overclocking for your Pi
8 Advanced Options	Configure advanced settings
9 About `raspi-config`	Information about this configuration tool

- Después de una instalación nueva y limpia, se recomienda elegir la primera opción: `Expand Filesystem`
- Si es europeo y no está familiarizado con la distribución del teclado en inglés, puede ir a `Internationalisation Options` y cambiar el idioma del teclado y más.
- Se recomienda evitar el menú `Overclock` como principiante. Especialmente si tienes la Raspberry Pi 3. Se sabe que el RPI 3 se calienta mucho, incluso si no está overlockeado. Si está pensando en hacer overclocking en el RPI 3, obtenga disipadores de calor o un ventilador para su placa, para evitar que se sobrecaliente.
- La habilitación del acceso SSH se puede hacer en el menú de `Advanced Options`, las credenciales serán el nombre de usuario y la contraseña predeterminados.

GUI:

Al instalar Raspbian Jessie se inicia automáticamente en una interfaz gráfica de usuario, las opciones mencionadas se presentan en el menú Opciones en la esquina superior izquierda. Se recomienda a los principiantes que sigan usando la GUI, pero si compró una Raspberry Pi

para aprender la línea de comandos. La opción para iniciar en la línea de comandos se puede encontrar en el menú Configuración.

Nota

El cambio de configuración en la Raspberry Pi generalmente provoca un reinicio que ocurre cuando acepta los cambios realizados.

Información extra

Nombre de usuario predeterminado: pi

Contraseña predeterminada: frambuesa

El apagado de tu Raspberry Pi se hace tirando del cable de alimentación USB.

(La elección de apagado en la interfaz gráfica de usuario de Raspbian simplemente pondrá a la Raspberry Pi en un estado disminuido, aún utilizando el poder, pero menos)

El reinicio de la Raspberry Pi se realiza simplemente enchufando el cable después de desenchufarlo.

(A menos que la opción de reinicio se elija en la GUI o `sudo reboot` en la línea de comandos.)

Raspberry Pi Introducción - Hello World in C

Pretexto

Esta es una introducción al ejemplo del programa Hello World en la Raspberry Pi escrito en C.

El siguiente ejemplo hace uso de la interfaz de línea de comandos y se configura como una guía paso a paso.

Junto con la creación de un programa Hello World, se presentarán al lector los comandos simples de la línea de comandos de Linux.

Nota: esta introducción fue escrita para principiantes.

Hello World - Mi primer programa

Primer paso:

Haciendo un directorio que contendrá código fuente.

- Localice su directorio de inicio escribiendo el siguiente comando `cd`
 - Crea una nueva carpeta para tu código fuente `mkdir programs`
(Consejo: Linux distingue entre mayúsculas y minúsculas al administrar nombres de archivos y directorios)
 - Cambia a tu nuevo directorio de `cd programs`
-

Segundo paso:

Escribiendo tu primer programa.

Los sistemas Linux ofrecen una gran variedad de editores de texto, de forma nativa encontrará Vim o Nano.

Este ejemplo hará uso del editor de texto Nano.

- Crea tu archivo de código fuente `nano helloworld.c`
- Esto abrirá un editor; al salir y guardar se creará el archivo en su carpeta.

El siguiente código es el código fuente del programa Hello World:

```
/* My first program */  
  
#include<stdio.h>  
  
int main()  
{  
    printf("Hello World\n");  
}
```

- Después de escribir el código `ctrl + x` para salir del editor, presione `y` y luego `enter` para guardar los cambios.
- Escriba el comando `ls` para verificar si el archivo está presente en su directorio.

Tercer paso:

Compilando tu primer programa.

- Para compilar nuestro archivo de código fuente `helloworld.c` necesitamos usar un compilador, usaremos el **Colección de compiladores GNU** - también conocida como GCC.
- El siguiente comando compila el código fuente en un programa binario ejecutable
`gcc helloworld.c -o myfirstprogram.bin`

El archivo de código fuente se ofrece como un argumento para el compilador GCC y `-o` define otro argumento que expresa que nos gustaría que el compilador produzca algo.

En este caso, queremos que `.bin` un archivo `.bin` que nosotros mismos nombramos.

Hay varios otros argumentos que puede usar al compilar con GCC, un ejemplo sería `-Wall` que habilita todas las advertencias. Esto le proporciona información sobre cualquier error que GCC pueda encontrar.

Cuarto paso:

Ejecutando tu primer programa.

- La ejecución de un programa en la Raspberry Pi se realiza agregando `./` delante del nombre del programa que desea ejecutar.
- Podemos ejecutar nuestro programa escribiendo `./myfirstprogram.bin`

El comando debe ejecutar el programa y producir `Hello World` en la ventana de la consola.

Iniciar sesión con IPv6

Por lo general, luchamos en el proceso de inicio de sesión en la Raspberry Pi utilizando `ssh` u otras herramientas similares. Pero podemos hacer el proceso más agradable.

Una vez que su Raspberry Pi está conectada a una red, obtiene una dirección IPv4 y una dirección **IPv6**, que se basa en la dirección MAC de la NIC. Lo bueno de esto es que la dirección IPv6 no cambia incluso si la red tiene un servicio DHCP.

Dicho esto, vamos a descubrir nuestra dirección IPv6. Solo necesita iniciar sesión una vez con IPv4 o con un monitor (HDMI o VGA). Abre una terminal y escribe:

```
ifconfig
```

Verás algo como:

```
eth0      Link encap:Ethernet  HWaddr 00:1C:C0:AE:B5:E6
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::21c:c0ff:feae:b5e6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:41620 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40231 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21601203 (20.6 MiB)  TX bytes:6145876 (5.8 MiB)
          Interrupt:21 Base address:0xe000
```

Como puede ver, su IPv4 se vería como `192.168.0.1` y su IPv6 sería `fe80::21c:c0ff:feae:b5e6` (observe la línea que comienza con `inet6 addr`).

Entonces, con esta información puede iniciar sesión con el comando `ssh` con la siguiente sintaxis:

```
ssh -6 pi@fe80::21c:c0ff:feae:b5e6%eth0
```

Este enfoque parece más complicado, pero la dirección IPv6 se define como el `link local` y, como se basa en la dirección MAC, a menos que usted la cambie, esto siempre funcionará independientemente de su dirección IPv4.

Configuración Proxy

Si está detrás de un proxy y necesita conectarse a Internet, puede usar:

```
export http_proxy="http://username:password@host:port/"
```

Para configurar el proxy dentro de `apt-get` :

```
cd /etc/apt/apt.conf.d
```

Crea un archivo llamado `10proxy`:

```
sudo nano 10proxy
```

Sin autenticación agregar la siguiente línea:

```
Acquire::http::Proxy "http://yourproxyaddress:proxyport/";  
Acquire::https::Proxy "http://yourproxyaddress:proxyport/";
```

Con autenticación:

```
Acquire::http::Proxy "http://username:password@yourproxyaddress:proxyport/";
```

El / al final fue importante. Sin ella no funciona.

Raspberry Pi v2 y v3 Arch Linux Installation --- desde Mac o Linux

Una de las mejores distribuciones de Linux actualmente para Raspberry Pi (de ahora en adelante, "RPI") es [Arch Linux](#) . Esta web muestra la instalación para:

- [RPI2](#) . Arquitectura ARMv7 (32 bits).
- [RPI3](#) . Hay dos opciones:
 - Arquitectura ARMv7 (32 bits) o
 - Arquitectura AArch (64 bits).

Este tutorial le aconseja que *no* instale la opción AArch --- "Es posible que parte del hardware de la placa no funcione o que funcione mal".

Lea [Empezando con Raspberry Pi en línea](#): <https://riptutorial.com/es/raspberry-pi/topic/5701/empezando-con-raspberry-pi>

Capítulo 2: Crear una estación de escaneo con scanbd (Raspbian)

Examples

Introducción e instalación

El propósito de esta documentación es utilizar un escáner sin ninguna interfaz de usuario. Un uso común es cargar un archivo escaneado en PDF directamente en Google Drive o Dropbox simplemente presionando los botones del escáner.

Scanbd permite activar acciones desde los botones del escáner, usa sane.

Lo que complica un poco las cosas es que al sondear el escáner, scanbd bloquea el dispositivo. Como también necesitamos escanear desde herramientas locales (como scanimage), necesitamos configurar un proxy que interrumpa el scanbd de sondeo cuando sea necesario. Este es el trabajo de scanbm, que controla la red y le da prioridad sobre scanbd a cualquier acceso local al escáner.

Por lo tanto, necesitamos:

- Configure la configuración de sonido local para escanear solo desde la red (a través de scanbm)
- Configure scanbd para tener acceso y sondear el escáner local

Como *raíz* :

```
apt-get update
apt-get install libconfuse-dev libsane-dev libudev-dev libusb-dev xinetd
```

La última versión de scanbd se puede encontrar aquí:

<https://sourceforge.net/projects/scanbd/files/releases/>

```
# Download the latest version of **scanbd**.
wget https://sourceforge.net/projects/scanbd/files/releases/scanbd-1.4.4.tgz/download

# Uncompress
tar xvzf scanbd-1.4.4.tgz && cd 1.4.4

# Configure with the installation path in /etc/
./configure --disable-Werror

# install it
make
make install

# Add a dbus policy to authorize "saned" user to manage scanbd :
cp integration/scanbd_dbus.conf /etc/dbus-1/system.d/
```

```
# Add the init script to manage the daemon :
cp integration/scanbd.debian /etc/init.d/scanbd
```

Scanbd ahora está instalado pero aún no está configurado.

Configurar sane para escanear desde la red

La configuración local de sane está dentro de `/etc/saned.d`

/etc/sane.d/dll.conf

```
# /etc/sane.d/dll.conf - Configuration file for the SANE dynamic backend loader
#
# Backends can also be enabled by configuration snippets under
# /etc/sane.d/dll.d directory -- packages providing backends should drop
# a config file similar to dll.conf in this directory, named after the package.
#
# The next line enables the network backend; comment it out if you don't need
# to use a remote SANE scanner over the network - see sane-net(5) and saned(8)
net
```

`dll.conf` contiene solo el backend `net` .

/etc/sane.d/net.conf:

```
# This is the net backend config file.

## net backend options
# Timeout for the initial connection to saned. This will prevent the backend
# from blocking for several minutes trying to connect to an unresponsive
# saned host (network outage, host down, ...). Value in seconds.
connect_timeout = 3

## saned hosts
# Each line names a host to attach to.
# If you list "localhost" then your backends can be accessed either
# directly or through the net backend. Going through the net backend
# may be necessary to access devices that need special privileges.

localhost
```

En este punto, **scanbm** aún no está configurado, no se puede acceder a ningún escáner desde la red. Necesitamos configurar `scanbm` junto con `scanbd` para probar la configuración.

Configure scanbd para sondear el escáner local

Identificar el escáner local.

Al usar `lsusb`, identifique el **ID de producto** (1909 aquí):

```
pi:# lsusb
pi:# Bus 001 Device 005: ID 04a9:1909 Canon, Inc. CanoScan LiDE 110
```

Con ese **productid** , grep el archivo de configuración correcto (depende del modelo de su escáner, para mí es **genesys.conf**):

```
pi:# grep 1909 /etc/sane.d/*conf
pi:# /etc/sane.d/genesys.conf:usb 0x04a9 0x1909
```

Copie el archivo dentro de su carpeta de configuración de scanbd:

```
cp /etc/sane.d/genesys.conf /usr/local/etc/scanbd/
cp /etc/sane.d/dll.conf /usr/local/etc/scanbd/
```

Edite **/usr/local/etc/scanbd/dll.conf** y reemplace `net` con el escáner correcto:

```
# /etc/sane.d/dll.conf - Configuration file for the SANE dynamic backend loader
#
# Backends can also be enabled by configuration snippets under
# /etc/sane.d/dll.d directory -- packages providing backends should drop
# a config file similar to dll.conf in this directory, named after the package.
#
# The next line enables the network backend; comment it out if you don't need
# to use a remote SANE scanner over the network - see sane-net(5) and saned(8)

genesys
```

Confirme que el escáner local se encuentra por scanbd

Ahora estamos listos para probar:

```
scanbd -d7 -f
```

Evite todo lo que pueda para manipular la env. `SANE_CONFIG_DIR`. Puede romper cosas, el valor predeterminado ya está configurado en `/usr/local/etc/scanbd/scanbd.conf` y en el script de inicio.

La salida debe ser similar a:

```
scanbd: foreground
scanbd: reading config file /usr/local/etc/scanbd/scanbd.conf
scanbd: debug on: level: 7
scanbd: dropping privs to uid saned
scanbd: dropping privs to gid scanner
scanbd: group scanner has member:
scanbd: saned
scanbd: pi
scanbd: drop privileges to gid: 110
scanbd: Running as effective gid 110
scanbd: drop privileges to uid: 110
```

```
scanbd: Running as effective uid 110
scanbd: dbus_init
scanbd: dbus_match type='signal',interface='org.freedesktop.Hal.Manager'
scanbd: SANE_CONFIG_DIR=/usr/local/etc/scanbd/sane.d
scanbd: sane version 1.0
scanbd: Scanning for local-only devices
scanbd: found device: genesys:libusb:001:005 Canon LiDE 110 flatbed scanner
```

El `found device` más importante aquí se `found device` .

¿Qué pasa si "dispositivo encontrado" no aparece?

Debe volver a verificar su configuración, especialmente la parte de identificación de esta documentación y borrar cualquier `env. SANE_CONFIG_DIR` eventual mal configurada al escribir: `unset SANE_CONFIG_DIR`

Configure `xinetd` para hacer que `scanbm` escuche en la red

Cree `/etc/xinetd.d/sane-port` que contiene:

```
service sane-port
{
    port          = 6566
    socket_type   = stream
    wait         = no
    user         = saned
    group        = scanner
    server       = /usr/local/sbin/scanbm
    server_args  = scanbm -c /usr/local/etc/scanbd/scanbd.conf
    disable     = no
}
```

! Comprueba dos **veces** la ruta de `scanbm` con la `which scanbm`

Luego, reinicie `xinetd` e inicie `scanbd` :

```
service xinetd restart
/etc/init.d/scanbd start
```

`scanimage -L` ahora debería mostrar el escáner.

La mayoría de los problemas en esta etapa se refieren a los permisos. Asegúrese de que `scanimage -L` funcione cuando haya iniciado sesión `saned` haciendo `sudo -u saned -s .`

Si no, revise los permisos de USB: `ls -al /dev/bus/usb/***`

Activar acciones desde los botones del escáner.

Las acciones se encuentran en `/usr/local/etc/scanbd/scanbd.conf` . Tengo 4 botones que son

escanear , copiar , correo electrónico y archivo .

El archivo de configuración predeterminado no incluye todas las acciones por defecto, probablemente tendrá que agregar el bloque manualmente. Puede tener menos o más botones dependiendo del modelo de su escáner.

Para cada acción, estableceremos una ruta personalizada para la opción de **script** .

```
action scan {
    filter = "^scan.*"
    numerical-trigger {
        from-value = 1
        to-value    = 0
    }
    desc    = "Scan to file"
    # script must be an relative path starting from scriptdir (see above),
    # or an absolute pathname.
    # It must contain the path to the action script without arguments
    # Absolute path example: script = "/some/path/foo.script"
    script = "/home/pi/scan.sh"
}
```

No olvide comentar cualquier otra acción predeterminada al final de scanbd.conf:

```
# devices
# each device can have actions and functions, you can disable not relevant devices
#include(scanner.d/avision.conf)
#include(scanner.d/fujitsu.conf)
#include(scanner.d/hp.conf)
#include(scanner.d/pixma.conf)
#include(scanner.d/snapscan.conf)
#include(scanner.d/canon.conf)
```

Ahora puede crear su script personalizado para manejar cada acción:

Cada línea relativa a `/sys/class/leds/led0/trigger` es para controlar el LED para monitorear lo que está sucediendo. Puedes hacer lo que quieras, `cat /sys/class/leds/led0/trigger` te da todos los diferentes patrones de luces.

/home/pi/scan.sh

```
#!/bin/bash

# don't forget to create the folder
scan_dir=/home/pi/scanned-files
datetime=`date +%F_%H%M%S`

echo none >/sys/class/leds/led0/trigger

case $SCANBD_ACTION in
    scan)
        filename=file-$datetime
        logger -t "scanbd: $0" "$SCANBD_DEVICE $SCANBD_ACTION - scanning --resolution 150 --mode
Color --depth 8 --format=tiff to $scan_dir/$filename.jpg"
        echo timer >/sys/class/leds/led0/trigger
```



```
    scanimage -d $SCANBD_DEVICE --resolution 150 --mode Color --depth 8 --format=tiff --
brightness 5 --contrast 20 | convert tiff:- -compress jpeg $scan_dir/$filename.pdf
    echo none >/sys/class/leds/led0/trigger
    logger -t "scanbd: $0" "Finished scanning"
    ;;

email)
    logger -t "scanbd: $0" "Emailing $scan_dir/file-*.pdf"
    echo heartbeat >/sys/class/leds/led0/trigger
    # here are the lines to send the file
    echo none >/sys/class/leds/led0/trigger

esac
```

Lea [Crear una estación de escaneo con scanbd \(Raspbian\)](https://riptutorial.com/es/raspberry-pi/topic/6701/crear-una-estacion-de-escaneo-con-scanbd-raspbian-) en línea:

<https://riptutorial.com/es/raspberry-pi/topic/6701/crear-una-estacion-de-escaneo-con-scanbd-raspbian->

Capítulo 3: Habilitar SSH en Raspbian

Introducción

Las versiones anteriores de Raspbian (antes de noviembre de 2016) tenían SSH habilitado de forma predeterminada. También tenían un nombre de usuario predeterminado (pi) y una contraseña (frambuesa); esto se hizo para facilitar la instalación por primera vez para los nuevos usuarios. Pero esto obviamente representa un gran agujero de seguridad. La nueva versión (noviembre de 2016) deshabilita SSH de forma predeterminada y también mostrará una advertencia si SSH está habilitado sin cambiar la contraseña predeterminada.

Observaciones

El contenido del archivo ssh no importa, puede crear un archivo vacío como hicimos anteriormente.

Durante el arranque, el Pi buscará este archivo. Si existe se habilitará SSH y se eliminará el archivo.

Examples

Habilitar SSH usando una computadora con Windows

Para habilitar SSH, cree un archivo llamado ssh en el directorio / boot de su tarjeta SD.

1. Identifique la letra de unidad asociada con su tarjeta SD.
2. Abra un símbolo del sistema (presione **Win + R** en su teclado para abrir la ventana Ejecutar. Luego, escriba **cmd**)
3. Ingrese lo siguiente en el símbolo del sistema (reemplazando DriveLetter con la letra que identificó en el paso 1):

```
echo.> DriveLetter: / ssh
```

4. Presiona `enter`

Verifique que su tarjeta SD contenga un archivo llamado ssh.

1. Ingrese lo siguiente en el símbolo del sistema (nuevamente reemplace DriveLetter con la letra que identificó en el paso 1)

```
dir DriveLetter:
```

2. Presiona `enter`

Debería ver un archivo ssh en la lista del directorio.

Para evitar la advertencia de seguridad mencionada anteriormente (re: SSH habilitado con la contraseña predeterminada), cambie la contraseña predeterminada para el usuario Pi. Esto se puede hacer desde la línea de comandos con el `passwd`, el guión Raspi-config, o el menú de configuración en el escritorio de píxeles.

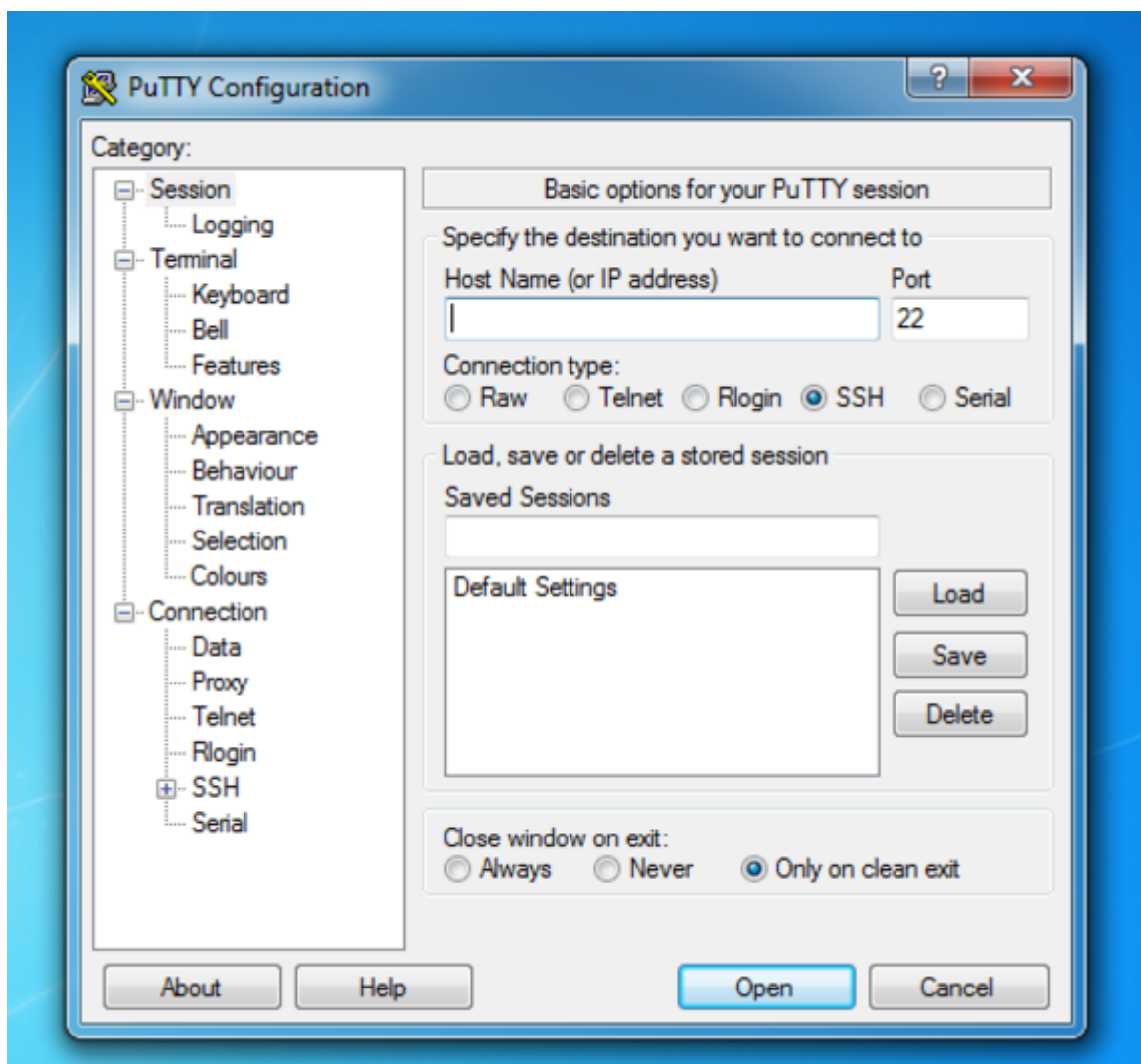
Usa Raspberry Pi como sistema sin cabeza

El ejemplo anterior nos ayuda a activar SSH en Pi. Este ejemplo es el requisito previo del ejemplo anterior.

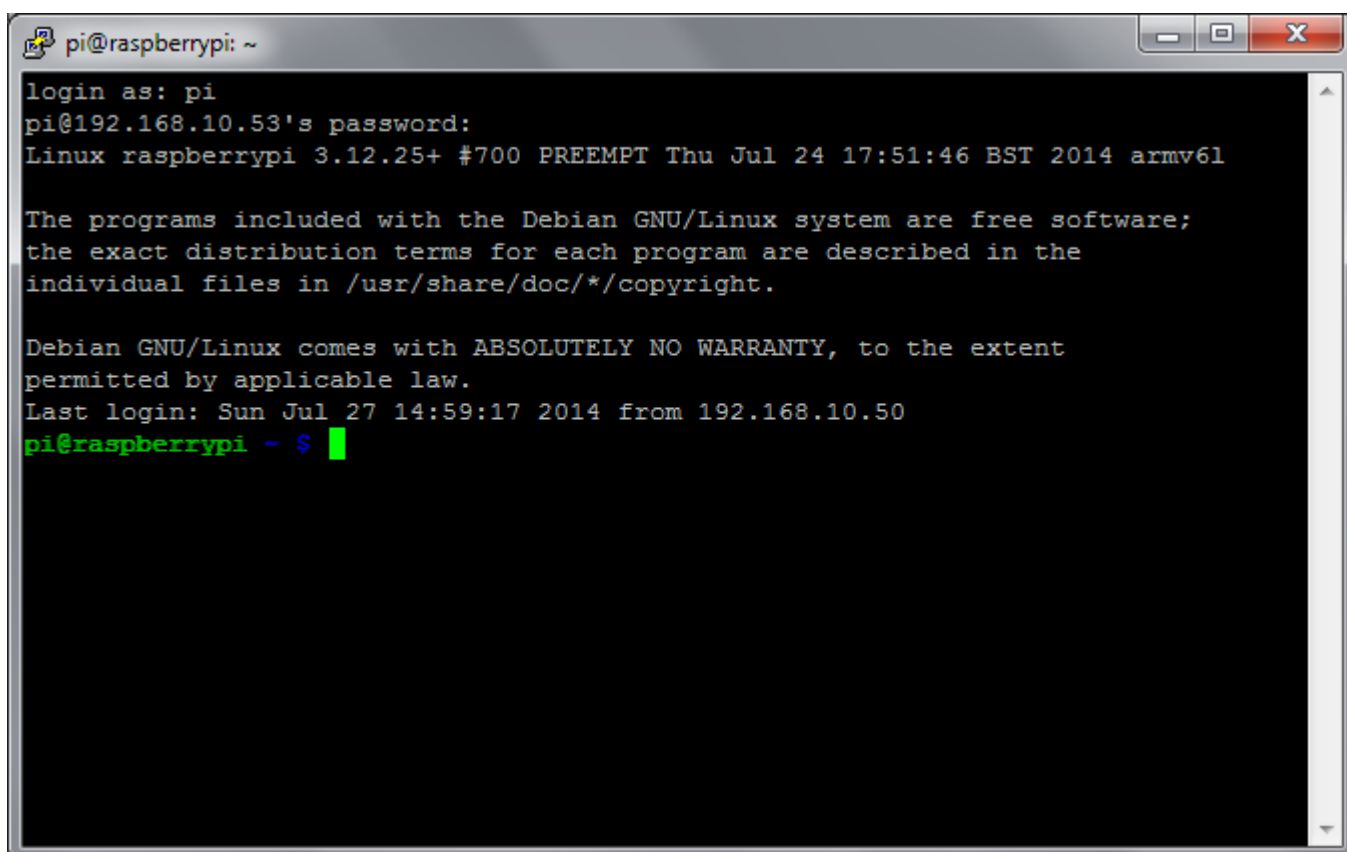
Para el sistema operativo Linux : abra el terminal y escriba esta `ssh pi@your local IP address` . A continuación, deberá proporcionar la contraseña.

Nota : aquí **pi** : su **nombre de usuario de raspberry pi** y su **dirección IP local** : la dirección IP de su dispositivo. Obtendrá esta dirección IP de su enrutador, lista de clientes DHCP. Sin embargo, este tema está fuera de alcance.

Para el sistema operativo Windows : podemos usar [Putty](#) para iniciar sesión a través de ssh. En Host Name escriba su dirección IP de Raspberry pi.



Después de presionar **Abrir** . Debe proporcionar su **nombre de usuario** y **contraseña** . Después del exitoso intento, obtendrás esta pantalla.



```
pi@raspberrypi: ~
login as: pi
pi@192.168.10.53's password:
Linux raspberrypi 3.12.25+ #700 PREEMPT Thu Jul 24 17:51:46 BST 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jul 27 14:59:17 2014 from 192.168.10.50
pi@raspberrypi ~ $ █
```

Lea **Habilitar SSH en Raspbian en línea**: <https://riptutorial.com/es/raspberry-pi/topic/8355/habilitar-ssh-en-raspbian>

Capítulo 4: Sistemas operativos

Introducción

Esta sección contiene información sobre varios sistemas operativos disponibles para Raspberry Pi.

Examples

Raspbian

El sistema operativo más popular para la Raspberry Pi es un Raspbian basado en Debian.

Es oficialmente apoyado por la Fundación Raspberry Pi.

Raspbian se puede descargar desde el sitio oficial de Raspberry Pi en una de dos variantes:

- Con entorno de escritorio
- Lite- Imagen mínima

A partir de septiembre de 2016, Raspbian se entrega con el entorno de escritorio PIXEL ("Pi Improved Xwindows Environment, Lightweight")

Sistemas operativos basados en Yocto.

Es posible crear una distribución propia de Linux usando Yocto Project.

Para Raspberry Pi, hay una capa específica [meta-raspberrypi](#) que debe usarse para crear una imagen.

Windows 10 IoT Core

Windows 10 IoT Core está disponible solo para Raspberry Pi 2 y 3. Es importante tener en cuenta que esta no es una versión completa de Windows como se podría esperar, está diseñada específicamente para las aplicaciones IoT (Internet of Things). No ejecutará la mayoría de las aplicaciones de Windows estándar y no tiene un menú de inicio. Se inicia en una página con algunos ejemplos de código y videos, y tiene una interfaz de línea de comandos.

OSMC

OSMC es un reproductor de medios gratuito y de código abierto basado en Linux.

LibreELEC

Distribución de Linux con características requeridas para ejecutar [Kodi](#) .

Kali

Un sistema operativo completamente diseñado para pruebas de penetración.

Enlace de descarga:

<https://www.offensive-security.com/kali-linux-arm-images/>

Requisitos:

1. Tarjeta de 8GB Clase 10 SD como mínimo
2. Tarjeta SD de 16GB clase 10 recomendada

Inicio de sesión predeterminado:

```
Username: root
Password: toor
```

Notas

1. Instalar gparted y expandir la partición a tamaño completo
2. Ejecute apt-get install kali-linux-full para obtener todas las herramientas estándar disponibles
3. Ejecuta entorno de escritorio XFCE

Lea Sistemas operativos en línea: <https://riptutorial.com/es/raspberry-pi/topic/8656/sistemas-operativos>

Creditos

S. No	Capítulos	Contributors
1	Empezando con Raspberry Pi	An Epic Person , andpei , atilacamura , Cicero , Community , ctafur , Patrick L , RvT , S.Spieker , SlySven
2	Crear una estación de escaneo con scanbd (Raspbian)	Pierre Maoui
3	Habilitar SSH en Raspbian	M Hornbacher , Saiful Azad , Steve Robillard
4	Sistemas operativos	lewiatan , M Hornbacher , Patrick L