LXTerminal Commands

## File Navigation

* ls – lists all files and directories in the current directory
* cd [dir] – change directory to the given [dir]
* cd .. – move up a directory to the previous directory
* cd – move to the home directory

## File Manipulation

* cp [file1] [file2]– copy from [file1] into [file2]
* mv [file1] [file2]– move [file1] into [file2], can also rename [file1]
* rm [file]– will remove the [file]
* mkdir [dir] – will create a new directory [dir] in the current directory
* nano [file] – open the file in a lightweight text editor
* chmod +x [file] – modify a [file] to make it executable
* ./[executablefile] – will execute an [executablefile]
* python [pythonfile.py] – will execute a [python file.py]

## System Commands

* startx – open a graphical desktop from a full-screen terminal
* echo [command] – will post the result of a given [command]
* echo “message” – will post a “message” to the terminal output
* sudo [command] – a prefix added to pass admin rights to the [command]
* apt-get install [software] – will install [software] on the system
* apt-get update – will update the system
* apt-get upgrade – will upgrade the system with the most recent update
* man [command]– will give you the manual page of a given [command]
* poweroff – will shut-down the system
* reboot – will reboot the system
* su - – will enter the pi into root mode
* su pi – will enter the pi back to pi mode
* raspi-config – opens the raspberry pi config window

## Networking Commands

* iwconfig – will post the wireless configuration
* ifconfig – will post information on all network infrastructure
* ping [ip] – will attempt to ping an [ip] address
* ifdown wlan0 – brings down the wireless interface wlan0
* ifup wlan0 – brings up the wireless interface wlan0
* service networking start – will start the networking service
* service networking restart – will restart the networking service
* service networking stop – will stop the networking service
* service networking status – will poll the status of the networking service
* ssh [devicename]@[ip] – will connect via secure shell to another device

## Nano Editor Shortcuts

* Ctrl+x – exit the editor, will prompt to save before exit if changes have been made
* Ctrl+k – cuts a line of text
* Ctrl+u – uncut text where the cursor is

Raspberry Pi Initial Setup

## Image

2016-02-09-raspbian-jessie.img

## Windows tools for SD card format and write

* SDFormatter
* Win32DiskImager

## Raspberry Pi Required Installations

* hostapd
* hostap-utils
* isc-dhcp-server
* xbitmaps
* xtightvncviewer
* libmtp-runtime
* python-smbus
* sudo apt-get install hostapd hostap-utils isc-dhcp-server xbitmaps xtightvncviewer

## Raspberry Pi Optional Installations

* tightvncserver
* iw
* python-matplotlib
* sudo apt-get install tightvncserver iw python-matplotlib
* sudo apt-get update
* sudo apt-get upgrade

## Setting up the Raspberry Pi

* sudo raspi-config
  + Expand Filesystem
  + Advanced
    - A6 SPI -> Enable -> Load module by default
    - A7 I2C -> Enable -> Load module by default
    - A8 Serial -> Enable

## Create an Archive

Create a new directory that will archive your raspberry pi data, giving todays date in the name

* mkdir /home/pi/Desktop/archived[date]

Then recursive copy to archive etc/ folder from original filesystem into new folder. (Use sudo for recursive copy of etc/)

* sudo cp –r /etc /home/pi/Desktop/archived[date]

## Disable Boot Messages

Switch off the tty terminal of the UART pins to create a “silent” boot

* sudo nano /boot/cmdline.txt

dwc\_otg.lpm\_enable=0 console=tty3 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=noop rootwait loglevel=3

Then stop and disable the ttyAMA0 device

* sudo systemctl stop serial-getty@ttyAMA0.service
* sudo systemctl disable serial-getty@ttyAMA0.service

Boot message “Uncompressing Linux … done, booting the kernel.” on each reboot

Message sent at 115200 baud. It’s just a transient at boot-up.

No “recovery mode” login prompts (not Noobs).

After boot, /dev/ttyAMA0 sits at 9800 baud.

See post 283774 on the Raspberry Pi forum: One solution is to decompress the kernel.

Setup Virtual Network Computing (VNC) on Boot

* sudo nano /etc/lightdm/lightdm.conf

[VNCServer]  
enabled=true  
command=Xvnc  
port=5900  
width=1024  
height=768  
depth=32

## Lightdm VNCServer Options

enabled – boolean to enable/disable the server  
command – command used to access the server directly  
port – the port used to connect to the server  
width – width of the screen that will be on the connecting device  
height – height of the screen that will be on the connecting device  
depth – colour depth of the screen that will be on the connecting device

Setting Up Ad-Hoc on Boot

* sudo nano /etc/rc.local

\_IP=$(hostname –I) || true  
if [ “$\_IP” ]; then

printf “My IP address is %s\n” “$\_IP”

fi  
ifdown --force wlan0; iwconfig wlan0 mode ad-hoc  
iwconfig wlan0 essid “RPiAdHocNetwork”  
ifconfig wlan0 192.168.x.y  
netmask 255.255.255.0  
  
exit 0

## Connecting to the VNC server through any network method

Make sure that DeviceA and DeviceB are connected. If you are using an ad-hoc connection then both devices need to be setup in the exact same way except they need to have differing IP addresses. If this is through WAP then the client just needs to connect to the host. To check to see if they are connected check by pinging one device from the other one.

* ping [DeviceA/BIP]

Then connect to the other device by using the following command:

* sudo vncviewer [ip]:5900

This will then bring up the devices graphical user interface for you to interact with it.

General Interface File Setups

## Wi-Fi

* sudo nano /etc/network/interfaces-wifi

# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd  
# For static IP, consult /etc/dhcpcd.conf and ‘man dhcpcd.conf’

# Include diles from /etc/network/interfaces.d:  
#source-directory /etc/network/interfaces.d

auto lo

iface lo inet loopback  
iface eth0 inet dhcp  
iface usb0 inet static

address 192.168.x.y  
netmask 255.255.255.0  
network 192.168.x.0  
broadcast 192.168.x.255

allow-hotplug wlan0  
iface wlan0 inet manual  
wpa-roam /etc/wpa\_supplicant/wpa\_supplicant.conf  
iface default inet dhcp

## Ad Hoc

* sudo nano /etc/network/interfaces-adhoc

# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd  
# For static IP, consult /etc/dhcpcd.conf and ‘man dhcpcd.conf’

# Include diles from /etc/network/interfaces.d:  
#source-directory /etc/network/interfaces.d

auto lo

iface lo inet loopback  
iface eth0 inet dhcp  
iface usb0 inet static

address 192.168.x.y  
netmask 255.255.255.0

allow-hotplug wlan0  
iface wlan0 inet static

wireless-channel 1  
wireless-essid RPiAdHocNetwork  
wireless-mode ad-hoc

## Wireless Access Point (WAP)

* sudo nano /etc/network/interfaces-hotspot

# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd  
# For static IP, consult /etc/dhcpcd.conf and ‘man dhcpcd.conf’

# Include diles from /etc/network/interfaces.d:  
#source-directory /etc/network/interfaces.d

auto lo  
allow-hotplug wlan0  
iface wlan0 inet static  
 address 10.5.5.1  
 netmask 255.255.255.0

**NB: Where x and y are given in the IP Address will need to be replaced with integers to work. With all occurrences of x are the same integer and all occurrences of y are the same integer**

Wireless Access Point (WAP) Setup

## Primary installs

* sudo apt-get install hostapd isc-dhcp-server hostap-utils iw

## Check the dongle supports access point

* iw list | less

Check for an entry relating to AP mode:

Supported interface modes:  
 \* IBSS  
 \* managed  
  **\* AP  
 \* AP/VLAN**  
 \* WDS  
 \* monitor  
 \* mesh point

## Set RPi to use static IP Address

* sudo nano /etc/network/interfaces

# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd  
# For static IP, consult /etc/dhcpcd.conf and ‘man dhcpcd.conf’

# Include diles from /etc/network/interfaces.d:  
#source-directory /etc/network/interfaces.d

auto lo  
allow-hotplug wlan0  
iface wlan0 inet static  
 address 10.5.5.1  
 netmask 255.255.255.0

## Configure the DHCP Server

* sudo nano /etc/dhcp/dhcpd.conf

Comment out the domain name servers as we don't need them for this set-up. They can be commented out by adding a hash character at the start of each line.

#option domain-name "example.org";  
#option domain-name-servers ns1.example.org, ns2.example.org;

Make the authoritative by un-commenting (removing the # character) entry:

# If this DHCP server is the official DHCP server for the local  
# network, the authoritative directive should be uncommented.  
authoritative;

Edit the following:

# A slightly different configuration for an internal subnet.  
subnet 10.5.5.0 netmask 255.255.255.0 {  
 range 10.5.5.100 10.5.5.150;  
 option routers 10.5.5.1;  
 option broadcast-address 10.5.5.255;  
 default-lease-time 600;  
 max-lease-time 7200;  
}

Now edit the default start script

* sudo nano /etc/default/isc-dhcp-server

INTERFACES="wlan0"

## Configure the Raspberry Pi as a WAP

* sudo nano /etc/hostapd/hostapd.conf

Add the following:

# Host access point config file   
# device name   
interface=wlan0

# Driver interface  
driver=nl80211

# SSID for the network  
ssid=RPiHotspot

# set appropriate country parameters (maybe required for regulatory reasons)  
country\_code=GB

# Operation mode - for 802.11n still use g to indicate using same band as g devices  
hw\_mode=g

# set channel - channel=0 for Automatic Channel Select  
channel=0

# mac address access list - 0 = accept unless in deny  
macaddr\_acl=0

## add deny rules here if required

#deny\_mac\_file=/etc/hostapd/hostapd.deny

# Use shared key authentication  
auth\_algs=1

# Enable WPA2  
wpa=2

# set passphrase  
wpa\_passphrase=raspberry

# Use WPA PSK  
wpa\_key\_mgmt=WPA-PSK

# Pairwise cipher for WPA (v1)   
wpa\_pairwise=TKIP

# Pairwise cipher for RSN/WPA2   
rsn\_pairwise=CCMP

Finally edit the default startup file for hostapd

* sudo nano /etc/default/hostapd

DAEMON\_CONF="/etc/hostapd/hostapd.conf"

## Enabling a Higher Colour Depth

* sudo nano /boot/config.txt

Edit it to look like this:

# uncomment the following to adjust overscan. Use positive numbers if console  
# goes off screen, and negative If there is too much border  
#framebuffer\_width=1280  
#framebuffer\_height=7290  
framebuffer\_depth=32  
framebuffer\_ignore\_alpha=1

## Network Switch Script

A script has been introduced to the project to help with configuring the LAN (Local Area Network)/WAP (Wireless Access Point). It also allows for the restarting of the networking service that is built into the Raspberry Pi itself.

Using the script