Raspbian
Raspbian tuning and optimisations

Login as root

All the items will need to be done as the root user. Make sure you become root after every restart.
Shell

```
sudo su
```

root@raspberrypi:

```
free -htl
```

Will return something like this

```
total used free shared buffers cached

Mem: 215M 46M 168M OB 8.4M 21M

Low:215M 46M 168M

High: OB OB OB

-/+ buffers/cache: 16M 199M

Swap: 99M OB 99M

Total: 31 5M 46M 268M
```

Replacing OpenSSH with Dropbear | Save: +10MB RAM

This will reduce the memory consumption by 10MB, however you will be sacrificing: ssh portforwarding and ssh
logging.

install dropbear

openssh-client is installed to provide SCP support for the dropbear server
Shell

```
apt-get install dropbear openssh-client
```

Stop OpenSSH server

you will not loose your current SSH connection, if you are connected via ssh
Shell

```
/etc/init.d/ssh stop
```

Enable dropbear to start at boot-time

```
sed -i 's/NO_START=1/NO_START=0/g' /etc/default/dropbear
```

Start Dropbear

```
/etc/init.d/dropbear start
```

Make sure you can connect to the server over ssh (using an ssh client).

Disable OpenSSH, but allow it to be used for SSH / SFTP

```
update-rc.d ssh disable
```

Replace Bash shell with Dash shell | Save: +1 MB RAM

Replacing Bash with Dash will increase the system's overall performance ie. speed up the system boot, reduce disk
space, use fewer libraries (save memory) and is more reliable:

Dash is an acronym for Debian Almquist shell (dash). It is a Unix and Linux shell which is much smaller than bash
but still aiming at POSIX-compliancy. dash is a POSIX-compliant implementation of/bin/sh that aims to be as small
as possible, dash is a direct descendant of the NetBSD version of ash (the Almquist SHell), ported to Linux in early
1997. It was renamed to dash in 2002.

```
dpkg-reconfigure dash
```

Use dash as the default system shell (/bin/sh)? Yes

Enable a 512MB swapfile

Customize the size of the swap file, 512MB recommended.

```
echo "CONF_SWAPSIZE=512" > /etc/dphys-swapfile
```

Initialize the swapfile

```
dphys-swapfile setup
```

Start/enable the swapfile

```
dphys-swapfile swapon
```

Enable better usage of the swap

Default swappiness is 1 , we will change this value to 1 0, which will allow for better memory usage at the expense of
more swap usage, note: this could reduce the life of your sdcard.

```
sed -i 's/vm.swappiness=1/vm.swappiness=10/g' /etc/sysctl.conf
```

Purge cached block devices before cached filesystem entries

```
echo 'vm.vfs_cache_pressure=50' » /etc/sysctl.conf
```

Optimize / mount

```
sed -i 's/defaults,noatime/defaults,noatime,nodiratime/g' /etc/fstab
```

Disable IPv6

Most users will not need ipv6 support and if you are only using a local there is no need. Disable it to save the resources and speed up networking.

```
echo "net.ipv6.conf.all.disable_ipv6=1" > /etc/sysctl.d/disableipv6.co
nf
```

Disable the kernel module

```
echo 'blacklist ipv6' » /etc/modprobe.d/blacklist
```

Remove IPv6 hosts

```
sed -i 7::/s% A %#%g' /etc/hosts
```

Reboot

```
shutdown -r now
```

Overclock cpu, sdram and gpu core without increasing voltage

Overclocking is now Officially Supported.

"None" "700MHz ARM, 250MHz core, 400MHz SDRAM, 0 overvolt"
"Modest" "800MHz ARM, 300MHz core, 400MHz SDRAM, 0 overvolt"
"Medium" "900MHzARM, 333MHz core, 450MHz SDRAM, 2 overvolt"
"High" "950MHz ARM, 450MHz core, 450MHz SDRAM, 6 overvolt"
"Turbo" "1000MHz ARM, 500MHz core, 500MHz SDRAM, 6 overvolt"

Select one of the following options:

800Mhz permanently: The settings below are safe for all Raspberry Pi's

```
echo -e "arm_freq=850\nsdram_freq=450\ncore_freq=350\nforce_turbo=1" »
 /boot/config.txt
```

OR

850Mhz permanently: The settings below are only to be used if you have heatsinks installed

```
echo -e "arm_freq=950\nsdram_freq=500\ncore_freq=450\nforce_turbo=1" »
 /boot/config.txt
```

OR

700Mhz-1000Mhz dynamic: Scales the cpu frequency according to the load

```
1 echo -e "force_turbo=0" » /boot/config.txt
```

Reboot

```
shutdown -r now
```

Replace Deadline Scheduler with NOOP Scheduler

NOOP scheduler is best used with solid state devices such as flash memory.
Shell

```
sed -i 's/deadline/noop/g' /boot/cmdline.txt
```

Reboot

```
shutdown -r now
```

Replace rsyslogd with inetutils-syslogd and remove useless logs

Reduce memory and cpu usage. We just need a simple vanilla syslogd. Also there is no need to log so many files.
Just dump them into /var/log/(cron/mail/messages)

remove rsystlog

```
apt-get -y remove --purge rsyslog
```

install syslogd

```
apt-get -y install inetutils-syslogd
```

Create a vanilla syslogd setup
Stop syslogd

```
service inetutils-syslogd stop
```

Remove old logs

```
/var/log/Mog /var/log/mail.* /var/log/debug /var/log/syslog; do [ -f "
$file" ] && rm -f "$file"; done
```

```
[ -d 7var/log/$dir" ] && rm -rf "/var/log/$dir"; done
```

Create syslog.conf

```
echo -e "*.*;mail.none;cron.none\t -/var/log/messages\ncron.*\t -/var/
log/cron\nmail.*\t -/var/log/mail" >
/etc/syslog.conf
```

Configure logrotate

```
mkdir -p /etc/logrotate.d

echo -e 7var/log/cron\n/var/log/mail\n/var/log/messages {\n\trotate

\n\tweekly\n\tmissingok\n\tnotifempty\n\tcompress\n\tsharedscripts\n\t
postrotate\n\t/etc/init.d/inetutils-syslogd
reload >/dev/null\n\tendscript\n}" > /etc/logrotate.d/inetutils-
syslogd
```

Start syslogd

```
service inetutils-syslogd start
```

Set a static IP and remove DHCP-Client

Reduces CPU and memory.
Create a backup config

```
cp -f /etc/network/interfaces /etc/network/interfaces. dhcp-backup
```

Edit file: /etc/network/interfaces

replace

```
iface eth0 inet dhcp
```

with

```
iface eth0 inet static

#set your static IP below
address 192.168.1.107

#set your default gateway IP here
gateway 192.168.1.1

netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
```

Reboot

```
shutdown -r now
```

Unique solution ID: #1048
Author: Karl Hudgell
Last update: 2016-01-20 16:23