Sommario

Raspberry - Come fare... 3
Creare l'utente root per copiare file con FILEZILLA 3
Creare un ramdisk 3
Duplicare una sd 4
Installare un programma per inviare mail 4
Installing and configuring SSMTP 5
Sending emails from command line 6
Sending emails from PHP scripts 7
Troubleshooting 7
Settare il web server 8

Raspberry - Come fare...

Creare l'utente root per copiare file con FILEZILLA

Per creare l'utente root sul linux del raspi digita questo comando per crearlo:

```
sudo passwd root
```

ti chiederà di inserire per due volte una password che non vedrai, per loggarti con FileZilla usa utente root e password che hai scelto.

Creare un ramdisk

```
#!/bin/basho
# ramdisk.sh

#  Un "ramdisk" è un segmento della memoria RAM
#  che si comporta come se fosse un filesystem.
#  Presenta il vantaggio di un accesso velocissimo (tempo di lettura/scrittura)
#  Svantaggi: volatilità, perdita di dati al riavvio o in caso di mancanza di
#             corrente elettrica, meno RAM disponibile al sistema.
#
#  Cos'ha di buono un ramdisk?
#  Tenere una serie elevata di dati, come una tabella o un dizionario,
#  su un ramdisk ne velocizza la consultazione, perché l'accesso
#  alla memoria è molto più veloce di un accesso al disco.

E_NON_ROOT=70                  # Deve essere eseguito da root.
NOME_ROOT=root

MOUNTPT=/mnt/ramdisk
DIMENSIONE=2000                # 2K blocchi (modificare in base alle esigenze)
DIMENSIONEBLOCCO=1024          # 1K (1024 byte)
DISPOSITIVO=/dev/ram0          # Primo dispositivo ram

nomeutente=`id -nu`
if [ "$nomeutente" != "$NOME_ROOT" ]
then
  echo "Devi essere root per eseguire \"`basename $0`\"."
  exit $E_NON_ROOT
fi

if [ ! -d "$MOUNTPT" ]         #  Verifica se già esiste il punto di mount,
then                           #  in modo che non ci sia un errore se lo script
  mkdir $MOUNTPT               #  viene eseguito più volte.
fi

############################################################################
dd if=/dev/zero of=$DISPOSITIVO count=$DIMENSIONE bs=$DIMENSIONEBLOCCO
                               #  Pone il dispositivo RAM a zero.
                               #  Perché questa operazione è necessaria?
```

```
mke2fs $DISPOSITIVO            #  Crea, su di esso, un filesystem di tipo ext2.
mount $DISPOSITIVO $MOUNTPT    #  Lo monta.
chmod 777 $MOUNTPT             #  Abilita l'accesso al ramdisk da parte di un
                               #  utente ordinario.
                               #  Tuttavia, si deve essere root per smontarlo.
###############################################################################
#  Occorre verificare se i precedenti comandi hanno avuto successo,
#  altrimenti potrebbero verificarsi dei problemi.
#  Esercizio: modificate lo script per renderlo più sicuro.

echo "\"$MOUNTPT\" ora è disponibile all'uso."
#  Il ramdisk è accessibile, per la registrazione di file, anche ad un utente
#  ordinario.

#  Attenzione, il ramdisk è volatile e il contenuto viene perso
#  in caso di riavvio del PC o mancanza di corrente.
#  Copiate tutto quello che volete salvare in una directory regolare.

# Dopo un riavvio, rieseguite questo script per reimpostare il ramdisk.
# Rifare il mount di /mnt/ramdisk senza gli altri passaggi è inutile.

#  Opportunamente modificato, lo script può essere invocato in
#  /etc/rc.d/rc.local per impostare automaticamente un ramdisk in fase di boot.
#  Potrebbe essere appropriato, ad esempio, su un server database.

exit 0
```

Duplicare una sd

Io di solito, da linux:

```
1. sulla vecchia SD monto le 2 partizioni originali
2. salvo il contenuto delle partizioni originali con tar
3. smonto le partizioni
4. sulla nuova SD creo le 2 partizioni a mano con fdisk
5. le formatto con mkfs.vfat e mkfs.ext4
6. monto le nuove partizioni
7. decompatto i file .tar precedentemente creati dentro le 2 partizioni
```

Installare un programma per inviare mail

There are many cases when it can be very useful to be able to send emails from the Raspberry Pi to arbitrary recipients. This is not the same as having a real MTA running on the Pi (like Sendmail, Postfix, Exim, QMail, etc.), which can also receive and store emails. In the following we are only going to cover the possibility of sending emails, not receiving. In most cases this is enough, as people tend to use GMail, Yahoo! Mailand other major email service providers and they store their emails on the servers of these providers. Still, sending out emails from the Raspberry Pi can come in handy in many situations. For example, you could have some sensors connected to the GPIO pins of the Piand you could program the Pi to send you an email when the temperature in the room rises above or drops below certain threshold values, when a gas sensor registers unwanted gas leaks or when the measured voltage of a monitored battery becomes too low. You could also have your Pi send you daily or weekly emails with summarized system data. Or maybe you could connect a webcam to the Raspberry Pi and set up some motion detection software, which would send you an email as soon as it detects motion in a

given area of your house. Maybe we don't even need to go this far. Maybe you are hosting a WordPress website on your Raspberry Pi and you would like to provide your readers with the possibility to subscribe to the posts. This all means that the Pi needs to be able to send out emails, which, unfortunately, can be complicated to accomplish.
In order to achieve this we are going to install a piece of software called SSMTP, which is a simple tool for sending emails. We are also going to configure PHP in a way which is going to make it possible to send emails from inside PHP scripts. This way it's going to be easy for web applications (like WordPress plugins) to send mails to chosen recipients.
Many email servers today have very strict rules for accepting emails. For example if the email is not coming from a machine with a static IP address, they might classify the email as SPAM. We don't want that to happen with the emails sent from the Raspberry Pi, so we are going to send the emails to a Goggle server, which will send them forward to the real recipients. In order to be able to accomplish this, you must have a GMail account.

Installing and configuring SSMTP

1. Make sure your repositories are up-to-date:

apt-get update

2. Install SSMTP and mail utilitites:

apt-get install ssmtp

apt-get install mailutils

3. Edit the SSMTP configuration file:

nano /etc/ssmtp/ssmtp.conf

a) Mandatory lines:

root=postmaster
mailhub=smtp.gmail.com:587
hostname=raspberrypi
AuthUser=YourGMailUserName@gmail.com
AuthPass=YourGMailPassword
UseSTARTTLS=YES

Be sure to specify the correct GMail user name and password here, otherwise you will get authentication errors.

If the host name of your Raspberry Pi is different from "raspberrypi", specify your actual host name here instead.

b) Optional lines:

rewriteDomain=your.domain

Specify this if you would like the outgoing emails to appear to be sent from your.domain (instead of from gmail.com).

FromLineOverride=YES

Specify this if you would like SSMTP to leave the From field of the emails untouched. Otherwise it will overwrite the from field with the name of the Linux user which sends the email. The overwriting name is taken from the 5th value in the line that corresponds to the sending user, from the */etc/passwd* file. If you plan to send emails from a website (for example from a WordPress plugin) and wish to have nice sender names like "John Doe", I recommend commenting this line (which is equal to setting the value to NO), otherwise your website will only be

able to send emails with less nice sender names, like johndoe@your.domain. In other words, you probably want SSMTP to overwrite the sender field with a nice name taken from the */etc/passwd* file.

4. Edit the SSMTP aliases file:

nano /etc/ssmtp/revaliases

This file contains data about the email accounts for existing Linux users in the format *local_account:outgoing_address:mailhub[:port]*

You should create one line for all the users in your system from which you plan to be able to send emails. For example:

root:root@your.domain:smtp.gmail.com:587
www-data:yourwebpagesname@your.domain:smtp.gmail.com:587

In case you wish to send out emails from a WordPress plugin, you must make sure that you have a line for the user **www-data**, which is the user under which WordPress runs.

5. Set the permissions of the SSMTP configuration file:

chmod 774 /etc/ssmtp/ssmtp.conf

The permissions of the file */etc/ssmtp/ssmtp.conf* determine who will be able to send emails from the Raspberry Pi. By default this file is owned by the user root and the group of the file is also root. So if you want other users, like www-data to be able to send emails (which you definitely want if you're using a WordPress plugin for example to send out emails), then you need to give read rights to the users who are not the owner of the file and not in the group of the file. The above permissions (774) mean that the owner (root) will be able to read/write/execute the file (7), the other users in the root group will be able to do the same (7) and the users which are not in the group will only have the right to read the file (4). For more details type **chmod –help**.

If you prefer not to allow every user on your system to send emails, then add the www-data user (or the user who you would like to grant permission for sending emails) to the root group and only give the rights to the users in this group:

sudo usermod -a -G root www-data
chmod 770 /etc/ssmtp/ssmtp.conf

Be careful though. Adding the www-data user to the root group might sometimes not be very safe, as it will allow your website to do many things on your system.

6. Nice sender names:

If you would like your website (WordPress for example) to be able to send emails which appear to be sent from a person with a nice name like "John Doe" or "Your super web site" instead of from a simple sender name like you@your.domain, then you need to make sure that in the **/etc/ssmtp/ssmtp.conf** file the **FromLineOverride** line is commented (#) or set to NO and you need to give a nice name to the **www-data** user. You can do this by editing the passwords file:
nano /etc/passwd

Find the line corresponding to **www-data** and set the fifth value in it to a nice name like "Your super website".

Sending emails from command line

Once you're done with the above setup and configuration process, you can send emails very easily from command line, which is great because you can also put the sending into Bash scripts, which can be called by other applications running on your Pi. A simple example looks like this:

echo “Test text” | mail -s “Test Mail” targetperson@example.com

The text “Test text” is sent to the email address targetperson@example.com (you may also specify multiple addresses separated by spaces) with the subject “Test Mail”. For more options type **mail –help**.

Sending emails from PHP scripts

If you would also like to be able to send out emails from PHP scrips (which is the case if you plan to send emails from your website, perhaps from a WordPress plugin like Subscribe2), then you need to configure PHP to find the mail sending application.

1. Edit your PHP configuration file:

nano etc/php5/apache2/php.ini

Find (**F6**) the line which contains *sendmail_path* and set it to the appropriate value:

sendmail_path = /usr/sbin/sendmail -t -i

2. To test if PHP is indeed able to send out emails, create a file named **mailtest.php**and put the following code into it:

```
<?php// //
$to = “targetperson@example.com”;// //
$subject = “PHP Test mail”;// //
$message = “This is a test email”;// //
$from = “you@your.domain”;// //
$headers = “From:” . $from;// //
mail($to,$subject,$message,$headers);// //
echo “Mail Sent.”;// //
?>
```

```
Test by calling your script from command line:
```

PHP mailtest.php

Troubleshooting

If you fail to configure things properly, you might run into some errors when trying to send emails. Some common error messages are the following:

1. Authentication failure:

//**Authorization failed (535 5.7.1 http://support.google.com/mail/bin/answ … swer=14257 a1sm11494547eep.2 – gsmtp** //

The most probable cause of this is that you have specified a wrong username or password for your GMail account in the */etc/ssmtp/ssmtp.conf* file. Double check the values of the **AuthUser** and **AuthPass** fields.

2. Connection lost in the middle of processing:

send-mail: Connection lost in middle of processing

Chances are you specified the wrong port for the Google smtp somewhere in */etc/ssmtp/ssmtp.conf* or in */etc/ssmtp/revaliases*.

3. Your website (for example your WordPress plugin) gives an error message when you try to send an email from it or it appears to have sent the email but it never arrives to the recipient(s):

Check the last lines of your logs. There should be some relevant error messages in there:

tail /var/log/mail.log

tail /var/log/syslog

One of the most common problems is that you have not given the **www-data**user rights to read the */etc/ssmtp/ssmtp.conf* file. Check the permissions of the file:

ls -l /etc/ssmtp/ssmtp.conf

and if something is not right, set the permissions, for example:

chmod 774 /etc/ssmtp/ssmtp.conf

Settare il web server

Optional) Install APC:

APC (Alternative PHP Cache) is an application which caches the PHP code and user variables, which results in more responsive, faster websites and less congested servers. This is particularly important on the Raspberry Pi because it has limited hardware resources and we need to optimize the web server as much as possible. To install APC, simply type:

apt-get install php-apc

6. (Optional) Test APC:

You should have a PHP script named apc.php inside your /var/www directory. Test it in your web browser by entering http://yourwebsite.com/apc.php or yourip/apc.php in the address bar.

You should also see some APC-related info if you test with the phptest.php script created in step 2.

7. Restart Apache:

service apache2 restart

impostare servizio mail

Fonte:
Wiki - Progetti

Autore:
Walter62

Ultimo aggiornamento: **2023/01/01 17:03**