

Mount

Reload fstab

```
mount -a
```

Creare link al nome device

Aggiungere script e dare permessi di esecuzione

```
/etc/usbmount/mount.d/01_create_label_symlink
```

```
chmod +x /etc/usbmount/mount.d/01_create_label_symlink
```

File:01_create_label_symlink

```
#!/bin/sh
set -e

# Exit if device or mountpoint is empty.
test -z "$UM_DEVICE" && test -z "$UM_MOUNTPOINT" && exit 0

# get volume label name
label=`blkid -s LABEL -o value $UM_DEVICE`

# Ricrea sempre il link: potrebbe puntare a altro [/media/usb0][/media/usb1]...
ln -sf "$UM_MOUNTPOINT" "/var/run/usbmount/$label"

exit 0
```

Lo script legge in nome del **volume** dei device montato.

E crea un link in /var/run/usbmount/NomeDelVolume

Permessi a tutti sulle mount

File:/etc/usbmount/usbmount.conf

```
# Configuration file for the usbmount package, which mounts removable
# storage devices when they are plugged in and unmounts them when they
# are removed.

# Change to zero to disable usbmount
ENABLED=1

# Mountpoints: These directories are eligible as mountpoints for
# removable storage devices.
MOUNTPOINTS="/media/usb0 /media/usb1 /media/usb2 /media/usb3
             /media/usb4 /media/usb5 /media/usb6 /media/usb7"
```

```
# Filesystem types: removable storage devices are only mounted if they
# contain a filesystem type which is in this list.
FILESYSTEMS="vfat ext2 ext3 ext4 hfsplus"

MOUNTOPTIONS="sync,noexec,nodev,noatime,nodiratime"

# umask 022 gives 755 permissions to the whole disk
# -----
FS_MOUNTOPTIONS="-fstype=vfat,gid=www-data,uid=www-data,umask=022,sync \
                -fstype=ntfs,gid=www-data,uid=www-data,umask=022,sync"

VERBOSE=no
```

Utilizzo dei link (1)

Il nome del volume utilizzato sui device e' WWWMEDIA150 / WWWMEDIA930

Indipendentemente dal mountpoint */media/usb0* */media/usb1* ... in */var/www/html* ci sono link simbolici che puntano ai link creati col nome volume:

/var/www/html/usbwww150 → */var/run/usbmount/WWWMEDIA150*

/var/www/html/usbwww930 → */var/run/usbmount/WWWMEDIA930*

Utilizzo dei link (2)

L'utente specifico per connettersi (da fuori) in FTP/SFTP ha la particolarita' di non avere alcun permesso sulla propria */home* e di non appartenere ad altri gruppi che non al suo.

- Nome utente **utenteFTP**
- Nome gruppo **utenteFTP**

Con questi permessi puo accedere solo alla cartella */home/utenteFTP/ftp* in qui ci sono due cartella (saranno il mount point) dei dischi USB

/home/utenteFTP/ftp

dr..r..r.. 2 root utenteFTP 4096 Apr 15 12:21 ftpStoreData150

dr..r..r.. 2 root utenteFTP 4096 Apr 15 12:21 ftpStoreData930

I permessi sono 444 cosi' se i dischi USB non sono montati non dovresti riuscire a scriverci sopra (eviti di scrivere sulla SD del server) e la lasci vuota condizione essenziale per usarli come mount point.

Uno script verifica la presenza dei dischi USB (del loro mount) e crea un **loop mount** nella home di **utenteFTP**

Vedi file:*/var/batch/chkMountDir.sh*

```
#!/usr/bin/php
<?php
GLOBAL $sep, $pathLog, $fileLog;
$sep = "\n";
```

```
// -----
// Path del file di log
// Nome del file di log
// -----
$pathLog = "/var/batch/log/";
$fileLog = "chkMountDir"."_" . getOggi() . ".log";

// -----
// Inizio log
// -----
prnTitle("START");

// -----
// Primo controllo
// -----
// -----
// Path del link al disco USB
// Il link da verificare/creare
// -----
$pathHD      = "/var/run/usbmount/WWWMEDIA150/ftpStoreData";
$pathTarget = "/home/utenteFTP/ftp/ftpStoreData150";
checkFolderFTP_1($pathHD,$pathTarget);

// ---
$pathHD      = "/var/run/usbmount/WWWMEDIA930/ftpStoreData";
$pathTarget = "/home/utenteFTP/ftp/ftpStoreData930";
checkFolderFTP_1($pathHD,$pathTarget);

// -----
// Chiusura log
// -----
prnTitle(" STOP");

die;
// #####
// Sotto solo funzioni
// #####
// CheckFolderFTP_1 =====
// Se e' presente disco USB con etichetta WWWMEDIA150 viene creato il link
// /var/run/usbmount/WWWMEDIA150
// Questa funzione crea un loop link con [mount --bind] alla home di [utenteFTP]
// [utenteFTP] e' l'utente FTP con accesso limitato a questa cartella (chroot)
// param@pathHD      - Path del link al disco USB
// param@pathTarget - Il link da verificare/creare
// -----
function checkFolderFTP_1($pathHD,$pathTarget) {
    prnMessage(__FUNCTION__." START");

    // -----
    // Path del link al disco USB
    // -----
    $fileChk = "sonoSuUSB.txt"; // File di controllo (deve esistere)

    // -----
    // Comando eseguito (comando di mount)
    // -----

```

```
$cmd = "mount --bind";  
$cmd .= " ";  
$cmd .= $pathHD;  
$cmd .= " ";  
$cmd .= $pathTarget;  
  
// -----  
// Path di partenza  
// -----  
$is_lnkHD = is_dir($pathHD);  
if ( $is_lnkHD ) {  
    prnMessage("pathUsb EXISTS [\".$pathHD.\"]");  
} else {  
    prnMessage("pathUsb MISSING [\".$pathHD.\"]");  
}  
  
// -----  
// File di controllo  
// Nel folder di partenza  
// -----  
$is_fChk = file_exists($pathHD."/\".$fileChk);  
if ( $is_fChk ) {  
    prnMessage("fileChk EXISTS [\".$pathHD.\"/"\".$fileChk.\"]");  
} else {  
    prnMessage("fileChk MISSING [\".$pathHD.\"/"\".$fileChk.\"]");  
}  
  
// -----  
// Path di arrivo  
// -----  
$is_target = is_dir($pathTarget);  
if ( $is_target ) {  
    prnMessage("pathToLink EXISTS [\".$pathTarget.\"]");  
} else {  
    prnMessage("pathToLink MISSING [\".$pathTarget.\"]");  
}  
  
// -----  
// File di controllo  
// Nel folder di arrivo  
// -----  
$target_file = file_exists($pathTarget."/\".$fileChk);  
if ( $target_file ) {  
    prnMessage("fileChkLnk EXISTS [\".$pathTarget.\"/"\".$fileChk.\"]");  
} else {  
    prnMessage("fileChkLnk MISSING [\".$pathTarget.\"/"\".$fileChk.\"]");  
}  
  
// -----  
// Esiste il folder e il file di controllo  
// -----  
if ( $is_lnkHD && $is_fChk ) {  
    // -----  
    // Non esiste il link  
    // Non esiste il file di controllo nel link
```

```

// -----
if ( !$is_target || !$target_file ) {
    prnMessage("Faccio il MOUNT");

    prnMessage("CMD [\".$cmd.\"]");
    $risu = Array();

    // -----
    // Eseguo il comando
    // -----
    exec($cmd,$risu);

    // -----
    // Log del risultato
    // -----
    for ($i=0; $i<count($risu); $i++) {
        prnMessage(str_pad($i,3,"0",STR_PAD_LEFT)." | ".$risu[$i]);
    }

    // -----
    // Verifico se il MOUNT e' andato bene
    // -----
    prnMessage("Verifica dopo il MOUNT");

    // -----
    // Path di arrivo
    // -----
    $is_target = is_dir($pathTarget);
    if ( $is_target ) {
        prnMessage("pathToLink EXISTS [\".$pathTarget.\"] DOPO");
    } else {
        prnMessage("pathToLink MISSING [\".$pathTarget.\"] DOPO");
    }

    // -----
    // File di controllo
    // Nel folder di arrivo
    // -----
    $target_file = is_file($pathTarget."/".$fileChk);
    if ( $target_file ) {
        prnMessage("fileChkLnk EXISTS [\".$pathTarget.\"/".$fileChk."]
DOPO");
    } else {
        prnMessage("fileChkLnk MISSING [\".$pathTarget.\"/".$fileChk."]
DOPO");
    }

    // -----
    // I permessi alla cartella
    // -----
    if ( $is_target && $target_file ) {
        $cmd = "chmod 777 ".$pathTarget;
        prnMessage("CMD [\".$cmd.\"]");
        $risu = Array();

        // -----
        // Eseguo il comando

```

```
        // -----
        exec($cmd,$risu);

        // -----
        // Log del risultato
        // -----

        for ($i=0; $i<count($risu); $i++) {
            prnMessage(str_pad($i,3,"0",STR_PAD_LEFT)." | ".$risu[$i]);
        }
    }

} else {
    if ( $is_target && $target_file ) {
        prnMessage("Il folder e' presente");
    }
}

prnMessage(__FUNCTION__." STOP");
} // -----<CheckFolderFTP_1>-----

// PrnMessage =====
// -----
function prnMessage($testo) {
    GLOBAL $sep;

    $msg = "";
    $msg .= getAdesso();
    $msg .= " ".$testo;
    $msg .= $sep;

    writeLog($msg);
    echo $msg;
} // -----<PrnMessage>-----

// PrnTitle =====
// -----
function prnTitle($testo) {
    GLOBAL $sep;

    $title = "[".__FILE__."] ".$testo;
    $msg = "";
    $msg .= str_repeat("-", strlen($title)).$sep;
    $msg .= getAdesso().$sep;
    $msg .= $title.$sep;
    $msg .= str_repeat("-", strlen($title)).$sep;

    writeLog($msg);
    echo $msg;
} // -----<PrnTitle>-----

// GetAdesso =====
// -----
```

```

function getAdesso() {

    $ms = "[".
        // giorno-mese-anno ora:minuti:secondi
        date("d")."-".
        date("m")."-".
        date("Y")." ".
        date("H").":".
        date("i").":".
        date("s").
        "]"";

    return $ms;
} // -----<GetAdesso>-----

// GetOggi =====
// -----
function getOggi() {

    // giorno-mese-anno ora:minuti:secondi
    $ms = date("Y")."-".
        date("m")."-".
        date("d")."";

    return $ms;
} // -----<GetOggi>-----

// WriteLog =====
// -----
function writeLog($msg) {
    GLOBAL $sep, $pathLog, $fileLog;

    $filename = $pathLog.$fileLog;
    $mode = "a";
    if ( $handle = fopen($filename,$mode) ) {
        $acapo= "";
        // -----
        // Se non ho un [a capo] lo aggiungo
        // -----
        $pos = strpos($msg,"\n",1);
        if ($pos === false) {
            $acapo = $sep;
        }

        fwrite($handle,$msg.$acapo);
    }
} // -----<WriteLog>-----
?>

```

Se se tutto va bene viene fatta la **mount** i percorsi puntano ai device esterni e sono scrivibili

/home/utenteFTP/ftp

drwxrwxrwx 2 utenteFTP utenteFTP 4096 Apr 11 12:21 ftpStoreData150

Last
update: 2026/03/02 12:33 brezza:chicco:software:mount <https://mmongodi.duckdns.org/dokuwiki/doku.php?id=brezza:chicco:software:mount&rev=1772451192>

drwxrwxrwx 2 utenteFTP utenteFTP 4096 Apr 11 12:21 ftpStoreData930

From:
<https://mmongodi.duckdns.org/dokuwiki/> - BrezzaWiki

Permanent link:
<https://mmongodi.duckdns.org/dokuwiki/doku.php?id=brezza:chicco:software:mount&rev=1772451192>

Last update: **2026/03/02 12:33**

