- Ubuntu
- Forum
- Wiki
- Linux
- Paste
- Chat

搜索 [ ] 进入 搜索

- UbuntuHelp
- 讨论
- 编辑
- 历史
- 简体
- 繁体

- 导航
  - 首页
  - 最近更改
  - 随机页面
  - 页面分类
  - 帮助
- 编辑
  - 编辑指南
  - 沙盒
  - 当前事件
  - 字词处理
- 工具箱
  - 链入页面
  - 链出更改
  - 所有特殊页面
- 个人工具
  - 登录

UbuntuHelp:AufsRootFileSystemOnUsbFlash

出自Ubuntu中文

文章出处:	https://help.ubuntu.com/community/aufsRootFileSystemOnUsbFlash
点击翻译:	English

1. title AUFS (Another Union File System) Root File System On Usb Flash

目录

AUFS support in main line kernel

I, Nicholas A. Schembri, attended the South East Linux Fest 2009. Peter Graner talked about the Ubuntu kernel for the 9.10 release. He pointed out that AUFS has failed to be included in the main line kernel for the third time. This was also noted in UbuntuWiki:FoundationsTeam/Specs/KarmicAufsUpdateManager I will looking into the replacement and post a link here. As of 2009.11.20 aufs has not been removed. I will contact Peter Graner about the status of 10.4 lts. The following sections will describe how to use aufs on a / (root file system). Aufs allows the creation of a root file system on a read only flash device that feel just like a hard disk. This is an alternative to the livecd image based USB flash systems. Note: Aufs has many other uses. This is not a how to on aufs. This adventure was inspired by the Voyage Linux Distribution.

Introduction

Flash has a limited number of write operations, 10,000 to 1,000,000. It would be

nice if we could boot a system on a read only file system to protect the flash. LiveCd have done this for years. A Live cd just does not feel like a real system. Goals: <
> 1.The system must protect the flash.<
> 2.The system must be ease to update and upgrade. <
> 3.The system must feel like the real thing. <
> 4.Keep It Simple and Small, KISS. <
>

Requirements

Install Ubuntu 8.04, 4+ GB Flash device(USB or sdhc Class 4/6).

Install Ubuntu

Use the livecd to install to flash device.

Before rootaufs script

This is from the eeepc test. Ubuntu 8.04 was installed on a sdhc class 4 flash device. <
> The system is booted off of /dev/sdb. The systems main hard disk, /dev/sda1, is not mounted.

```
rootaufs:~$ df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sdb1              3916992   2044272   1673744  55% /
varrun                 1033584       100   1033484   1% /var/run
varlock                1033584         0   1033584   0% /var/lock
udev                   1033584        40   1033544   1% /dev
devshm                 1033584        12   1033572   1% /dev/shm
lrm                    1033584     38176    995408   4% /lib/modules/2.6.24-12-generic/volatile
/dev/sda1              3842376   3238736    408452  89% /media/sda1
```

After rootaufs script

After the rootaufs script is installed, the system is rebooted. <
> /dev/sdb1 is now mounted on /ro in a read only state. The system can not write to /ro. Aufs-tmpfs is a ram disk mounted on /rw in a read write state. The aufs is using both /ro and /rw to mount / and run the system. Everything work just the way you would expect. <
> Because changes are stored in ram, nothing is saved between reboots. See remountro and remountrw for more information on making changes without rebooting. Note: the system is running from a ram disk. Do not fill the ram disk with your log files. <
> Updates to the system are simple. Remove aufs=tmpfs from the end of the command line in grub before the system boots and the system will boot in the “Before” state. <
>

```
rootaufs:~$ df
Filesystem           1K-blocks      Used Available Use% Mounted on
varrun                 1033584       104   1033480   1% /var/run
varlock                1033584         0   1033584   0% /var/lock
udev                   1033584        40   1033544   1% /dev
devshm                 1033584        12   1033572   1% /dev/shm
lrm                    1033584     38176    995408   4% /lib/modules/2.6.24-12-generic/volatile
/dev/sdb1              3913620   2130640   1584180  58% /ro
aufs                   1033584      3668   1029916   1% /
aufs-tmpfs             1033584      3668   1029916   1% /rw
```

Tested Laptop

System: Toshiba<
> RAM: 2GB<
> Flash: USB Flash 4GB <
> System: TOSHIBA Satellite A305 <
> Version: PSAG0U-02D00M <
> RAM: 4 GB<
> Flash: 8 GB SDHC <
> (Good Linux Laptop, boots SDHC card, Thank you best buy for helping test ubuntu 8.04.1 on each of your 27 demo laptops)<
> System HP tx1000<
> RAM: 2GB <
> Flash: SDHC 8GB<
> (This is not the best laptop for newbee's) <
>

Tested eeepc

Note: Tested with Alpha 4 and apt-get dist-upgrade to Alpha 6. Tested with 8.04 beta 1.<
> System: eeepc<
> RAM: 2GB<
> Flash: SDHC 8GB || 4GB Class 4<
> Note: When installing on flash, use advanced option to put the boot loader on the flash drive, /dev/sdb . <
>

Step by Step eeepc

Note: Backup your data. If you make a mistake, you can over write /dev/sda on the eeepc with the ubuntu installer.<
> This will not change the data on /dev/sda (internal Flash)<
> Put the SDHC card in the side of the eeepc. Boot UBUNTU 8.04 - beta from an external USB CD-Rom. <
>

Open Applications->Accessories->Terminal

If mounted, umount all disks.

```
# umount all devices before installing ubuntu
 sudo umount /dev/sda1
 sudo umount /dev/sdb1
```

Click Install icon on desktop

Step 1 → Step 3

Forward

Step 4

Pick Manual<
> Remove all partitions from device /dev/sdb<
> Select /dev/sdb → free space<
> Click New<
> Select Prmary,Max size, Beginning, Ext2, / <
> Note: When you exit the partition editor, you will be asked to enable swap; just continue.<
> Or <
> Pick “Guided - use entire Disk”<
> Pick “SCSI4 (0,0,0) (sdb) - 4.1 GB USB2.0 CardReader SD0<
> Forward<
> Note: If you use "Guided", turn off swap in /etc/fstab after install.

Step 5

Name : <your name here><
> Password: <passwd><
> Name of this computer: rootaufs <
>

Step 6

Forward

Step 7

Click on Advanced<
> Install boot loader is checked.<
> Device for boot loader installation: <
> /dev/sdb <
> OK <
> Note: Please read and understand the information in the scroll box before you click Install.<
>

Install

Click Install.

Restart

Note: Hit ESC and boot off of the CardReader

First Boot

Update the System

```
 apt-get update
 apt-get dist-upgrade
 apt-get install aufs-tools
```

Install the rootaufs Script

```
echo aufs >> /etc/initramfs-tools/modules
# copy the script from "Script" section to rootaufs
vi /etc/initramfs-tools/scripts/init-bottom/rootaufs
chmod 0755 /etc/initramfs-tools/scripts/init-bottom/rootaufs
mv /etc/initramfs-tools/scripts/init-bottom/rootaufs /etc/initramfs-tools/scripts/init-bottom/__rootaufs
```

Remake the initramfs

```
update-grub
update-initramfs -u
```

Edit Grub Menu

This is how rootaufs is used. <
> Add aufs=tmpfs to the end of the menu.lst entry you wish to use read only. Do not add aufs=tmpfs after entries with "single".<
>

```
  #sed edit /boot/grub/menu.lst
  cp /boot/grub/menu.lst /boot/grub/menu.lst.aufs
  cat /boot/grub/menu.lst.aufs|sed  s/'ro quiet splash'/'ro quiet splash aufs=tmpfs'/ >/boot/grub/menu.lst

  # or vi edit

  vi /boot/grub/menu.lst
```

```
title           Ubuntu hardy (development branch), kernel 2.6.24-12-generic
root            (hd1,0)
kernel          /boot/vmlinuz-2.6.24-12-generic root=UUID=77a02dc5-aab7-41d5-a743-4659f2a16131 ro quiet splash a
initrd          /boot/initrd.img-2.6.24-12-generic
quiet
```

Adding software to the system is easy

When the grub loader appears, hit e to edit. Remove aufs=tmpfs and hit b, to boot. < > Make the needed changes and reboot.< >

Make Grub Menu changes persistent

The above method undoes your changes every time you run update-grub, this shouldn't be to big of a problem, since the root filesystem is going to be ro. But if you install a updated kernel in rw mode you would have to edit /boot/grub/menu.lst again. This method doesn't have that problem and also enables you to make a nice bootmenu entry for the rw mode find this part in /boot/grub/menu.lst

```
# defoptions=quiet splash
```

and add "aufs=tmpfs" to the last rule, so it looks like this now

```
# defoptions=quiet splash aufs=tmpfs
```

now find this part, it's a bit further down the file

```
# altoptions=(recovery mode) single
```

and add this line

```
# altoptions=(writable mode) quiet splash
```

and now that's part is looking like this

```
# altoptions=(writable mode) quiet splash
# altoptions=(recovery mode) single
```

after editing the file execute this command in the command line

```
sudo update-grub
```

Now when booting and you want to boot to the writable mode press 'esc' to enter the boot menu en select the 'writable mode' boot entry from your preferred kernel

Remountrw and Remountro

remountrw remounts /ro in a read / write mode. Files can now be copied from / to /ro.

```
sudo remountrw
```

remountro remounts /ro in a read only mode

```
sudo remountrw
```

/Ro

/ro is the real root file system. /ro is mounted ro by default and is not harmed by power cycling the system.<
> Use remountrw and remountro to make changes to the system without rebooting into single user mode.<
>

/RW

/rw is a file system in RAM. As the system runs files are updated on / . /rw is the real location of the files that are changed. <
> /rw is mounted to allow df to show the real space used by the read / write part of the file system. <
> Note: Removing a file from /rw restores the file found in /ro. Removing a the file from / hides the file found in /ro <
>

Home Accounts on /

Note: if your home account is on the root file system, your files are in ram and not saved.

rootaufs Script

This was sent as an email to Voyage Linux mailing list. Copy the following code to /etc/initramfs-tools/scripts/init-bottom/rootaufs

```
#! /bin/sh
#  Copyright 2008 Nicholas A. Schembri State College PA USA
#
#  This program is free software: you can redistribute it and/or modify
#  it under the terms of the GNU General Public License as published by
#  the Free Software Foundation, either version 3 of the License, or
#  (at your option) any later version.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public License
#    along with this program.  If not, see
#    <http://www.gnu.org/licenses/>.

# Thank you Voyage Linux for the idea, http://voyage.hk/ Great job on release 0.5
#
# Tested with 8.04.1
# tested with 9.10
#
#
# ****************************************************************************************
#
#                                 Change log
#
# 2008.08.01  Added debugging comments in "drop to a shell" section. grub option aufs=tmpfs-debug will stop the
#             reviewed *********** fix fstab on tmpfs ******************
#             rootaufs failed when system was booted with /dev/xxx and fstab had uuid= info.
#             BlaYO pointed out the best and simplest solution was to use grep -v. Grep replaces a sed one liner
#             Add the comment block to fstab
#
# 2009.12.07 Corrected issue caused by Apparmor.
#            Name changed to __rootaufs.
#
#

case $1 in
prereqs)
    exit 0
     ;;
esac

export aufs

for x in $(cat /proc/cmdline); do
    case $x in
    root=*)
        ROOTNAME=${x#root=}
         ;;
    aufs=*)
        aufs=${x#aufs=}
        case $aufs in
        tmpfs-debug)
            aufs=tmpfs
            aufsdebug=1
             ;;
        esac
         ;;
    esac
done


if [ "$aufs"  != "tmpfs" ]; then
    #not set in boot loader
    #I'm not loved. good bye
    exit 0
fi
```

```
# This is a simple overview of the steps needed to use aufs on the root file system and see the /rw and /ro  bra
# initramfs init-botton script
# move the root file system to aufs/unionfs readonly /ro
# root is mounted on ${rootmnt}
# create tmpfs on /rw
# create a aufs using /ro and /rw
# put some files on the tmpfs to fix mtab and fstab
# move aufs to rootmnt to finish the init process.
# No changes to the root file system are made by this script.
#
#  Why!
#  This will allow you to use a usb flash drive and control what is written to the drive.
#  no need to rebuild the squashfs file just to add a program.
#  boot to single user mode.  The system works the way you expect. boot aufs=tmpfs and no changes are written to
#  run ubuntu on an eeePC .

# Install
# Install ubuntu 8.04 Hardy. Hardy has aufs installed by default
# apt-get update
# apt-get dist-upgrade
# apt-get install aufs-tools
# echo aufs >> /etc/initramfs-tools/modules
# put this file in /etc/initramfs-tools/scripts/init-bottom/rootaufs
# chmod 0755 rootaufs
# # clean up menu.lst
# update-grub
# update-initramfs -u
# vi /boot/grub/menu.lst
# add aufs=tmpfs to the default entry.
# do not add this line to single user mode.
# boot to single user mode in order to install software.
# note: if your home account is on the root file system, your files are in ram and not saved.
#



echo
echo "       root-aufs:  Setting up aufs on ${rootmnt} as root file system "
echo

modprobe -q --use-blacklist aufs
if [ $? -ne 0 ]; then
    echo    root-aufs error:      Failed to load aufs.ko
    exit 0
fi

#make the mount points on the init root file system
mkdir /aufs
mkdir /rw
mkdir /ro

# mount the temp file system and move real root out of the way
mount -t tmpfs aufs-tmpfs /rw
mount --move ${rootmnt} /ro
if [ $? -ne 0 ]; then
    echo    root-aufs error:    ${rootmnt}  failed to move to /ro
    exit 0
fi


mount -t aufs -o dirs=/rw:/ro=ro aufs /aufs
if [ $? -ne 0 ]; then
    echo    root-aufs error:      Failed to mount /aufs files system
    exit 0
fi
```

```
#test for mount points on aufs file system
[  -d /aufs/ro ] || mkdir /aufs/ro
[  -d /aufs/rw ] || mkdir /aufs/rw

# the real root file system is hidden on /ro of the init file system.  move it to /ro
mount --move /ro /aufs/ro
if [ $? -ne 0 ]; then
    echo    root-aufs error:      Failed to move /ro /aufs/ro
    exit 0
fi

# tmpfs file system is hidden on /rw
mount --move /rw /aufs/rw
if [ $? -ne 0 ]; then
    echo    root-aufs error:      Failed to move /rw /aufs/rw
    exit 0
fi



#*********** fix fstab on tmpfs ******************
# test for /dev/sdx
# this is not on the real file system.  This is created on the tmpfs each time the system boots.
# The init process will try to mount the root filesystem listed in fstab. / and swap must be removed.
# the root file system must be mounted on /ro not on /

if [ "$aufsdebug" -eq 1 ]; then
    echo  "   root-aufs debug:    Remove the root file system and swap from fstab "
    echo
    echo
    echo  "         ROOTNAME $ROOTNAME "
    echo  "         resume   $resume   "
    echo
    echo  '     BlaYO pointed out that grep can be used to quickly remove '
    echo  '      the root file system from fstab. '
    echo
    echo  '     Thank you BlaYO for the debug info.'
    echo

fi
# old code
# I'm sure that sed can do this in one step but I want to correct on the rootname  not matching the root in fsta
#cat /aufs/ro/etc/fstab|sed -e s/$ROOTNAME/\#$ROOTNAME/ -e s/$resume/\#$resume/ >/aufs/etc/fstab

#Add the comment block to fstab
cat <<EOF >/aufs/etc/fstab
#
#   RootAufs has mounted the root file system in ram
#
#  This fstab is in ram and the real fstab can be found /ro/etc/fstab
#  the root file system ' / ' has been removed.
#  All Swap files have been removed.
#

EOF

#remove root and swap from fstab
cat /aufs/ro/etc/fstab|grep -v ' / ' | grep -v swap >>/aufs/etc/fstab
if [ $? -ne 0 ]; then
    echo    root-aufs error:      Failed to create /aufs/etc/fstab
    #exit 0
fi




# add the read only file system to fstab
#ROOTTYPE=$(/lib/udev/vol_id -t ${ROOT})
ROOTTYPE=$(cat /proc/mounts|grep ${ROOT}|cut -d' ' -f3)
ROOTOPTIONS=$(cat /proc/mounts|grep ${ROOT}|cut -d' ' -f4)
```

```
echo ${ROOT} /ro $ROOTTYPE $ROOTOPTIONS 0 0 >>/aufs/etc/fstab


# S22mount on debian systems is not mounting  /ro correctly after boot
# add to rc.local to correct what you see from df
#replace last case of exit with #exit
cat /aufs/ro/etc/rc.local|sed 's/\(.*\)exit/\1\#exit/' >/aufs/etc/rc.local
echo mount -f  /ro >>/aufs/etc/rc.local

# add back the root file system. mtab seems to be created by one of the init proceses.
echo "echo aufs / aufs rw,xino=/rw/.aufs.xino,br:/rw=rw:/ro=ro 0 0 >>/etc/mtab" >>/aufs/etc/rc.local
echo "echo aufs-tmpfs /rw tmpfs rw 0 0 >>/etc/mtab" >>/aufs/etc/rc.local
echo exit 0 >>/aufs/etc/rc.local

#  Copyright 2008 Joaquín I. Bogado García
#fix para apparmor, se desactiva y listo ( From the lethe project. )
rm /scripts/init-bottom/_apparmor
rm /aufs/etc/init.d/apparmor


#build remountrw
echo \#!/bin/sh >/aufs/bin/remountrw
echo mount -o remount,rw ${ROOT} >>/aufs/bin/remountrw
chmod 0700 /aufs/bin/remountrw

#build remountro
echo \#!/bin/sh >/aufs/bin/remountro
echo mount -o remount,ro ${ROOT} >>/aufs/bin/remountro
chmod 0700 /aufs/bin/remountro

# This should drop to a shell. (rewrite)
if [ "$aufsdebug" -eq 1 ]; then
    echo
    echo "   root-aufs debug:    mount --move /aufs ${rootmnt} "
    echo
    echo '   root-aufs debug:   init will stop here.   '
    echo
    exit 0
fi

mount --move /aufs ${rootmnt}

exit 0
```

Apparmor

Note: an issues has been posted to the forum, "AUFS root partition breaks networking" ,http://ubuntuforums.org/showthread.php?t=1220145 updated rootaufs works with 9.10 as of 2009.12.7 status: apparmor is now part of initramfs. Apparmor is starting at the before rootaufs can pivot the root file system into place. Apparmor is updating it's access control lists on a read only drive. This issue was not corrected in the live cd. When apparmor works on the live cd it should work with rootaufs.

Credits

nschembr - the entire turorial, and the rootaufs script<
> mannes - added [[UbuntuHelp:[Make Grub Menu changes persistent|Make Grub Menu changes persistent]]] section Joaquín I. Bogado García - Lethe project and Apparmor

error

Links

- Topic on this tutorial on the Ubuntu forums

取自“http://wiki.ubuntu.org.cn/UbuntuHelp:AufsRootFileSystemOnUsbFlash”

2个分类: 翻譯請求 | UbuntuHelp

此页面已被浏览过5,002次。

- 此页由Ubuntu中文用户Wikibot于2010年5月19日 (星期三) 21:38的最后更改。
- 关于Ubuntu中文
- 免责声明