

Backups on CD

Holger Nassenstein <backups@tuxoncd.de>

17 April 2000

Abstract

BackupOnCd 0.9.1 is a set of shell scripts which make gzipped multi-volume backups on CD-RW possible. It also makes the backup faster using cdrecord (<http://www.fokus.gmd.de/research/cc/g1one/employees/joerg.schilling/private/cdrecord.html>) in a parallel sub-shell and minimizes the amount of temporary needed disk space. Currently provided: tar and afio. It's not finished yet, but should be usefull. This description is also available as PDF (`../backuponcd.pdf`)  file. See the HISTORY (`../HISTORY`) for details on release.

Copyright Notice

Published under the terms of GNU GPL (<http://www.gnu.org/copyleft/gpl.html>), Version 2 or later by Holger Nassenstein.

Contents

1 Overview	1
1.1 Usage	1
1.2 What does it do?	1
1.3 How does it work?	2
2 Quick Reference	3
2.1 Installing BackupOnCd	3
2.1.1 Dependencies	3
2.1.2 Customizing (a must)	3
2.2 Enjoying	5
3 Details	7
3.1 The Burning Process	7
3.2 <code>toast.sh \$params fifo</code>	7
4 Important Options, Global Settings	11
4.1 <code>/usr/doc/backuponcd/samples/excludelist</code>	11
4.2 <code>/etc/backuponcd/global.rc</code>	11
5 Download / Contributes	15

Chapter 1

Overview

1.1 Usage

Just type `backuponcd -s (directory) -e (excludelist)`, where `(directory)` is the full path to the directory which you want to archive and `(excludelist)` is a grep-script.

The output of `backuponcd -h` is:

```
-s directory which is to archived
-e excludelist may be a empty file, usually a list of mountpoints,
  view example in /etc/backuponcd/excludelist
-a archiver (optional) overrides your setting (archiver=..)
  in /etc/backuponcd/global.rc
-rc global settings (optional).
  Overrides built in parameter "-rc /etc/backuponcd/global.rc".
  Must be full path to rc file.
```

If you want to generate a backup on a running system, some folders must be excluded, have a look on an example exclude list (see `‘/usr/doc/backuponcd/samples/excludelist’` on page 11).

1.2 What does it do?

The script generates a backup providing tar or afo relative to `(directory)` and stores it temporary in the `backuppath`, depending on your settings in `‘/etc/backuponcd/global.rc’` on page 11. The archive can have any length, it is splitted into smaller pieces and burned while the backup runs continuous.

This is done by using pipes and burning multisession CDs, so don't worry if the gzipped tarball of your system amounts 1.5 + Gbyte. In this case you need 3 CDs with a media cost of 6\$ if you use CD-RW.

1.3 How does it work?

Extracting an archive looks like this:

```
cat /cdwriter/tar.gz.* | tar --compare --gzip --file=- | \\  
tee -a /tmp/logfile
```

while the filenames of the archive are tar.gz.00, tar.gz.01 and so on.

A bit more complicated is reading a multisession backup:

```
#!/bin/sh  
#  
# /usr/doc/backuponcd/samples/simple_read  
# a very simple example how to read an archive stored on  
# several CDs. Must be killed by typing ^C if reading is  
# finished.  
#  
device="0,3,0"  
rawdevice=/dev/cdrom  
mountpt=/cdrom  
  
while true; do  
    cdrecord dev=$device -eject > /dev/null 2>&1  
    echo -e "\a\n Please insert the (next) volume" > /dev/tty  
    read ans  
    mount -t iso9660 -o ro,noexec $rawdevice $mountpt  
    if test $? -eq 0; then  
        echo "Now reading the volume..." > /dev/tty  
        cat /cdrom/*gz*  
    else  
        echo -e "\a\n Mount failed." > /dev/tty  
    fi  
    umount $mountpt  
done 2> /dev/tty < /dev/tty
```

The script is used as follows:

```
~/bin/simple_read | tar --compare --gzip --file=- | \\  
tee -a /tmp/logfile
```

BackupOnCd comes with longer shell scripts which recognize multi- and singlesession CD and type out better comments, know which volume is inserted and so on.

If you assume that burning does work similar you're right, it does (see 'Details' on page 7 if you are interested).

Chapter 2

Quick Reference

2.1 Installing BackupOnCd

Just extract the tarball. Copy the executables `/usr/local/sbin` and the folder `etc/backuponcd` to `/etc`. With the debian package you should have nothing to do.

2.1.1 Dependencies

On a debian system you need to have installed the following packages:

```
textutils
shellutils
fileutils
sed
awk
findutils
mount
cdrecord
mkisofs >> 1.8a
tar or
  afio
gzip
```

As you see everything is done with standard tools.

2.1.2 Customizing (a must)

- With the settings BackupOnCd comes with, you need 400 Mbyte free disk space under `/tmp/`. If free space doesn't fit you need to edit `'/etc/backuponcd/global.rc'` on page 11.

You have several options:

- Change the variable `TMP=" . . . "` to a directory where is enough disk space
- Enlarge the variable `tracks=` and decrease `tracksize=`
 Note, that a so-called lead-out is needed for each track, the amount of space on the disk is decreased by the count of tracks. For details have a look at `/usr/doc/cdrecord/README.multi` (<http://localhost/doc/cdrecord/README.multi>), or visit the homepage of `cdrecord` (<http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html>). You can play with these settings, if one track doesn't fit on the CD it is stored on the next CD.
- If free space is very small, you can force `BackupOnCd` to do the burning *not* parallel to the backup. If you did that, only one `trackspace` is needed. Edit `/sbin/backuponcd_toast.sh`, delete the single ampersand at the end of the line:

```
( nice -n -10 mkisofs -R -J -L $vol $volid \\  
$multiinfo $toast_dir | cdrecord -v $dev \\  
$devdepend $blank fs=8m -multi - && rm \\  
$toast_dir/* ) 2>> $logfile &
```

- A better option: Get an old 850 Mbyte disk and mount it to `/tmp`.
- Try out to change the variable `devdepend="speed=2 -eject"` to `devdepend="speed=x"`, while `speed` is the speed of your cdwriter. Please email me if it works or not. A Philipps CDD 3600 does not read the trackinfo if the CD is not ejected after each track. I think this toaster does a reset if the CD is ejected. You cannot run into trouble by playing this setting, a failed backup is all what can happen.
- Edit the line `archiver=`, choose `tar` or `afio`. Note, that `find` is *not* used if you choosed `tar` as archiver. The main disadvantage is, that no filelist is generated. `Tar` often is used as followed:

```
find . -not -type d | grep -v -f excludelist | \\  
tee -a filelist | tar -cvz --file=archive.tar.gz \\  
--files-from=-
```

With the above line empty directories and each directory found in `excludelist` (usually mountpoints) would *not* be archived. In my opinion this wouldn't be a backup, so `tar` is used as below:

```
tar -cvz -X $excludelist --file=- .
```

It should be easy for you to edit `/sbin/backupond` if you don't like this quality.

- Edit `rawdevice=` and `mountpt=`

2.2 Enjoying

See 'Usage' on page 1 for commandline options. Copy or move the file `/usr/doc/backuponcd/samples/exclu` on page 11 to a place where you want and edit it, so that it fits to your system or the system which you want to backup.

The remaining is fun. Everything is done automaticly, the backup will get compared, logfiles and scripts are burned on the last CD. It should be easy to do a restore with a system started from a floppy.

Chapter 3

Details

3.1 The Burning Process

I think the easiest way to learn about this is to have a look on the source.

If you don't understand how it works please don't email me. Learn more about sh, tar, baseutils and cdrecord.

Assume that tar is invoked as followed:

```
mkfifo fifo
tar -cpvz -X $excludelist --file=fifo * &
```

3.2 toast.sh \$params fifo

```
#!/bin/sh
#
#
if test "$1"; then
# > 0?
    source $1
else
    exit 1
fi

mkdir $toast_dir
let count=tracks

for i in {0,1,2,3,4,5,6,7,8,9}{0,1,2,3,4,5,6,7,8,9}; do
    dd of=$archive.$i bs=1k count=$(expr $tracksize \* 1024)
    wait
```

```

if ! test -s "$archive.$i" && ! test -e "$toast_dir/$sarch_name.*"; then
    # file existists but has length of null bytes: finished
    rm $archive.$i
    $jectcmd > /dev/null 2>&1
    break
else
    if test $(expr $i % $count) -eq 0; then
        if test $(expr $i) -gt 0; then
            $jectcmd > /dev/null 2>&1
            echo -e "\a\nPlease change the disk. Insert volume $(expr $i / $count
tinue." > /dev/tty
            read ans < /dev/tty
            fi
            let count=tracks
            if test -e "$toast_dir/$sarch_name.*"; then
                # Last CD 'lost' one Track
                let count-=1
            fi
            volid="-V $volume_name$(expr $i / $count + 1)"
            vol="-volset $(expr $i / $count + 1)"
            touch $toast_dir/"$volume_name$(expr $i / $count + 1)"
            blank="-blank=fast"
            multiinfo=""
        else
            volid=""
            vol=""
            blank=""
            let rec=1
            while test $rec -ne 0; do
                multiinfo="-C $(cdrecord $dev -msinfo) -M $rawdevice" 2>> $logfile
                rec=$?
            done
            fi

            mv $archive.$i $toast_dir
            if test $(expr $(ls -s $toast_dir/$sarch_name.$i | awk '{ print $1 }') /
lt $tracksize; then
                touch $toast_dir/$end_mark
            fi

            mking="mkisofs -r -J"
            tsize="$(($mking -q -print-size $toast_dir 2>&1 | tail -n 1 | awk '{prin
( nice -n -10 $mking -L $vol $volid $multiinfo $toast_dir | \
cdrecord -v $dev $devdepend $blank -multi tsize=$tsize - && rm $toast_d
file &
            fi

```

done

Chapter 4

Important Options, Global Settings

4.1 `/usr/doc/backuponcd/samples/excludelist`

This file must be named via the switch `-e`. Excludelist may be a empty file, but can't be missed.

```
# Example exclude list: will be adapted for using with afio
# (find | grep -v -f) by adding the leading dash and tar
# (adding a following asterisk).
#
amnt/
cdrom/
floppy/
initrd/
misc/
mnt/
proc/
tmp/
```

4.2 `/etc/backuponcd/global.rc`

This file is normally known by a built in parameter which can be overridden by the switch `-rc` (see 'Usage' on page 1).

```
#
# Exmaple rc-file for backuponcd. Don't rename or remove it.
#
#
# Most important parameter are TMP, tracks and tracksize.
# Read The Fine Manual if you don't guess what the
# signification is.
```

```
#

tracksize=200
tracks=3

# Change this to a directory where you have enough free disk
# space. You need 2 * $tracksize (Mbyte), see above.
#
# Notice that each track needs a space of $tracksize + ~12 MByte
# on the CD. So try out 'tracksize=94' 'tracks=6', it should
# work also. It doesn't matter when the last track doesn't fit.
# A 'lost' track is stored on the next CD.

TMP=/tmp

# No comment:

rawdevice=/dev/cdwriter
mountpt=/cdwriter

# Current possible: afio or tar.
archiver=afio
afioexcludes=".Z .z .gz .arc .gif .zip .zoo .lha .tpz .taz .tgz .tazg .deb ."

# This is a part of the volumename of the CD
# Use only valid chars of filenames, no spaces.

dateformat="%Y-%m-%d"

#          --- Parameter of cdrecord ---
#
# Type 'man cdrecord' if you didn't hear about the following
# parameter:
dev="dev=0,4,0"

# '-eject' depends on the philipps CDD 3600 isn't able to
# read trackinfo without reset. Try out devdepend="speed=x"
# with any other cdwriter.
devdepend="speed=2 fs=8m -eject"

# Minimizing package dependencies, could also be
# "cdeject -d $rawdevice":
ejectcmd="cdrecord $dev -eject"
loadcmd="cdrecord $dev -load"

#-----
```

```
# You shouldn't need to change anything below this line
# The next two items must be found in $PATH:

burn_it=backuponcd_toast.sh
verify=backuponcd_fifo-in.sh

# Internal stuff:

arch_dir=$TMP/backup.$$
log_dir=$arch_dir/log
script_dir=$arch_dir/scripts
toast_dir=$arch_dir/burn
variable=$script_dir/variable.sh
notcomp=$log_dir/compress-exclude
filelist=$log_dir/filelist
logfile=$log_dir/logfile
archive=$arch_dir/$arch_name
```

Note: Change the internal stuff if you don't like the filenames *and* if you know what you are doing.

Chapter 5

Download / Contributes

Please email me if you have ideas or bug reports. I would enjoy if you had fixes. Feel free to contribute

BackupOnCd 0.9.1 download ([./backuponcd.tar.gz](#))