Faig els backups amb luckybackup, per clonar discs, o carpetes entre discs de treball i els discs de backup.

També empro Backintime per al Correu-e, per exemple, per tenir copies exactes en determinats moments del temps.

**Introducció a Backintime i LuckyBackup**

A la xerrada al CCívic Can Basté (5 Abril 2014):

---

**1.1.1. Backintime**

Open the program with "Menu > System > Back in Time", select the appropriate profile, go to properties, and ensure that the path to store the snapshots exists (disks mounted, etc), and it doesn't contain the ending "backintime" folder, since this will be added by the program later on at run time. See the screenshot:

![Click to expand](image1)

Select also the directories (or whole units, in /media) that you want to back up. In this case, the sd card from the smartphone is selected. (media/xavi/CDF2-6BE2):

![Click to expand](image2)

---

**1.1.2. Luckybackup**

Iniciat el programa a Coprinus a "Menu > Sistema > Luckybackup (superusuari)"

(preneu nota que obrir el programa des d'una consola amb sudo luckybackup no és el mateix, i no carrega els perfils de configuració de l'usuari root, preparat per al clonat de discs de casa)

![Click to expand](image3)

---

Còpia dels arxius de configuració a data 14 Octubre 2013 a Coprinus:

Luckybackup Coprinus 131014
1.2. Redo Backup and Recovery

http://redobackup.org/[1]

GNU/Linux LiveCD with a complete set of tools used to clone hardisk of whole computers, with their partitions, operating systems, and only taking the space occupied with real data, and not the full size of the cloned harddisk.

Currently, tool used to clone PenguinBookPro (tri-boot: MacBookPro laptop with GNU/Linux, MacOSX and Windows in it).

1.2.1. Initial work

Insert Live CD and boot with it. You will see an interface like this one, where you will click at "Backup":

In case the program is closed for some reason, you can re-open it again using the menu in the bar at the bottom of the window:

Menú detail:

Then the Backup process starts to get configured. It will ask you a few questions in a few progressive steps.

1.2.2. Step 1: Select the Source Drive

1.2.3. Step 2: Select Partitions to Save

1.2.4. Step 3: Select Destination Drive

1.2.5. Step 4: Select Destination Folder

Here you can select the destination folder, or create a new one for you in your desired location in the chosen drive.

The selection must be done in the /mnt/backup/ base folder, which seems symlinked to the real destination drive and folder. If needed, you will be prompted to choose your destination folder within the chosen drive within the /mnt/backup base path.

Once everything is fine, you will see listed the destination folder in the appropriate field of the step 4 of the backup wizard.

1.2.6. Step 5: Name your backup
1.2.7. Step 6: Creating the Backup image

Once finished, you will be shown the total time invested to create your backup.

1.2.8. Checking sizes

For curiosity, you can check what is the size of that folder where you stored your backup. 53 Gb in our case.

Then, you can also have a look at the used space in the disk partitions in the laptop:

Suming them all, you get more than 70Gb as used space, and nearly 150Gb of all space in the drive (/dev/sda). So it seems that Redo Backup and Recovery did quite a good job taking not only the used space to clone your disks/drives, but also taking less space than the one in the real source drive (more than 20Gb less space).

Nice job!

1.3. Custom shell script

In some cases (mainly at work, but also in some servers and computers at home), I use some custom backup script which I have been developing and tweaking over the years. Feel free to re-use it and extend it for your own needs, under the terms of the GNU/GPL or GNU/LGPL copyright licenses.

See this example:
http://seeds4c.org/Pentinella+with+Ubuntu+14.04

Computer at work

Manual local Backups

```
#!/bin/bash

#PARAMETERS TO CUSTOMIZE THE SCRIPT
MLABEL="xavi_pentinella"
MUSER="mysqluser"
MPASS="mysqlpass"
MHOST="localhost"
FTPU="ftpuser"
FTPP="ftppass"
FTPS="ftpserver"
FTPF="./backups/pentinella"
NOWD=$(date +%Y-%m-%d)
NOWT=$(date +%H_%M_%S)

# Some paths defined
MYSQL="$(which mysql)"
MYSQLDUMP="$(which mysqldump)"
BAKPATH="/home/xavi/backups.locals" # TODO: make the following paths relative to this one
BAK="backup_local_pentinella"
TIKIFILESABSPATH="/var/www/tiki_files"
RBAK1="mysql"
RBAK2="tikifiles"
RBAK3="serverfiles"
RBAK4="homexavifiles"
```
EMAILF="ueb@vhir.org"
EMAILT="xavier.depedro@vhir.org"
SMTP="servirmtal.ir.vhebron.net"

### End of parameters

# Base path for backup folders
BBAK=$BAKPATH/$BAK/$NOWD
# Absolute paths to backup folders (base path + relative path)
ABAK1=$BBAK/$RBAK1
ABAK2=$BBAK/$RBAK2
ABAK3=$BBAK/$RBAK3
ABAK4=$BBAK/$RBAK4
# Other variables used
GZIP="$(which gzip)"
# Relative paths for each log file
RLOGF=log-$MLABEL-SUM.$NOWD.txt
RLOGF1=log-$MLABEL-$RBAK1.$NOWD.txt
RLOGF2=log-$MLABEL-$RBAK2.$NOWD.txt
RLOGF3=log-$MLABEL-$RBAK3.$NOWD.txt
RLOGF4=log-$MLABEL-$RBAK4.$NOWD.txt
# Base log path (set by default to the same base path for backups)
BLOGF=$BBAK
# Absolute path for log files
ALOGF=$BLOGF/$RLOGF
ALOGF1=$BLOGF/$RLOGF1
ALOGF2=$BLOGF/$RLOGF2
ALOGF3=$BLOGF/$RLOGF3
ALOGF4=$BLOGF/$RLOGF4

### These next parts (1) & (2) are related to the removal of previous files in these folders if they exist, and create dirs as needed for new set of periodic backups ###

## (1) To remove all previous backups locally at the server and at the same base backup folder, uncomment the following line

```
[ ! -d $BAKPATH/$BAK ] && mkdir -p $BAKPATH/$BAK || /bin/rm -f $BAKPATH/$BAK/*
```

## (2) To avoid removing previous backups from teh same day locally, keep the last part commeted out (with ## just in front of "|| /bin/rm -f ...")

```
[ ! -d $ABAK1 ] && mkdir -p $ABAK1 || /bin/rm -f $ABAK1/*
[ ! -d $ABAK2 ] && mkdir -p $ABAK2 || /bin/rm -f $ABAK2/*
[ ! -d $ABAK3 ] && mkdir -p $ABAK3 || /bin/rm -f $ABAK3/*
[ ! -d $ABAK4 ] && mkdir -p $ABAK4 || /bin/rm -f $ABAK4/*
```

### $MYSQL="$(MYSQL -u $MUSER -h $MHOST -p$MPASS -Bse 'show databases')"
for db in $DBS
do
FILE=$ABAK1/$db.$NOWD-$NOWT.gz
$MYSQLDUMP -u $MUSER -h $MHOST -p$MPASS $db | $GZIP -9 > $FILE
done

#### Backup tikifiles####
```
tar -czvf $ABAK2/00-$RBAK2-$MLABEL.$NOWD-$NOWT.tgz $TIKIFILESABSPATH/* > $ALOGF2
```

#### Backup serverfiles####
```
```

#### Backup home user files####
```
tar -czhv $ABAK4/00-$RBAK4-$MLABEL.$NOWD-$NOWT.tgz /home/xavi/scripts/* /home/xavi/.local/* /home/xavi/.config/* /home/xavi/.Skype/* /home/xavi/.luckyBackup/* /home/xavi/.ssh/* /home/xavi/.purple/* > $ALOGF4
```
### Send files over ftp ###

```bash
#lftp -u FTPU,FTPP -e "mkdir $FTPF/$NOWD; cd $FTPF/$NOWD; mput $ABAK1/*.gz; mput $ABAK2/*.tgz; mput $ABAK3/*.tgz; quit" $FTPS > $ALOGF
```

cd $ABAK1; ls -lh * > $ALOGF1

Add a short summary with partial dir sizes and append all partial log files into one ($ALOGF)

cd $BBAK; du -h $RBAK1 $RBAK2 $RBAK3 $RBAK4 > $ALOGF; echo "--- $RBAK2 uncompressed: -----------" >> $ALOGF; du $TIKIFILESABSPATH -h --max-depth=2 >> $ALOGF

### Compress and Send log files ###

```bash
tar -czvf $ALOGF1.tgz -C $BLOGF $RLOGF1
```

```bash
#tar -czvf $ALOGF2.tgz -C $BLOGF $RLOGF2
tar -czvf $ALOGF3.tgz -C $BLOGF $RLOGF3
tar -czvf $ALOGF4.tgz -C $BLOGF $RLOGF4
```

```bash
#lftp -u FTPU,FTPP -e "cd $FTPF/$NOWD; put $ALOGF1.tgz; put $ALOGF2.tgz; put $ALOGF3.tgz; put $ALOGF4.tgz; quit" $FTPS
```

### Send report through email ###

```bash
sendemail -f $EMAILF -t $EMAILT -u '[Pentinella Manual Local Backup Report]' -m 'Short report attached' -a $ALOGF -a $ALOGF1 -s $SMTP -o tls=no
```

---

### Crontab ###

Crontab for root as of Dec 29th, 2015

```bash
#Back In Time system entry, this will be edited by the gui:
0 14 * * 1 /usr/bin/nice -n 19 /usr/bin/ionice -c2 -n7 /usr/bin/backintime --backup-job >/dev/null 2>&1
```

```bash
#Back In Time system entry, this will be edited by the gui:
0 14 * * * /usr/bin/nice -n 19 /usr/bin/ionice -c2 -n7 /usr/bin/backintime --profile-id 2 --backup-job
```

```bash
30 13 * * Wed sh '/home/xavi/scripts/backup_manual_pentinella.sh'
```

```bash
# Sending "Daily reports" of watch-emails
* * * * * cd /var/www/html/12.x/;php console.php daily-report:send > /dev/null 2>&1
```

Alias names for this page: Pentinella

## 1.4. How to stop Backintime backups in progress

If for some reason you need to stop an ongoing backup job made with Backintime, you can do so in two steps:

1) Ensure that there is no process related to "backintime" in the jobs list. In case there is some, kill it:

```bash
Command in a console at cep
ps auxfww|grep backintime|awk '{print $2 }' |xargs sudo kill -9
```

2) Delete the temporal folder and files where backintime stores the information of the ongoing back-up process, and restart backintime. Don't worry, the configuration files are elsewhere (/home/username/.config/backintime or /root/.config/backintime, for a standard user or the root, respectively). Then restart Back In Time.

```bash
Folders to be deleted their contents
# For a standard user:
sudo rm /home/username/.local/share/backintime/* -R
# For the root user:
sudo rm /root/.local/share/backintime/* -R
```

Noms alias d'aquesta pàgina:
Seeds For Change - seeds4c.org

Backup | Backintime | LuckyBackup | rsync