Backup with tar and rsync

CIS 2235 Linux System Administration

Backup strategies

Several methods to back up data

1. Apps - 3rd party apps as described in last class
2. `dump / restore`
   - "Levels" for diff / inc
   - Entire partition (if you want levels)
3. `tar` command
4. `rsync` command

Backup strategies

Tar or rsync compared to dump/restore:

Pros:

- Easy to script
- Not limited to a partition. Any user can use it for their own custom dirs.

Cons:

- These commands don't have “levels” built-in like dump.
- So we'll need to be smarter.

Archiving Files with tar

- For historical reasons, it stands for "tape archive."
- Encapsulates many files in a single file
  - Known as a tar archive or a tarball
- Used for both distribution and backup
- Has unusual command-line option syntax
  - Common options are given as single letters
  - Yet no hyphen is needed (optional)

Archiving Files with tar

Format:

```
tar [-]action [options] path(s)
```

tar must be given exactly one action option

- Indicates which operation to perform
- 3 main actions (*required*): create (c), list (t), extract (x)

Creating Archives with tar

Use the c option to create an archive

```
$ tar czf docs.tar.gz ~/documents
```

- f specifies the archive's filename
  - Must be followed directly by the filename (therefore, usually last)
  - Common to use .tar extension (Does Unix require it?)
- The z option compresses the archive with gzip
  - .tar.gz extension is used to indicate compression
  - .tgz extension is also popular
- The other arguments to tar are a list of files and directories to archive

Listing the Files in tar Archives

To check that a tar file has been made correctly, use the t operation (for 'list'):

```
$ tar tzf docs.tar.gz
```

The z and f options work as before.

z isn't required — tar realizes the file is compressed

To show more information about files, add the v (for 'verbose') option

Shows information similar to ls -l

Extracting Files from tar Archives

Use the x operation to extract files from an archive:

```
$ tar xvf docs.tar.gz
```

The v option lists the files as they are extracted (verbose)

Extracting Files from tar Archives

To extract individual files, list them as arguments

```
$ tar xvf docs.tar.gz documents/phone-numbers.txt
```

Other useful options:

k (--keep-old-files) will not overwrite any existing files, only extracting missing ones

p (--preserve-permissions) will set extracted files to have the permissions they had when archived

Moving entire directories

Can use tar to move file hierarchies

```
$ tar -cf - -C srcdir . | tar xpc - -C destdir
```

or more traditionally:

```
$ cd srcdir; tar -cf - . | (cd destdir;  tar -xpf -)
```

Does this remind you of moving /home?

Using tar in a backup strategy

Recall that dump only works on an entire partition

What if we want to back up part of a directory tree?

In cron for daily or hourly (or anacron):

```
$ tar cvzf /data/mydocs.tgz ~/Documents
```

However, this is a "full" backup (every time). Not *exactly* the most efficient for daily backups.

Incremental backups with tar

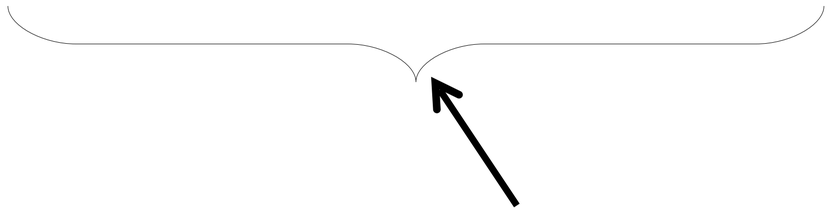
2 methods with `find`

Use `find` and `-mtime` option:

```
$ tar c `find /home -mtime -1 ! -name \*.o  \
  -type f -print`
```

Use find and --newer option:

```
$ tar czvf /backup/home.1.tar.gz `find /home \
  --newer /backup/home_full.tar.gz -print`
```



List files changed “newer” than the last full backup.

Another format

Some newer versions of tar (Linux GNU tar) have it built-in

—after-date

```
$ tar cvf test2.tar --after-date "`date -r test.tar`"
tarTesting/
```

- However, using —after-date for incremental has a major issue
  - It doesn't correctly detect some changes
  - Renaming directories and files does not change the modification date/ time

Gnu tar snapshots

A gnu snapshot file keeps track of what has previously been backed up

```
-g, --listed-incremental FILE
      handle new GNU-format incremental backup

-G, --incremental
      handle old GNU-format incremental backup
```

It will be read-in and used as the time-date stamp for previously archived files.

The INC (or DIFF) archive will be only files changed after this SNAPSHOT file.

Gnu tar snapshots

The SNAPSHOT file will change → it will be written/updated with the current changes.

If no snapshot file exists, then a new one will be created → this is a FULL backup

Gnu documentation calls these "SNAR" files (snapshot archive)

Using gnu tar snapshot files

creates FULL backup

```
$ tar --create \
          --file=archive.1.tar \
          --listed-incremental=/var/log/usr.snar \
          /usr
```

since /var/log/usr.snar didn't exist

The next incremental uses a copy of the original SNAR (which will be updated), and archives to an incremental tarball

```
$ cp /var/log/usr.snar /var/log/usr.snar-1
$ tar --create \
          --file=archive.2.tar \
          --listed-incremental=/var/log/usr.snar-1 \
          /usr
```

The /var/log/usr.snar.1 is *updated*

Other backup notes

Some areas which shouldn't be backed up are:

- `/tmp` — usually doesn't contain anything of lasting value
- `/proc` — automatically generated by the kernel
- `/sys` — automatically generated by the kernel
- `/dev` — if using devfs this is also generated automatically
- `/mnt` — media mounted here, like CD ROMS

Filesystems mounted remotely whose backup is taken care of elsewhere

rsync

synchronizes files and directories from one location to another while *minimizing* data transfer by only copying changed files

'mirrors' a directory – used for distribution or backup

only copies "what's changed" (*i.e.*, a delta)

i.e., don't have to use 'find -newer' like we did with tar

rsync

Simple format:

```
rsync [option] source dest
```

-a option is “archive” which is recursive and preserves attributes

```
$ rsync -va ~/Documents/ /data/rsync/documents
```

source dest

rsync and ssh

Can also use ssh-like "`user@host:file`" for source or dest

source dest

```
$ rsync -va ~/Documents ldamon@diffMachine.com:/data/
                        rsync/documents
```

rsync for backups

If you use `rsync` to backup, then you always have current, synced, mirrored up-to-date copy.

One copy “only”.

This one copy is updated not with a 'full' copy, but always as an 'incremental' copy. That’s good...

rsync snapshots

Sometimes want to keep older versions — "snapshots"

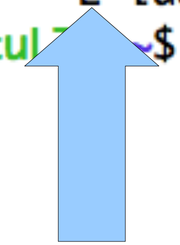
We can use the `–link-dest=DIR` option to only copy new files (inc) and hard-link old files to a previous snapshot dir

```
$ rsync -va –link-dest=/data/rsync/documents.1   ~/Documents/   rsyncTest
```

prev snapshot source dest

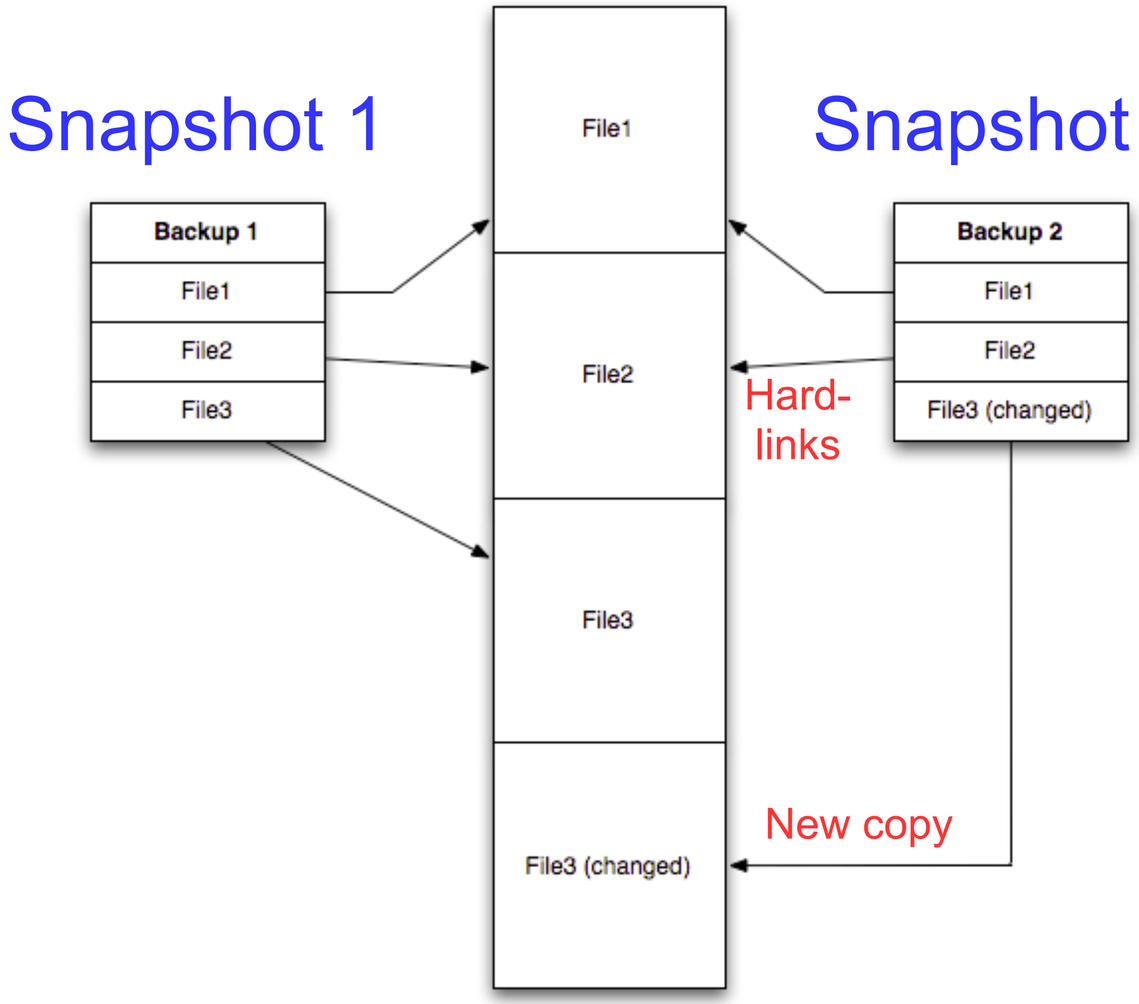
Snapshots

```
ldamon@ubuntuLTS:~$ ls -li rsyncTest
total 0
323 -rw-r--r-- 2 ldamon svn 0 Apr  9 14:18 file1
324 -rw-r--r-- 2 ldamon svn 0 Apr  9 14:19 file2
325 -rw-r--r-- 2 ldamon svn 0 Apr  9 15:07 file3
326 -rw-r--r-- 2 ldamon svn 0 Apr  9 15:09 file4
327 -rw-r--r-- 2 ldamon svn 0 Apr  9 14:18 file5
328 -rw-r--r-- 2 ldamon svn 0 Apr  9 14:18 file6
332 -rw-r--r-- 2 ldamon svn 0 Apr  9 15:16 file7
330 -rw-r--r-- 1 ldamon svn 0 Apr  9 15:09 file8
329 -rw-r--r-- 2 ldamon svn 0 Apr  9 15:16 file9
ldamon@ubuntuL~$
```



Which files are “new” (part of inc copy) and which are “old” (already in early rsync)?

Hard links and snapshots



Simple script for rsync snapshots

```
#!/bin/sh

date=`date "+%Y-%m-%dT%H:%M:%S"`
rsync -aP --link-dest=$HOME/Backups/current /path/to/important_files $HOME/Backups/back-$date
rm -f $HOME/Backups/current
ln -s back-$date $HOME/Backups/current
```

Consistency

What if a file changes during backup?

- tar – not so bad. Just that 1 file corrupt
- dump – pretty bad → the entire backup is corrupt! Ack!

Solution:

- Telinit 1, (single root user)
- Then re-mount as ro (read-only)

Read-only partition - to avoid inconsistent backup

You need to mount partition as read only, so that no one can make changes to file and make a backup:

```
# umount /home
# mount -o ro /home
# tar -cvf /dev/st0 /home
# umount /home
# mount -o rw /home
```

Other backup commands (FYI)

`cpio` — alternative archiving program

`afio` — similar to cpio, but allows files to be compressed individually

Compressed archives can be made which are resilient to corruption

`pax` - a program which reads/writes both tar and cpio formats

`dd` - input/output stream which copies and also preserves attributes, owners, mtimes, etc.

In summary

Dump – whole partition, 1 file, -<level> make incremental easy

Tar - 1 file, compression, not partition only, any dirs or files, deal with incremental using SNARs

Rsync - makes a mirror, only copies what's changed by default, can use links to make snapshots, uses user@host:file for ssh to a remote box