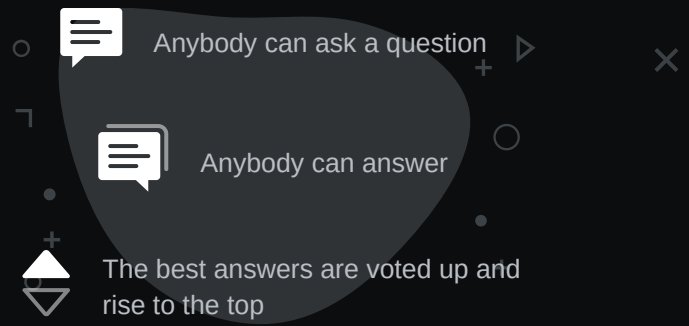


Super User is a question and answer site for computer enthusiasts and power users. It only takes a minute to sign up.

Sign up to join this community



Indexed and seekable tar files

Asked 8 years, 7 months ago Modified 1 year, 7 months ago Viewed 4k times



When backing up large amounts of files, I've been taring them so that I only need to move one file around. But listing the content of tar files and extracting specific files out of them is really slow.

11



Is there a better alternative for this? Some way to tell tar to build an index and be seekable, or another archive format altogether?



linux backup archiving



Share Improve this question Follow

asked Mar 5, 2015 at 21:59



user831885

111 ● 1 ● 4

3 Answers

Sorted by: Highest score (default) ◆



Yes, there is a project (now a bit old) called [tarindexer](#) that can do this. You will need to create an index file on your `tar` file but after it's created, you can then do random seeks.

4



Here is an example usage:

```
tarindexer -i tarfile.tar tarfile.tar.idx
tarindexer -l tarfile.tar tarfile.tar.idx mydir/myfile > myfile
```



The tar index file is itself a simple text file with byte offsets and lengths of the files in the archive. For example:

```
$ cat tarfile.tar.idx
mydir 512 0
mydir/myfile1.txt 1024 51
mydir/myfile0.txt 2048 7
mydir/myfile 3072 15
```



abetusk

141 ● 3

Seems like a nice idea, that extends the format capabilities w/o actually changing the format itself. It's a pity the project didn't widespread and become mainstream. – Denis Kulagin Mar 21, 2022 at 11:55

about the practicality of this. It's in python, so is it really fast enough for places where it really matters (huge, hundreds of gigabyte, archives), and does it handle what we have in practice - compressed tar archives? – creanion Jul 14 at 9:44 ✎

- 1 @creanion The source for tarindexer is 125 lines of code, including headers, so you can see what it's doing. The speed of extraction and archiving is passed on through the Python 'tarfile' library which, I assume, optimizes for execution speed. Large files only means the overhead of Python and parsing the index file become more negligible. In terms of compressed tar archives, one can use a combination of block gzip (with its own index) and this method to do efficient lookups. So, no this tool doesn't do what you ask but that's not what this question asked for in the first place. – abetusk Jul 14 at 21:03

@abetusk I see now that it could be layered, so it's a great stepping stone. I'll be honest and I think it depends on how you read the question. In my world, tar files used for backup are compressed, so that would be part of the question. But it's not explicitly stated, so it's unclear. – creanion Jul 15 at 8:45



1



I've been struggling with the same problem. Large tarballs become pretty much inconvenient for storing large backups that you only need partial access to (e.g. to extract particular files or directories).

Nice solution that I found is called [SquashFS](#). *SquashFS* is a compressed read-only file system for Linux that can be used to create browsable and portable backups as well. To access and browse contents partially one just needs to mount SquashFS image and browse it like an ordinary filesystem.

Squashfs is intended for general read-only filesystem use, for archival use (i.e. in cases where a .tar.gz file may be used), and in constrained block device/memory systems (e.g. embedded systems) where low overhead is needed.

From here: <https://github.com/plougher/squashfs-tools/blob/master/USAGE>

What is also nice about *SquashFS* is that it's a modern solution with support for multithreading. It shows quite an impressive performance when creating an image compared to targz-ing single-threaded.

Share Improve this answer Follow

answered Mar 21, 2022 at 16:54



Denis Kulagin

173 ● 2 ● 9

You do need root privileges to mount the squashfs file system. Not particularly suited for users backing up their own files or creating archives for distribution. – doneal24 Mar 21, 2022 at 17:16

@doneal24 It supports *unsquashing* (full or partial) and browsing an image file w/o root privileges as well. So there is an option for such a case too. – Denis Kulagin Mar 21, 2022 at 17:52 ✎




0

Options `-n`, `--seek` assumes the archive is seekable, `--no-seek` assume the archive is not seekable, but it works only when reading (listing or extracting). Tar tries to determine this automatically.

So, try to use `-n` when listing/extracting, for example `tar -ntf archive.tar.bz2`.



Does this also apply when creating the archive? – [user831885](#) Jan 6, 2016 at 20:20

-
- 2 No, only when listing/extracting the archive. Tar is stream archiver. If you want to create a listing, you could use 'v' option, e.g. 'tar cvvjf /tmp/foo.tar.bz2 /path/to/files > /tmp/list.txt' and search through this file later. Look here, it could be useful (serverfault.com/questions/59795/...) – Konstantin Morenko Jan 7, 2016 at 6:22 
-