File Archiving and Compression Commands

Archiving and compression

Archives:

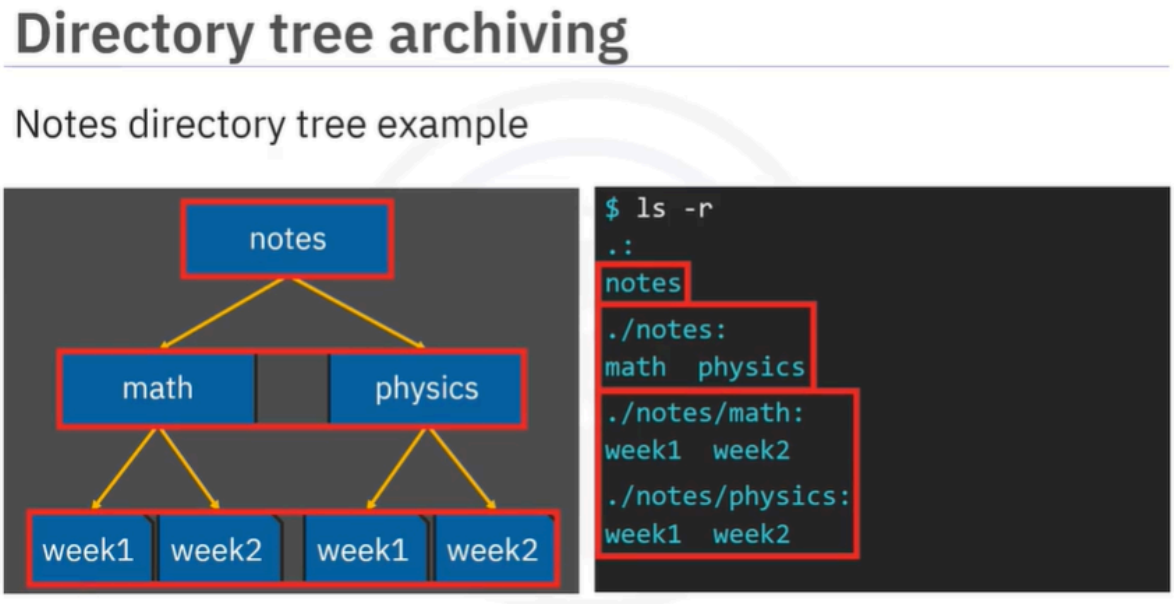
- store rarely used information and preserve it
- are a collection of data files and directories stored as a single file
- make the collection more portable and serve as a backup in case of loss or corruption

Archiving and compression

File compression:

- reduces file size by reducing information redundancy
- preserves storage space, speeds up data transfer, and reduces bandwidth load

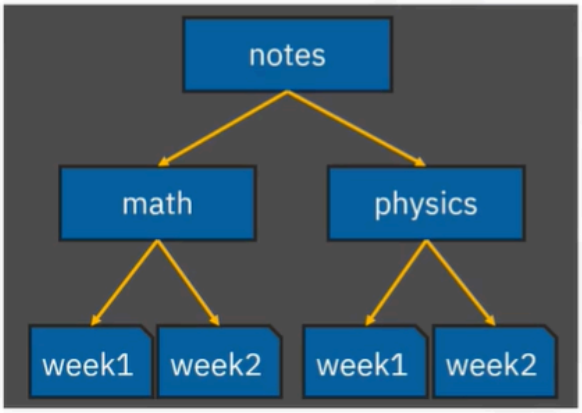
- Archiving and compression are distinct processes, which are usually combined. Archiving is the process of storing information that you don't use regularly but want to preserve. An "archive file" is a collection of data files and directories that are stored as a single file. Archiving makes the collection more portable and serves as a backup in case of loss or corruption.
- File compression, on the other hand, involves reducing the size of a file by taking advantage of redundancy in its information content. The main advantages of compression include preserving storage space, speeding up file transfers, and reducing bandwidth loads.



- Now, suppose you have created a "notes" directory for keeping track of your course materials. You decide it would be a good idea to archive your notes in case you need them in the future. Your "notes" directory tree has the following structure: It contains two subfolders, called "math" and "physics," each of which contains files called by the same names, "week 1" and "week 2." Using the LS command with the –R option, you can recursively list all the directories and files in your current directory tree. You can see the correspondence with the graphical representation of the tree, starting with the parent "notes" directory, the math and physics subdirectories, and the week one and week two files within the math and physics folders.

File archiving and compression

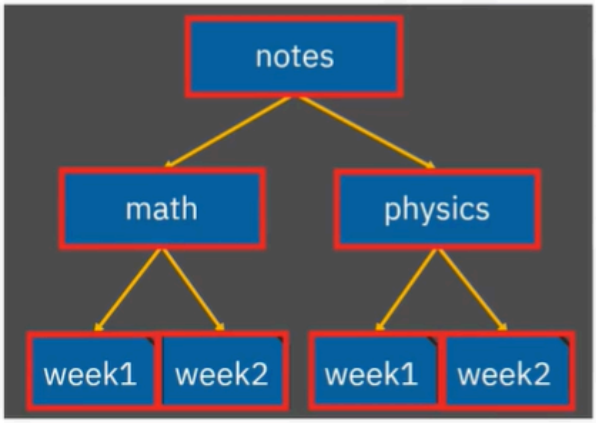
`tar` (tape archiver) – Archive and extract files




```
$ tar -cf notes.tar notes
$ ls
notes notes.tar
$ tar -czf notes.tar.gz notes
$ ls
notes notes.tar notes.tar.gz
```

Checking your archive contents

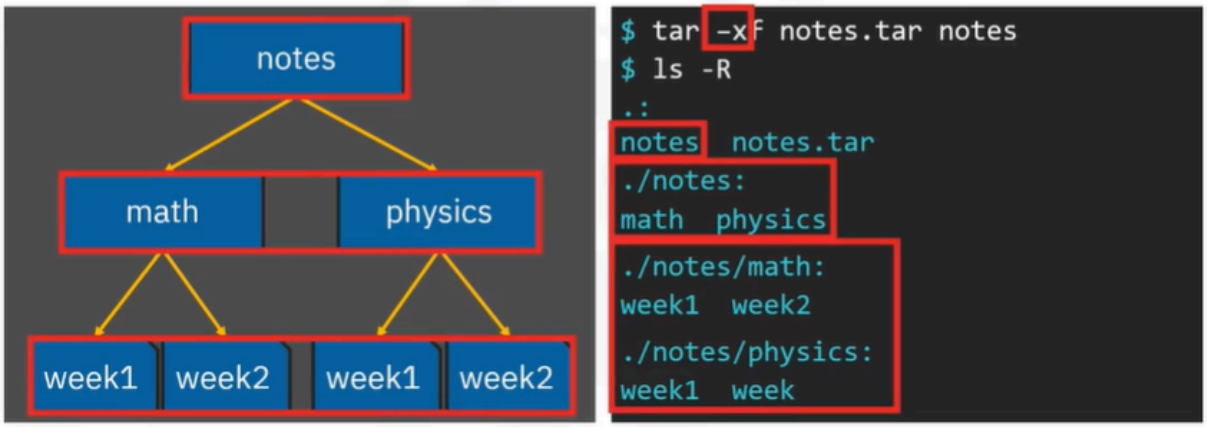
`tar` – List archive contents




```
$ tar -tf notes.tar
notes/
notes/math/
notes/physics/
notes/physics/week1
notes/physics/week2
notes/math/week1
notes/math/week2
```

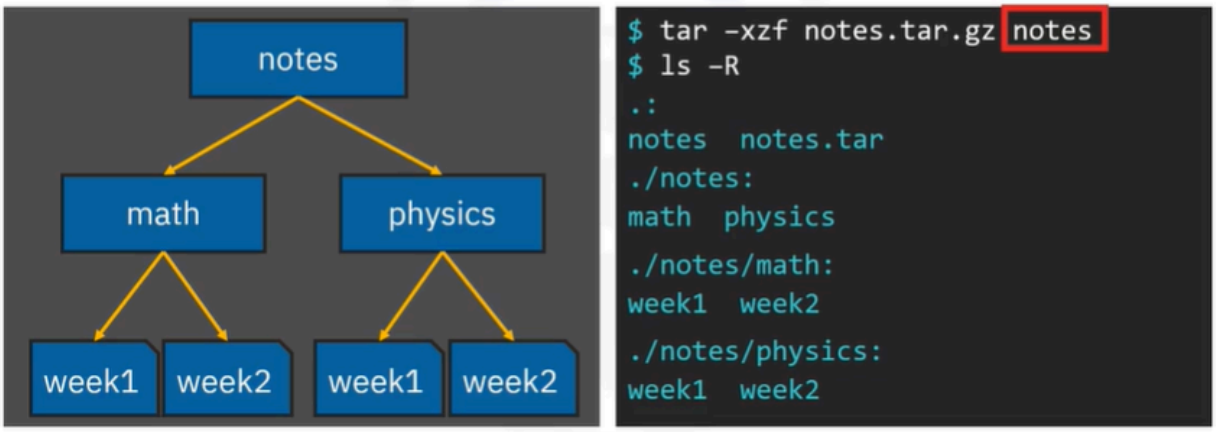
Extracting archived files

`tar` – Extract files and folders



Decompressing and extracting archives

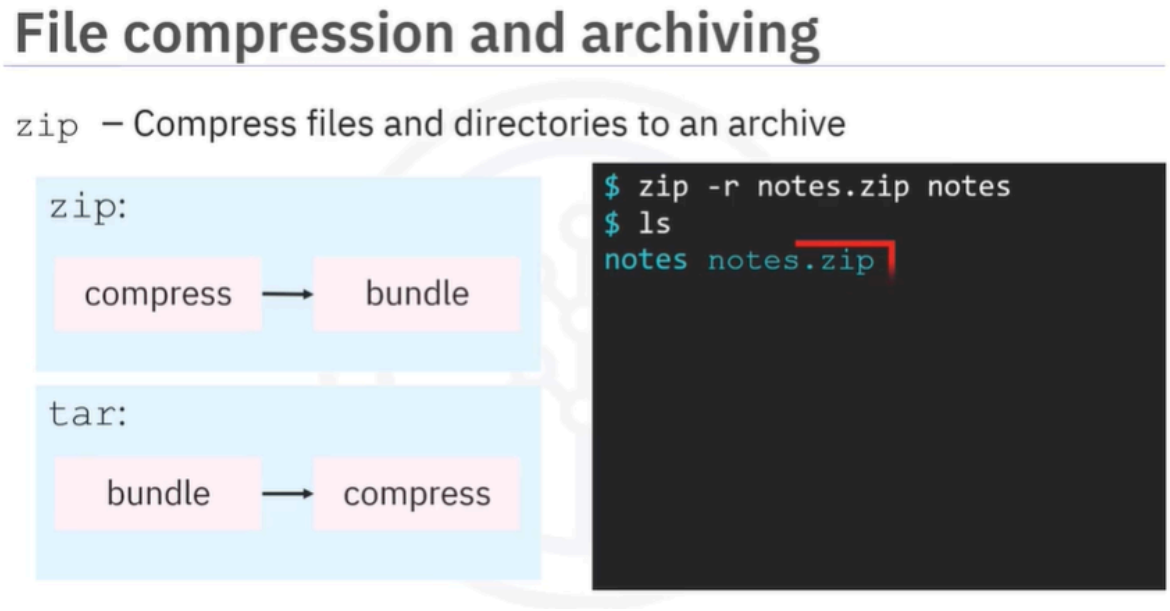
`tar` – Decompress and extract



- You can use the “tar,” or “tape archiver” command to archive and de-archive files and directories. A popular term for an archived tar file is a ”tar ball.” To archive your entire notes directory, including its subdirectories and all files within, enter the command: “tar minus c f,” a name for the archived file, such as “notes dot tar,” followed by the file or directory you wish to archive, which is “notes.” The “c” option means ”create a new archive,” and the “f” flag tells

tar to interpret its input from the file rather than from the default, which is standard input.

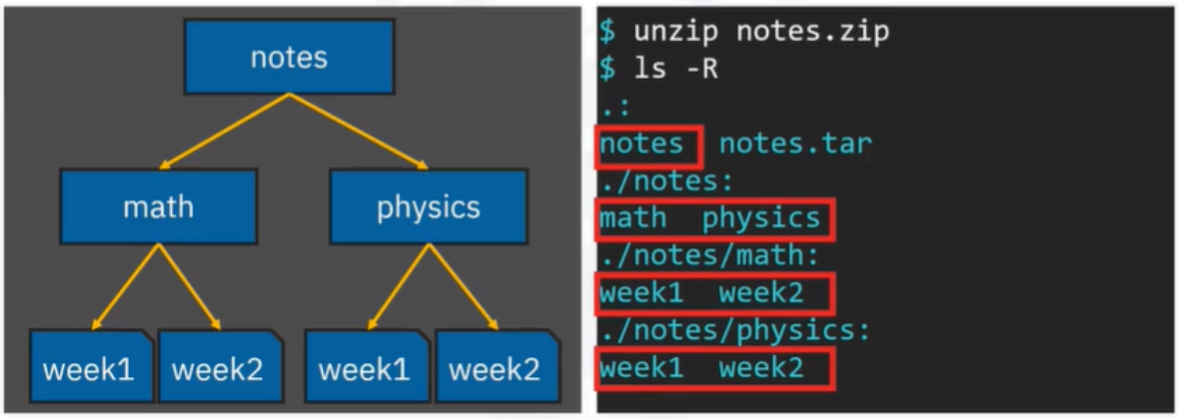
- Entering “L S” shows that your current directory now contains the original notes folder as well as the notes.tar archive file. If you would also like your archive to be compressed, you can enter the same command, except now you include the “-z” option, which filters the archive file through a GNU (pr. “geh-noo”) compression program called g-zip. Adding the suffix “dot g z” to the output name, ensures that Windows-based programs, for example, will correctly recognize the file type. Entering “L S” now shows the compressed “notes dot tar dot gz” file that you created.
- You can check the contents of your archived notes file by calling tar on your "notes" tar ball with the “minus T,” or "list" option. This lists all the files and directories in your tar ball. And as expected, it has the same structure as your original notes folder, with "notes" as the parent directory, "math" and "physics" as subdirectories, and the week 1 and week 2 files at the terminal nodes. You can also unpack, or “de-archive” your archived files using the tar command. You can enter "tar minus x f" followed by the archive file name, "notes dot tar", plus an optional destination name, for example, "notes," which happens to be the default.
- The "minus x“ option tells tar to extract file and directory objects from the archive. Now if you enter "L S minus R," you can see that the archived notes folder has been de-archived into a parent folder called notes. subfolders called “math” and “physics,” and the four week one and two files you started with. This verifies that the original structure of your "notes" directory is intact. Similarly, you can decompress a "tar dot gz" file and extract the files from it. To unpack and decompress the notes.tar.gz file, you can enter “tar minus x z f” followed by the compressed archive file name, “notes tar dot g z,” and an optional destination folder, “notes.” Again, by entering “L S minus R,” you can see the directories and files have been unpacked as expected.



- You can use the “zip” command to compress files and directories and package them into a single archive. Notice the order of operations that zip implements. Zip compresses files prior to bundling them, Whereas tar, with the “minus z option,” achieves compression by applying “g zip” on the entire tarball, but only after bundling it. To compress your “notes” directory and package it to a “zip” file, enter “zip notes dot zip notes.” And after entering “L S,” you can see that the notes.zip archive has been created.

Extracting and decompressing archives

`unzip` – Extract and decompress zipped archive



- The “unzip” command, as you might guess, extracts compressed files from a zip archive and decompresses them. To unzip your notes.zip file, simply enter “unzip notes dot zip”. After entering “L S minus R,” You can see that unzip has created your notes folder, and unpacked your directories and your “week one and two” files, as expected.

Recap

In this video, you learned that:

- Compression preserves storage space, speeds data transfer, and reduces system load
- `zip` compresses files and folders prior to archiving them
- `tar` archives files and directories into a tarball and can also compress it
- `unzip` unpacks, and decompresses a zipped archive
- `tar` decompresses and unpacks a `tar.gz` archive