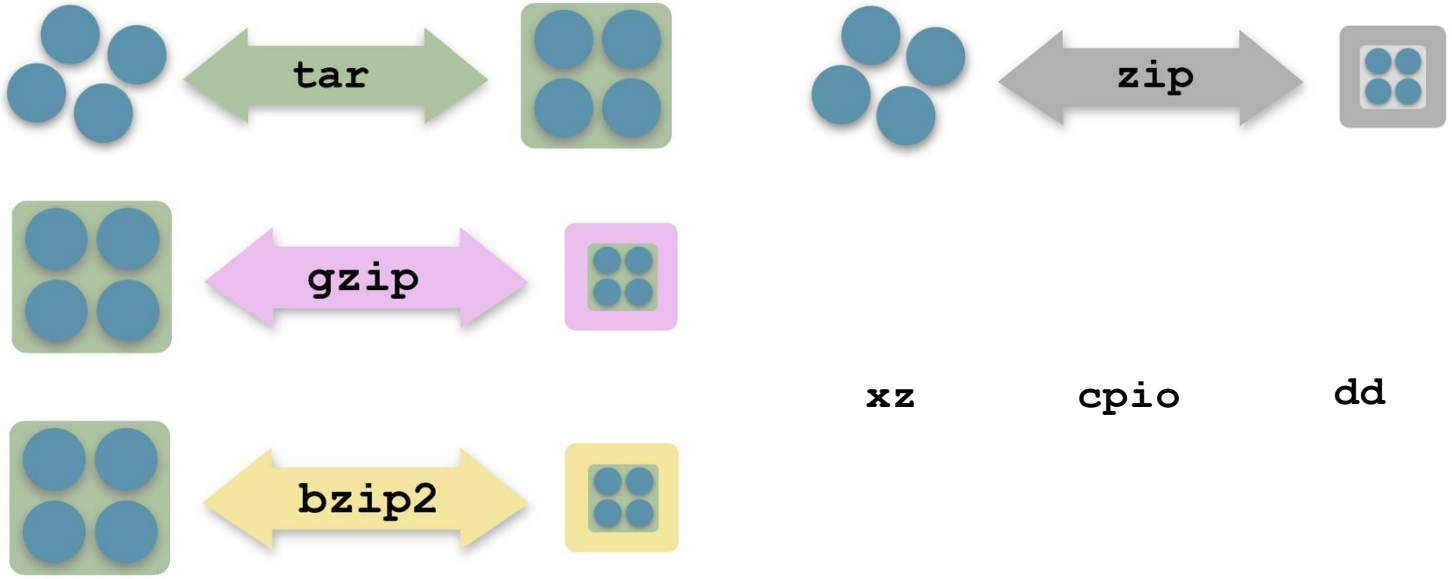
Chapter 12: Archive Commands

Introduction to Archiving

- Linux provides several utilities for compressing and archiving data. Each one has its' own unique advantages and disadvantages.



The tar Command

- The `tar` (tape archive) command merges multiple directories and files into a single file
- Does not compress files by itself
- Supports `gzip` (with `-z` option) and `bzip2` (with `-j` option) compression
- Customary to add ".tar" extension

The tar Command

- Three basic functions:
  - c (create or combine) – creates a tar file
  - t (table of contents) – display tar file contents
  - x (extract) – extract file(s) from a tar file
- tar function modifiers used in in conjunction with one of the three function characters:
  - f (file) – tar file to create, view or extract from
  - v (verbose) – display results of tar as they happen
  - z (compress) – use gzip compression
  - j (compress) – use bzip2 compression

The tar Command Examples

- Create a tar file: **$**tar –cvf *file.tar file(s)*

```
$ tar -cvf  systemd-config.tar  /etc/systemd
tar: Removing leading `/' from member names
/etc/systemd/
/etc/systemd/system/
/etc/systemd/system/multiuser.target.wants/
/etc/systemd/system/multiuser.target.wants/rsyslog.service
/etc/systemd/system/multiuser.target.wants/bind9.service
/etc/systemd/system/multiuser.target.wants/ssh.service
/etc/systemd/system/syslog.service
/etc/systemd/system/sshd.service
```

The tar Command Examples

- Use –tf (basic file list) or –tvf (file details) to view tar file contents:

```
$ tar -tf  systemd-config.tar
tar: Removing leading `/' from member names
etc/systemd/
etc/systemd/system/
etc/systemd/system/multiuser.target.wants/                      /
etc/systemd/system/multiuser.target.wants/rsyslog.service       /
etc/systemd/system/multiuser.target.wants/bind9.service         /
etc/systemd/system/multiuser.target.wants/ssh.service           /
etc/systemd/system/syslog.service
etc/systemd/system/sshd.service
```

The tar Command Examples

- Use –xvf to extract file(s) from a tar file to the current directory (default). Use –C (uppercase) to specify an alternate directory to extract to:

```
$ tar -xvf  systemd-config.tar  -C  /tmp
etc/systemd/
etc/systemd/system/
etc/systemd/system/multiuser.target.wants/
etc/systemd/system/multiuser.target.wants/rsyslog.service
etc/systemd/system/multiuser.target.wants/bind9.service
etc/systemd/system/multiuser.target.wants/ssh.service
etc/systemd/system/syslog.service
etc/systemd/system/sshd.service
```

The gzip Command

- `gzip` (GNU zip) is similar to the `zip` command, but differences include:
  - It replaces the original file with the zipped file
  - When it compresses recursively, it doesn't merge the files together, but rather compresses each file individually
- Example:

```
$ ls red*
red.txt
$ gzip red.txt
$ ls red*
red.txt.gz
```

The gunzip Command

- Use `-l` option (gunzip –l) to view the amount of compression of an existing archived file
- Gunzip unzips a gzipped file
- Compressed file will be removed and replaced with original file
- Example:

```
$ ls red*
red.txt.gz
$ gunzip red.txt
$ ls red*
red.txt
```

Avoid replacement

- Use the `-c` option to not have the `gzip` command replace the original file with the compressed file.
- Result is sent to standard output
- Can be redirected into a file:

```
$gzip -c words > word.gz
```

The bzip2 and bunzip2 Commands

- Very similar to `gzip` and `gunzip` commands, but has different technique to compress data
- Supports previously mentioned options for `gzip` EXCEPT the `-r` option

The zip Command

- Used to merge files together into a single file and compress the resulting file
- Syntax:

  `zip [options…] zipfile files…`
- Example:

  $ zip ./example/package ./example/*
  adding: example/one (stored 0%)
  adding: example/three (stored 0%)
  adding: example/two (stored 0%)
- The zipfile name will automatically have a ".zip" extension added if it is not specified

The zip Command

- Use the `-r` option to zip entire directory structures:

```
$ zip -r ./example/logfiles /var/log/cups/
  adding: var/log/cups/ (stored 0%)
  adding: var/log/cups/access_log.3.gz (stored 0%)
  adding: var/log/cups/access_log.2.gz (stored 0%)
  adding: var/log/cups/access_log.6.gz (stored 0%)
  adding: var/log/cups/error_log (stored 0%)
  adding: var/log/cups/error_log.1.gz (stored 0%)
adding: var/log/cups/access_log (deflated 85%)
  adding: var/log/cups/access_log.4.gz (stored 0%)
  adding: var/log/cups/access_log.7.gz (stored 0%)
  adding: var/log/cups/page_log (stored 0%)
$ ls ./example/
logfiles.zip  one  package.zip  three  two
```

The unzip Command

- Use the `-l` option to view the contents of a zip file
- Use the `unzip` command with no options to unzip a file

The xz Command

- The `xz` command is a highly-efficient compression utility
- Compresses/decompresses single files as input, and does not bundle multiple files into a single archive
- Common use is to compress an archived file created with `tar` or `cpio`

The xz Command

- Use –z option to compress a group of files individually
- Use –d option to decompress files
- The following example archives the test directory using tar before compressing it with xz:

```
$ tar -cf testdirectory.tar ./test/
$ ls
Desktop   Downloads  Pictures  Templates  test
Documents  Music     Public    Videos     testdirectory.tar
$ xz -z testdirectory.tar
$ ls
Desktop   Downloads  Pictures  Templates  test
Documents  Music     Public    Videos     testdirectory.tar.xz
```

The cpio Command

- `cpio` (copy in-out) is an archive command which can merge many files into a single file
- Three modes:
  - -In copy-out mode: copy files into an archive
  - -In copy-in mode: either list the archive file contents or copy files out of an archive.
  - -In copy-pass mode: copy files from one directory to another

The cpio Command

- To create a cpio archive, use the following:
  ```
  $ls | cpio -ov > archive.cpio
  ```
- The `ls` command generates a list of files to back up
- The `-o` puts the `cpio` in "copy-out" mode
- The `-v` makes the command "verbose"
- Output would go to the screen; `> archive.cpio` captures the output to a file

The cpio Command

- To extract a cpio archive, use the following:

```
$echo "/tmp/home.cpio" | cpio -iud
```

- The `echo` command sends the file name into the command
- Options:
  - `i` = extract from cpio archive
  - `d` = create directories
  - `u` = override existing files

The cpio Command

- To "pass through", use the following:
  ```
  $find ~ | cpio -pd /tmp/destination
  ```
- `-p` = Pass data to another directory
- Consider using `-print0` with the `find` command to avoid problems of file name with spaces in the name
- If you use `-print0` with the `find` command, then use the `-null` option for the `cpio` command

The dd Command

- A utility for copying files at the *bit* level
  - It can be used to clone or delete (wipe) entire disks or partitions
  - It can be used to copy raw data to removable devices, such as USB drive and CDROMs
  - It can backup and restore the MBR (Master Boot Record)
  - It can be used to create large files for virtual memory

The dd Command

- Uses non-hyphen options:
  - `if` - the input file to be read
  - `of` - the output file to be writte.
  - `bs` - the block size to be used
  - `count` - the number of blocks to read from the input file

The dd Command

- Create a large file:

```
$dd if=/dev/zero of=/tmp/swapex
  bs=1M count=500
```

- Clone from a hard drive:

```
$dd if=/dev/sda of=/dev/sdb
```