**cpio command in Linux with Examples**

**cpio** stands for “**copy in, copy out**“. It is used for processing the archive files like *\*.cpio* or *\*.tar*. This command can copy files to and from archives.

**Synopsis:**

* **Copy-out Mode:** Copy files named in name-list to the archive

**Syntax:**

cpio -o < name-list > archive

* **Copy-in Mode:** Extract files from the archive

**Syntax:**

cpio -i < archive

* **Copy-pass Mode:** Copy files named in name-list to destination-directory

**Syntax:**

cpio -p destination-directory < name-list

**Policy Options:**

* **-i, –extract:** Extract files from an archive and it runs only in copy-in mode.
* **-o, –create:** Create the archive and it runs only in copy-out mode.
* **-p, –pass-through:** Run in copy-pass mode.
* **-t, –list:** Print a table of contents of all the inputs present.

**Operation modifiers valid in any Mode:**

* **-B:** Changes the I/O block size to 5120 bytes.
* **-c:** Use the old portable (ASCII) archive format.
* **-C, –io-size=NUMBER:** Set the I/O block size to the given particular NUMBER of bytes.
* **-D, –directory=DIR:** Changes to Directory *DIR*.
* **-H, –format=FORMAT:** Use given arc.
* **-v, –verbose:** List the files processed in a particular task.
* **-V, –dot:** Print “.” for each file processed in a particular task.
* **-W, –warning=FLAG:** Control warning display. Currently FLAG is one of ‘*none*‘, ‘*truncate*‘, ‘*all*‘.

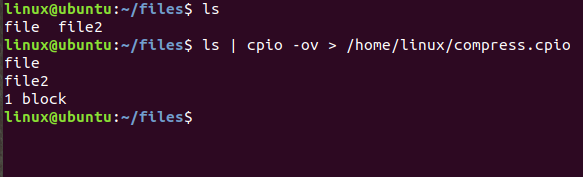
**Examples:**

* **To create a \*.cpio file :** We can create \*.cpio files containing files and directory with the help of cpio command.

**Syntax:**

cpio -ov < name-list > archive

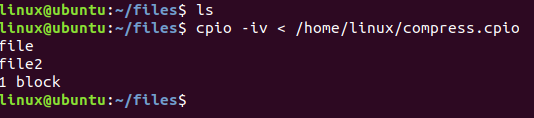
Here *-ov* is used as -o create the new archive and -v list the files processed.



* **To extract a \*.cpio file:**We can extract *\*.cpio* files containing files and directory with the help of cpio command.

**Syntax:**

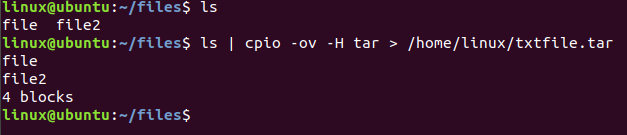
cpio -iv < archive



* **To create \*.tar archive file using cpio:**The cpio helps to create a *\*.tar* archive.

**Syntax:**

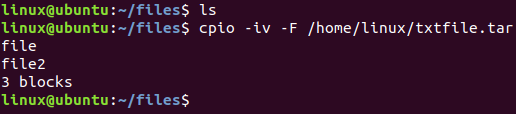
cpio -ov -H tar > archive



* **To extract \*.tar archive file using cpio:**The cpio helps to extract \*.tar files containing files and directory.

**Syntax:**

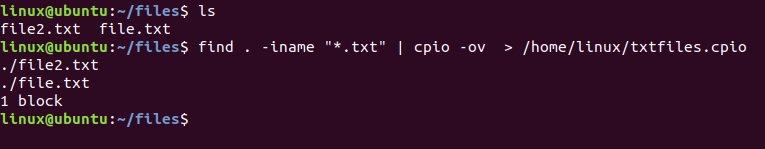
cpio -iv -F < archive



* **To create a \*.cpio archive with selected files:**We can create *\*.cpio* files containing specific files with the help of cpio command. In the example we are using *.txt* files.

**Syntax:**

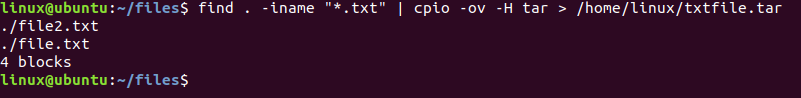
find . -iname "\*.txt" | cpio -ov > archive



* **To create a \*.tar archive with selected files:**We can create *\*.tar* files containing specific files with the help of cpio command. In the example we are using *.txt* files.

**Syntax:**

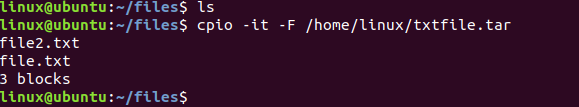
find . -iname "\*.txt" | cpio -ov -H tar > archive



* **To only view \*.tar archive file using cpio:** The cpio helps to view *\*.tar* files containing files and directory.

**Syntax:**

cpio -it -F < archive



**Note:**

* To check for the manual page of cpio command, use the following command:

**man cpio**

* To check the help page of cpio command, use the following command:

**cpio --help**

# Linux cpio Examples: How to Create and Extract cpio Archives

cpio command is used to process archive files (for example, \*.cpio or \*.tar files).

cpio stands for “copy in, copy out”.

cpio performs the following three operations.

* Copying files to an archive
* Extracting files from an archive
* Passing files to another directory tree

cpio takes the list of files from the standard input while creating an archive, and sends the output to the standard output.

### **1) Create \*.cpio Archive File**

You can create a \*.cpio archive that contains files and directories using cpio -ov

$ cd objects

$ ls

file1.o file2.o file3.o

$ ls | cpio -ov > /tmp/object.cpio

As seen above, the ls command passes the three object filenames to cpio command and cpio generates the object.cpio archive.

### **2) Extract \*.cpio Archive File**

cpio extract: To extract a given \*.cpio file, use cpio -iv as shown below.

$ mkdir output

$ cd output

$ cpio -idv < /tmp/object.cpio

### **3) Create \*.cpio Archive with Selected Files**

The following example creates a \*.cpio archive only with \*.c files.

$ find . -iname \*.c -print | cpio -ov >/tmp/c\_files.cpio

### **4) Create \*.tar Archive File using cpio -F**

We already know how to use the [tar command](https://www.thegeekstuff.com/2010/04/unix-tar-command-examples/) effectively.

Did you know that you can also use cpio command to create tar files as shown below?

$ ls | cpio -ov -H tar -F sample.tar

As seen above, instead of redirecting the standard output you can mention the output archive filename with the option -F.

### **5) Extract \*.tar Archive File using cpio command**

You can also extract a tar file using cpio command as shown below.

$ cpio -idv -F sample.tar

### **6) View the content of \*.tar Archive File**

To view the content of \*.tar file, do the following.

$ cpio -it -F sample.tar

### **7) Create a \*.cpio Archive with the Original files that a Symbolic Link Points**

cpio archive can be created with the original files that a symbolic link is referring to as shown below.

$ ls | cpio -oLv >/tmp/test.cpio

### **8) Preserve the File Modification Time while restoring \*.cpio**

The modification time of the files can be preserved when we are restoring the cpio archive files as shown below.

$ ls | cpio -omv >/tmp/test.cpio

### **9) Manipulate Linux and Kernel image files using cpio**

[How to View, Modify and Recreate initrd.img](https://www.thegeekstuff.com/2009/07/how-to-view-modify-and-recreate-initrd-img/) – As we discussed a while back, we can also use cpio command to manipulate initrd.img file.

### **10) Copy Directory Tree from One to Another**

cpio allows you to copy one directory contents into another directory without creating an intermediate archive. It reads the file list from the standard input and pass it to the target directory.

The example below copies the files and sub-directories of objects directory into /mnt/out directory.

$ mkdir /mnt/out

$ cd objects

$ find . -depth | cpio -pmdv /mnt/out

In the above example:

* cpio option -p makes cpio to use pass through mode. Its like piping cpio -o into cpio -i.
* cpio option -d creates leading directories as needed in the target directory.