<https://linuxconfig.org/how-to-use-find-command-to-search-for-files-based-on-file-size>

**How to use find command to search for files based on file size**

This config will list few examples on how to search files using find command based on the file size.

## Example 1

Let's start by searching for all files in our current working directory with file size of 6MB:

$ find . -size 6M

The suffix M denotes Megabytes that is 1048576 bytes. The other available suffixes to our disposal are:

* b - 512-byte blocks (this is the default if no suffix is used)
* c - bytes
* w - two-byte words
* k - Kilobytes
* M - Megabytes
* G - Gigabytes

## Example 2

The below example will search for all files greater than 2 Gigabytes. Note the use of+ sign:

$ find . -size +2G

## Example 3

The above find command was used to search for all files greater than specified size. Next, find command example will search for all files with less than 10 Kilobytes in size. Note the use of- sign:

$ find . -size -10k

## Example 4

In this example we will use find command to search for files greater than 10MB but smaller than 20MB:

# find . -size +10M -size -20M

## Example 5

In this example we use the find command to search for files in /etc directory which are greater than 5MB and we also print its relevant file size:

$ find /etc -size +5M -exec ls -sh {} +

6.1M /etc/udev/hwdb.bin

## Example 6

Find first 3 largest files located in a in a current directory recursively:

$ find . -type f -exec ls -s {} + | sort -n -r | head -3

## Example 7

Find first 3 smallest files located in a in a current directory recursively:

$ find /etc/ -type f -exec ls -s {} + | sort -n | head -3

## Example 8

In the last example we will use find command to search for empty files:

$ find . -type f -size 0b

OR

$ find . -type f -empty

<https://stackoverflow.com/questions/20224822/how-to-check-if-a-file-contains-only-zeros-in-a-linux-shell>

# [How to check if a file contains only zeros in a Linux shell?](https://stackoverflow.com/questions/20224822/how-to-check-if-a-file-contains-only-zeros-in-a-linux-shell)

How to check if a large file contains only zero bytes ('\0') in Linux using a shell command? I can write a small program for this but this seems to be an overkill.

If you're using bash, you can use read -n 1 to exit early if a non-NUL character has been found:

<your\_file tr -d '\0' | read -n 1 || echo "All zeroes."

where you substitute the actual filename for your\_file.

The "file" /dev/zero returns a sequence of zero bytes on read, so a cmp file /dev/zero should give essentially what you want (reporting the first different byte just beyond the length of file).

If you have Bash,

cmp file <(tr -dc '\000' <file)

If you don't have Bash, the following should be POSIX (but I guess there may be legacy versions of cmp which are not comfortable with reading standard input):

tr -dc '\000' <file | cmp - file

Perhaps more economically, assuming your grep can read arbitrary binary data,

tr -d '\000' <file | grep -q -m 1 ^ || echo All zeros

I suppose you could tweak the last example even further with a dd pipe to truncate any output from tr after one block of data (in case there are very long sequences without newlines), or even down to one byte. Or maybe just force there to be newlines.

tr -d '\000' <file | tr -c '\000' '\n' | grep -q -m 1 ^ || echo All zeros

It won't win a prize for elegance, but:

xxd -p file | grep -qEv '^(00)\*$'

xxd -p prints a file in the following way:

23696e636c756465203c6572726e6f2e683e0a23696e636c756465203c73

7464696f2e683e0a23696e636c756465203c7374646c69622e683e0a2369

6e636c756465203c737472696e672e683e0a0a766f696420757361676528

63686172202a70726f676e616d65290a7b0a09667072696e746628737464

6572722c202255736167653a202573203c

So we grep to see if there is a line that is not made completely out of 0's, which means there is a char different to '\0' in the file. If not, the file is made completely out of zero-chars.

(The return code signals which one happened, I assumed you wanted it for a script. If not, tell me and I'll write something else)

EDIT: added -E for grouping and -q to discard output.