I'm not robot

reCAPTCHA

**Continue**

I'm not robot

reCAPTCHA

# Argument list too long mv

I'm trying to rename a number of files and folders. I tried to rename this command 's/test/test2/', but I wasn't able to perform /usr/bin/rename: Argument list too long I searched a lot and tried different solutions, but I don't get it. The last one I tried was this one find. -type f -exec rename -n 's/test/test2/' ** which I got the same error. Thank you very much! Edit: Also tried xargs -0 rename -n's/test/test2/' ** and I have xargs: can't fit single argument within argument list size limitation If you do a lot of command line stuff in Linux, sooner or later you'll be on the argument list too long error when copying, moving, deleting, listing etc. files in a directory that contains a lot of files. This article looks at a few ways to solve this problem. For the tests performed in this article, I created two directories, source and target, and created 20,000 blank text files as follows: mkdir source mkdir target for i in 'seq 1 20000'; Touch Source/$i.txt; where: mv source/* target/ cp source/* target/ rm source/* ls source/* ls source/* leads to errors such as: -bash: /bin/mv: Argument list too long -bash: /bin/rm: Argument list too long to work around the problem, you can use xargs or a combination of two. With the ls command, you can do this to move all files from the source to the target: ls -1 source | xargs -i mv source/- target/ The -i flag in xargs instructs it to use string nerds, and the wildcard is the placeholder. You cannot use a pattern for ls such as ls *.txt because you get the rurugment list errors too long before the data can be piped through to xargs. To make pattern matches, you must use the find command. If you use find to do the same shift from source to target, you can do this: Find source/ -name *.txt -exec mv - Target ; or this, with xargs instead of the -exec flag: Find source/ -name *.txt | xargs -i mv 'target/ In the above find examples, you can omit the name *.txt part to move everything and/or add more pattern matches, such as .B -type f, to match only files. All three examples above take about the same time to run, although i found while testing the find ... -exec method to be a bit faster than find ... xargs The find method is better than the ls method because it allows you to specify file name patterns and can give you much more control over what you copy, move, delete, etc. As the last set of examples, you can perform the following steps to: Delete files from the target directory: ls -1 target | xargs -i rm target/ or find target/ -name *.txt -exec rm or find target/name *.txt | having to delete/move/etc. files in this way is more difficult and time-consuming than just doing something like rm -f source/*, but sometimes this is the Way to do it if there are too many files that match your pattern. Over time, the memory used on Linux systems that you manage will grow. Therefore, at some point, you will try to delete, move, search, or otherwise edit thousands of files with commands such as rm, cp, ls, mv, and others, all of which are subject to this restriction. As such, you will eventually come to the argument list too long errors listed below. The Errorbash: /usr/bin/rm: Argument list too long All regular system commands such as rm, ls, mv, cp, and so on are subject to this limitation. What does argument list too long argument list too long specifies if a user feeds too many arguments into a single command that reaches the ARG_MAX limit. The ARG_MAX defines the maximum length of arguments for the exec function. An argument, also called a command-line argument, can be defined as input to a command to control that command-line process. Arguments are entered into the terminal or console after the command is entered. Multiple arguments can be used together. they are processed in the order in which they entered, from left to right. This limit on the length of a command is set by the operating system. You can check the maximum arguments limit on your Linux system with this command:getconf ARG_MAXWhich will return something like this: hydn@centos: getconf ARG_MAX 2097152The error argument list too long means that you have exceeded the maximum command line length allowed for arguments in a command. The solutionThere are a number of solutions to this problem (bash: /usr/bin/rm: argument list too long). Remove the folder itself, and then recreate it. If you are trying to delete ALL files and folders in a directory, instead of using a wildcard * to remove (i.e.rm *), try:rm -r /path/to/directory/If you still need this directory, recreate it with the mkdir command. Bulk deletion of files with the find command:You can use the find command to find each file and then delete it:find . -type f -deleteOr to delete only certain file types (i.e. . txt files) or in other ways such as:find. -name '.txt' -type f -delete If I run the folder mv folder2/*.*, I get the error argument list too long. I find an example of ls and rm, dealing with this error, with find folder2 -name *.*. But I have difficulty applying them to mv. xargs is the tool for the job. That, or find with -exec ... {} +. These tools execute a command multiple times, with as many arguments as can be passed in one go. Both methods are simpler if the variable argument list is at the end, which is not the case here: the last argument for mv is the target. With GNU utilities (i.e. on non-embedded Linux or Cygwin), the -t to mv option is useful to pass the target first. If the file names are neither empty nor can simply specify the file names as input for xargs (the Echo command is a bash-integrated so that it is not subject to the command line length limit): echo ! (*.jpg|*.png|*.bmp) | xargs mv -t targetdir You can use the -0 option for xargs to use a zero-separated input instead of the default input character format. printf '%s'0' ! (*.jpg|*.png|*.bmp) | xargs -0 mv -t targetdir Alternatively, you can generate the list of file names with find. To avoid repetition in subdirectories, use -type d -prune. Because no action is specified for the listed image files, only the other files are moved. To find. Name. -o -type d -prune -o -name '*.jpg' -o -name '*.png' -o -name '*.bmp' -o -exec sh -c 'mv exec mv -t targetdir/ If you don't have GNU utilities, you can use an intermediate envelope to get the arguments in the correct order. This method works on all POSIX systems. To find. Name. -o -type d -prune -o -name '*.jpg' -o -name '*.png' -o -name '*.bmp' -o -exec sh -c 'mv """"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""extended_glob"""""""""""""""""""""""""mv and other names to continue to refer to the external commands: setopt extended_glob zmodload -Fm zsh/files b:zf_*zf_mv -- . (jpg|png|bmp) targetdir/ or with ksh-like globs: setopt ksh_glob zmodload -Fm zsh/files b:zf_*zf_mv -- ! (*.jpg|*.png|*.bmp) targetdir/ Alternative with GNU mv and zargs: autoload -U zargs setopt extended_glob zargs -- ./*. (jpg|png|bmp) -- mv -t targetdir/ I am a new Linux user. We have a backup server and most of our developers sftp or ftp to upload files. I just need to backup the last four weeks. When I try to list or remove directories, an error message cannot be removed. I experience the message argument list too long. How do I resolve this issue when commands are executed in the bash/ksh/csh/tcsh/zsh shell? Each command on Linux and UNIX-like systems accepts a parameter commonly referred to as command arguments (or args). For example, the command 'cd /etc' has a command-line argument, namely /etc. Some commands can accept more than two arguments and respond to deployed args. For example, consider the following cp command: cp /etc/file1 /etc/file2 /etc/file3 /mnt/pen[donotprint] Tutorial details Difficulty Easy (rss) Root privileges No Requirements find &amp; xargs commands Time N/A [/donotprint] The cp command has a total of four command line arguments. The shell can contain a maximum of 131072 bytes for command-line arguments. You try to pass more than this number, you will be greeted with an error that reads as follows: Argument list too long Remember that each character needs a byte of memory. The maximum length of the command line argument The maximum length of the command line argument is limited to *128k* within the Linux kernel. To find your limits, type: Example 2621440 The boundary affects the execve() kernel function used by all other exec() functions (execl, execlp, execle, etc.). Option #1: How to get rid of argumentlist too long error problem while rm, ls, cp, mv or other shell commands The best way to deal with this problem is the combination of find and xargs commands. The syntax is: find /dir/to/search -type f -name 'file-pattern' -print0 | xargs -0 -n 10 command name find /dir/to/search -type f -iname *. EXT -print0 | xargs -0 Command name find /dir/to/search -type f -iname *. EXT command Only list files if you error argument list too long Use one of the following syntax to find only files /dir/to/search -type f -iname*. EXT -ls find /nas01/data/ -type f -iname *.doc -ls OR the following portable syntax: find /nas01/data/ -name '*.doc' -exec ls ' + 'GNU/Find' find /nas01/data/ -type f -name '*.doc' -exec ls -l Avoid recursion Use the following GNU Find option: find /nas01/data/ -name '*.doc' -maxdepth 1 -exec ls -l ' + Simply delete files if argument list is too long error The syntax is as follows: 'GNU/find and BSD/find can't work on older *nix find /dir/to/search -type f -iname*. EXT -delete find '/Downloads/ -type f -iname *.avi -delete find '/Projects/ -type f -iname *.pdf -delete' use xargs and rm command "find '/Downloads/ -name '*.avi' -print0 | xargs -0 -n 10 rm -f More portable syntax: find '/Projects/ -name '*.pdf' -exec rm -vf '+ To avoid recursion, use the following GNU Find option: . GNU finds and can also work with bsd find .pdf . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .xargs -0 -n 10 rm -f You can also use the GNU parallel command: .,man parallel for more ' find '/Downloads/ -type f -iname '*.avi' -print0 | parallel -X -0 rm -vf Where, -name *.pdf : Search pattern. -type f : Match only files. -print 0 : Avoid nasty surprises when dealing with strange file names with spaces, ', , tabs and line re-enmities. The primary -print0 in conjunction with xargs -0 as an effective alternative to dealing with strange file names. -maxdepth 1 : Descend at most one directory level below the command line arguments, i.e. avoid recursion. -exec rm -f + OR -exec ls + : Run rm or ls commands on as many files as possible. Option #2: Iterating with the -0 to avoid arg list too long errors The syntax is as follows and it is very slow and can not work at all: for f in *; do command $i; delete all pdf files for f in *.pdf; do /bin/rm -rf $f; done .avi .mp4 . , . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . , . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . , reine Bash-Lösung mit Arrays, basierend auf . , . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . $HOME/Downloads/*.mp4) for ((i=0; i&lt; #files[*]; i+=100)); do /bin/rm -vf -files[*]:i:100' done command summary, to avoid the limit in a shell, deals with strange file names to find /dir/to/search/ -name file-pattern-search -exec-command + find and xargs based solution - find /dir/to/search/ -name file-pattern-search -print0.xargs -0 Command - For gnu parallel fan - find /dir/to/search/ -name 'file-pattern-search' parallel -X -0 command - older and not recommended solution; but here for historical reasons only kept for i in *; do command $i; more info see man pages: mv(1) mv(1)