
gfast-copy Documentation

Release 1.0

Brüggemann Eddie

Oct 19, 2017

Contents:

1	gfast-copy	1
1.1	Introduction	1
1.2	Graphical User Interface description	2
1.3	NOTE	2
1.4	Credits	3
1.5	License	3
2	fast-copy	5
2.1	fast-copy command-line tool	5
2.2	Options	5
2.3	How does the algorithm work	7
2.4	Wrapper script	7
2.5	Copyright	8
	Index	9

program **gfast-copy** (a graphical fast chunk file copy program.)

author Eddie Brüggemann <mrcyberfighter@gmail.com>

language C

version 1.0

release Oct 19, 2017

1.1 Introduction

gfast-copy stands for **Graphical Fast Copy** and it is build on a *very fast copying algorithm*.

Note: This tool is implemented into the UNIX spirit:

A single program which can do an unique task but do it well.

It copy a single file to a destination in a practice way.

As fast as it can !!!

gfast-copy only provides a graphical interface for the **fast-copy** command-line tool, that I have implement. Which can copy a file like a film (Around 700 Mo) in few seconds.

Note: I have implement it because I often copy videos from a location to another:

Per example from the folder named ToWatch to the folder Seen.

Or *.iso images from the folder Download to the folder ISO_Collection.

Because the O.S (Operating System) can be slow by chunk copy operations.

1.2 Graphical User Interface description

1.2.1 Interface

You get to vertically align buttons:

1. Source
2. Destination

Use the `Source` labeled button to set *the file to copy* Source (Ctrl + S).

This will display the *basename* of the file to copy *onto the button*.

Use the `Destination` labeled button to set *the file destination* Destination (Ctrl + D).

This will display the *basename* you have choose for the destination file.

And at they are 2 icons buttons:

- Remove source file toggle button (Ctrl + R).
- Run copying (Ctrl + Return).

note Take a look at my self-built `GtkSmartIconButton` widgets.

Once the copying done, the button contains their original string and icons.

1.2.2 Menu bar

You get a menu bar too with few menu buttons.

1. Files
 - *Files* -> *S*et source file (Ctrl + S).
 - *Files* -> *S*et destination file (Ctrl + D).
 - *Files* -> *R*un copying (Ctrl + Return).
 - Quit (Ctrl + Q).

2. Configuration
 - *Configuration* -> *L*ight theme.
 - *Configuration* -> *D*ark theme.

these are radiobuttons to choose your theming.

 - *Configuration* -> *D*elete source file (Ctrl + R).

3. Info ?
 - *Info ?* -> *R*ead me (Ctrl + N).
 - *Info ?* -> *A*bout (Ctrl + A).

1.3 NOTE

I have implement this tool for my personal requirement: good by Hollywood !

So if you think your system is too slow by copying chunk (big) files use (g)fast-copy !

1.4 Credits

Thanks to my beloved mother, my family and the doctors.

Stay away from drugs drugs destroy your brain and your life.

1.4.1 Windows *.exe file

Thanks to the gnome-bugzilla team which help to provide you a version of (g)fast-copy.

Without their help only the MSYS2 and Cygwin users would be able to launch (g)fast-copy.

1.5 License

gfast-copy an easy to use graphical fast copy tool for big files.

Copyright (C) 2017 Brüggemann Eddie.

This file is part of gfast-copy.

gfast-copy is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

gfast-copy is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with gfast-copy. If not, see <http://www.gnu.org/licenses/>

program **fast-copy** (a fast chunk file copy program).

author Eddie Brüggemann <mrcyberfighter@gmail.com>

language C

version 1.0

release Oct 19, 2017

2.1 fast-copy command-line tool

fast-copy is a **command-line tool** which can work in 2 different fashions:

- Using SYSCALL'S (System Calls).
- Using a FILE *streams* pointer.

note Per default it make usage of the file streams.

fast-copy determine the best buffer size and then process displaying a fancy progress-bar if needed into the terminal.

Fast copy takes at least 2 argument the source and the destination to copy the file on **GNU/Linux** and **MacOSX**.

windows The command-line tool is available under **Windows** using only the *file streams* and no **progress-bar** feature is provided.

2.2 Options

```
$ fast-copy -h
fast-copy - a fast chunk file copy program.

Usage : fast-copy -r input-file -w output-file [-o] [-e] [-s] [-h]
```

```
-r Read from file (required).
-w Write to file (required).
-o Overwrite destination file (optional).
-e Erase source file (optional).
-s Use syscalls instead of streams (optional only UNIX).
-h Print this help message.

- Copyright (©) 2017 Brüggemann Eddie <mrcyberfighter@gmail.com> GPLv3.
$
```

—

-r
-r (read from file) -> Input file (required).

-w
-w (write to file) -> Output file (required).

-o
-o (overwrite) -> Overwrite destination file (optional).

-e
-e (erase) -> Erase source file (optional).

-s
-s (syscalls) -> Use syscall instead of streams (optional only UNIX).

-h
-h (help) -> Print help message.

—

The default behavior is to copy *source* to *destination* file using **file stream**, which you can change using the **-s** switch on UNIX systems.

If the destination file exists **fast-copy** will ask interactively what to do: overwrite or not the file.

The progress bar is only displayed for files greater than : `sizeof_file > (blocksize * 8 * 16 * 16)`

This value mean to be 16 Mio using **syscalls** or using **file streams** on my system.

note The `blocksize` is computed to get the best buffer size: the number of bytes by the copying operation and is system dependent.

Finally **fast-copy** will synchronize the file system.

—

You can inhibit the interactive overwriting questioning by using the **-o** > (*overwrite*) switch.

—

You can tell program: *fast-copy* to use syscalls (open, read and write) using the **-s** -> (*syscall*) switch.

To know that by every **blocksize** copy a SYSCALL (System Call) will be made by handle by the **kernel**.

Else **file streams** are buffered operations avoiding calling SYSCALL to often.

—

You can tell **fast-copy** to erase the source file once the copy work done using the **-e** -> (*erase*) switch.

2.3 How does the algorithm work

1. At first it will check for the best buffer size.

Note:

- Using file streams: it will look up the `BUFSIZ` constant if it's great enough else the program set it to 8192 octets.
 - Using syscalls it will look up the `st_blocksize` from the `stat()` function and set it as buffer size.
-

2. Then it will define a buffer with the optimal size.
3. It will copy the source file to the destination in a loop using the defined buffer.
4. Finally it will synchronize the file-system using `fflush()` or `fsync()`.

2.4 Wrapper script

You can build a wrapper script around the **fast-copy** command-line tool,

per example using **zenity**, **xclip** or like, to get the first argument.

And for the second you need a tool which provides a file-chooser.

As like this one, using zenity and nautilus-action:

```
#!/bin/bash

#####
# Wrapper for fast-copy using zenity and nautilus-actions for #
# speed up the file copying.                                #
#####

function usage() {

    echo "usage $(basename $0) -r input-file -w output-file [-o] [-s] [-e] [-h]" ;

    exit 1 ;
}

if [[ $# -ne 1 ]] ; then

    # The first argument is given from nautilus-actions.

    usage ;

fi

destination=`zenity --file-selection --save --confirm-overwrite --filename="
↳$(basename "${1}") "`

# Here the source file comes from nautilus action i.e
# the contextual gnome-desktop add-on nautilus-actions
source_filepath=${1}

if [[ -n "${destination}" ]] ; then
```

```
fast-copy -r "${source_filepath}" -w "${destination}" -o
zenity --info --text="Copy completed" --no-wrap --no-markup
fi
exit 0
```

2.5 Copyright

fast-copy - a fast single chunk file copy program.

Copyright (C) 2017 Brüggemann Eddie.

This file is part of fast-copy.

fast-copy is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

fast-copy is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with gfast-copy. If not, see <<http://www.gnu.org/licenses/>>

Symbols

- e command line option, 6
- h command line option, 6
- o command line option, 6
- r command line option, 6
- s command line option, 6
- w command line option, 6

C

command line option

- e, 6
- h, 6
- o, 6
- r, 6
- s, 6
- w, 6