



GPL-3-free replacements of coreutils

1	<b>Contents</b>	
2	<b>Coreutils GPLv2</b>	<b>2</b>
3	<b>Alternatives</b>	<b>3</b>
4	utils-coreutils . . . . .	3
5	BSDutils . . . . .	4
6	Busybox . . . . .	5
7	Nbase . . . . .	5
8	FreeBSD . . . . .	6
9	Sbase and Ubase . . . . .	6
10	Heirloom . . . . .	7
11	<b>Replacement: utils-coreutils</b>	<b>7</b>
12	<b>Testing</b>	<b>9</b>
13	<b>Initial test and results</b>	<b>9</b>
14	<b>Migration</b>	<b>10</b>
15	Due to the nature of Apertis and its target markets there are <a href="#">licensing terms that</a>	
16	are <a href="#">problematic</a> <sup>1</sup> and that forces the project to look for alternatives packages.	
17	The <code>coreutils</code> package is good example of this situation as its license changed	
18	to GPLv3 and as result Apertis cannot provide it in the <code>target</code> repositories and	
19	images. The current solution of shipping an old version which precedes the	
20	license change is not tenable in the long term, as there are no upgrades with	
21	bugfixes or new features for such important package.	
22	This situation leads to the search for a drop-in replacement of <code>coreutils</code> , which	
23	need to provide compatibility with the standard GNU <code>coreutils</code> packages. The	
24	reason behind is that many other packages rely on the tools it provides, and	
25	failing to do that would lead to hard to debug failures and many custom patches	
26	spread all over the archive. In this regard the strict requirement is to support	
27	the features needed to boot a target image with ideally no changes in other	
28	components. The features currently available in our <code>coreutils-gplv2</code> fork are a	
29	good approximation.	
30	Besides these specific requirements, there are general ones common to any Open	
31	Source Project, such as maturity and reliability. Particularly important aspects	
32	are also the available community support, the development process and user	
33	adoption.	
34	As a summary, below is the list of attributes	
35	• License suitable for inclusion in Apertis	
36	• Compatible with GNU <code>coreutils</code>	
37	• Support for the features needed to boot a target image	

<sup>1</sup><https://martyn.pages.apertis.org/apertis-website/policies/license-expectations/>

- 38 • User adoption
- 39 • Community support
- 40 • Long term solution

## 41 Coreutils GPLv2

42 Currently Apertis provides `coreutils-gplv2`, with the following features

```
43 [ base64 basename cat chgrp chmod chown chroot cksum comm cp csplit cut date dd
44 df dir dircolors dirname du echo env expand expr factor false fmt fold groups
45 head hostid id install join link ln logname ls md5sum md5sum.textutils mkdir
46 mkfifo mknod mktemp mv nice nl nohup od paste pathchk pinky pr printenv printf
47 ptx pwd readlink rm rmdir seq sha1sum sha224sum sha256sum sha384sum sha512sum
48 shred shuf sleep sort split stat stty sum sync tac tail tee test touch tr true
49 tsort tty uname unexpand uniq unlink users vdir wc who whoami yes
```

## 50 Alternatives

51 In order to perform a comparison among different projects this section list dif-  
52 ferent projects and metrics of each them. These metrics are quantitative ones,  
53 which can obtain from the Git log, and qualitative that can be derive from the  
54 first ones. The value of showing all these metrics is to allow non-technical users  
55 to clearly understand the comparison.

### 56 **utils-coreutils**

57 Link: <https://github.com/uutils/coreutils>  
58 Language: Rust  
59 License: MIT  
60 GNU compatibility: High (it is the project goal)  
61 User adoption: Low  
62 Completeness: Missing 14 commands  
63 Started: 2013  
64 Developers in last year: 40  
65 Commits in last year: 885  
66 Project status: Very active  
67 Community support: High  
68 Maturity: Medium

### 69 **Pros**

- 70 • High GNU compatibility
- 71 • High community support
- 72 • High community impact
- 73 • Portability in mind
- 74 • Ongoing development

- 75 • Implemented in a modern memory safe language
- 76 • Interest from Debian developers

### 77 **Cons**

- 78 • Missing commands and features
- 79 • Not used in production environments
- 80 • Depends on many Rust crates, which may not all be already available in
- 81 Debian

### 82 **Notes**

- 83 • Semi-done: `cp expr install ls more od printf sort split tail test date`
- 84 `join df`
- 85 • To do: `chcon csplit dd numfmt pr stty`
- 86 • Missing compared to `coreutils-gplv2`: `csplit dd dir pr stty vdir`
- 87 • Builds successfully on Apertis using the available Rust compiler
- 88 • Initial tests for basic features were successful
- 89 • A Debian Developer already ran some tests booting a Debian graphical
- 90 session with GNOME [using `utils-coreutils`](#)<sup>2</sup>

### 91 **BSDutils**

92 Link: <https://github.com/dcantrell/bsdutils>

93 Language: C

94 License: BSD

95 GNU compatibility: Low (project is only a port of OpenBSD compatible with

96 Linux)

97 User adoption: Very low

98 Completeness: Missing 25 commands, long options unsupported, other differ-

99 ences Started: 2019

100 Developers in last year: 1

101 Commits in last year: 86

102 Project status: Active

103 Community support: Low (base project high)

104 Maturity: Medium (base project high)

### 105 **Pros**

- 106 • Linux support
- 107 • Based on OpenBSD, which is a mature project

### 108 **Cons**

- 109 • Missing commands and features
- 110 • Not fully compatible with GNU as it is a port from OpenBSD
- 111 • Low community support for the port itself
- 112 • Not used in production environments

---

<sup>2</sup><https://sylvestre.ledru.info/blog/2021/03/09/debian-running-on-rust-coreutils>

- 113 • Original project only supports OpenBSD, Linux support added in a low  
114 activity fork
- 115 • Requires libbsd-dev

## 116 Notes

- 117 • This project is a port of tools from OpenBSD to have an BSD-licensed  
118 and lightweight replacement of GNU coreutils
- 119 • Provides a set of scripts to import new OpenBSD versions and a set of  
120 patches to be applied and provide Linux compatibility
- 121 • In order to upstream contributions might need to be done to this specific  
122 project or to OpenBSD
- 123 • Missing from coreutils-gplv2: base64 cksum dir dircolors hostid link  
124 md5sum md5sum.textutils od pathchk pinky ptx seq sha1sum sha224sum  
125 sha256sum sha384sum sha512sum shred shuf sum tac tail unlink vdir

## 126 Busybox

127 Link: <https://busybox.net/>

128 Language: C

129 License: GPLv2

130 GNU compatibility: High (compatibility in mind but a subset of features)

131 User adoption: Very high

132 Completeness: Commands with limited features

133 Started: 1999

134 Developers in last year: 27

135 Commits in last year: 299

136 Project status: Very active

137 Community support: High

138 Maturity: High

## 139 Pros

- 140 • High GNU compatibility
- 141 • High community support
- 142 • Very low footprint
- 143 • Already part of Apertis

## 144 Cons

- 145 • Supports a subset of features

## 146 Nbase

147 Link: <https://github.com/cheusov/nbase>

148 Language: C

149 License: BSD

150 GNU compatibility: Low (project is only a port of NetBSD compatible with  
151 Linux)

152 User adoption: Very low  
153 Completeness: Missing 33 commands  
154 Started: 2015  
155 Developers in last year: 1  
156 Commits in last year: 119  
157 Project status: Active  
158 Community support: Low  
159 Maturity: Medium

#### 160 **Pros**

- 161 • Linux support
- 162 • Based on NetBSD, which is a mature project

#### 163 **Cons**

- 164 • Missing commands and features
- 165 • Not fully compatible with GNU as it is a port from NetBSD
- 166 • Low community support
- 167 • Not used in production environments
- 168 • Requires NetBSD make, mk-configure, libbsd
- 169 • Original project only supports NetBSD, Linux support added in a low  
170 activity fork

#### 171 **Notes**

- 172 • This project is a port of tools from NetBSD compatible with other Unix  
173 like systems
- 174 • Missing from coreutils-gplv2: [ base64 chgrp chown chroot dir dircolors  
175 factor groups hostid install link md5sum md5sum.textutils od pathchk  
176 pinky ptx readlink sha1sum sha224sum sha256sum sha384sum sha512sum shred  
177 shuf sum tac unlink users vdir who whoami

#### 178 **FreeBSD**

179 Link: <https://github.com/freebsd/freebsd/tree/master/bin>  
180 Link: <https://github.com/freebsd/freebsd/tree/master/usr.bin>  
181 Language: C  
182 License: FreeBSD  
183 GNU compatibility: Very low  
184 User adoption: High  
185 Developers in last year: 72 (on usr.bin)  
186 Commits in last year: 423 (on usr.bin)  
187 Project status: Active  
188 Community support: High  
189 Maturity: High

#### 190 **Pros**

- 191 • High community support

192 **Cons**

- 193 • Missing commands and features
- 194 • No Linux support
- 195 • No GNU compatibility

196 **Sbase and Ubase**

197 Link: <https://gitlab.com/garbeam/src/-/tree/master/bin/sbase>

198 Link: <https://gitlab.com/garbeam/src/-/tree/master/bin/ubase>

199 Language: C

200 Project status: Inactive, no activity since 2016

201 Community support: None

202 **Pros**

- 203 • Linux support

204 **Cons**

- 205 • Project inactive

206 **Heirloom**

207 Link: [https://en.wikipedia.org/wiki/Heirloom\\_Project](https://en.wikipedia.org/wiki/Heirloom_Project)

208 Link: <https://wiki.archlinux.org/index.php/Heirloom>

209 Language: C

210 Project status: No activity since 2007

211 Community support: None

212 **Pros**

- 213 • Linux support

214 **Cons**

- 215 • Project inactive

216 **Replacement: `utils-coreutils`**

217 Based on the above comparison the best option is `utils-coreutils`, since it is  
218 the only one with the explicit goal of providing a fully compatible alternative  
219 to GNU `coreutils`, and it has a good community support which most probably  
220 will continue and improve in the future. The main risk is the current low user  
221 adoption and the lack of usage in production scenarios. It is worth to mention  
222 that the main license used in the project is MIT but further analysis needs to  
223 be done to confirm the licensing of all the used dependencies.

224 These risks enumerated will be handled by the testing and migration in order  
225 to provide a reliable approach.

226 As it has been mentioned the license used is MIT, and detailed information  
227 about its dependencies can be found in the [FOSSA analysis](#)<sup>3</sup>. Unfortunately,  
228 this report is not reliable since it shows several incorrect dependencies.

229 The following list shows the dependencies as reported by `cargo`

Package	License
ansi_term	MIT
arrayvec	MIT OR Apache-2.0
autocfg	MIT OR Apache-2.0
backtrace-sys	MIT OR Apache-2.0
bitflags	MIT OR Apache-2.0
bit-set	MIT OR Apache-2.0
bit-vec	MIT OR Apache-2.0
blake2-rfc	MIT OR Apache-2.0
byteorder	Unlicense OR MIT
cfg-if	MIT OR Apache-2.0
chrono	MIT OR Apache-2.0
constant_time_eq	CC0-1.0
data-encoding	MIT
dunce	CC0-1.0
either	MIT OR Apache-2.0
failure	MIT OR Apache-2.0
fake-simd	MIT OR Apache-2.0
fnv	MIT OR Apache-2.0
fs_extra	MIT
glob	MIT OR Apache-2.0
half	MIT OR Apache-2.0
hex	MIT OR Apache-2.0
ioctl-sys	MIT OR Apache-2.0
isatty	MIT OR Apache-2.0
maybe-uninit	MIT OR Apache-2.0
md5	MIT OR Apache-2.0
num-integer	MIT OR Apache-2.0
onig	MIT
onig_sys	MIT
pkg-config	MIT OR Apache-2.0
platform-info	MIT
ppv-lite86	MIT OR Apache-2.0
rand_chacha	MIT OR Apache-2.0
rand_pcg	MIT OR Apache-2.0
rust-ini	MIT
semver	MIT OR Apache-2.0

<sup>3</sup>[https://app.fossa.io/projects/git%2Bgithub.com%2Fuutils%2Fcoreutils?ref=badge\\_large%22](https://app.fossa.io/projects/git%2Bgithub.com%2Fuutils%2Fcoreutils?ref=badge_large%22)



Package	License
semver-parser	MIT OR Apache-2.0
sha1	BSD-3-Clause
sha2	MIT OR Apache-2.0
sha3	MIT OR Apache-2.0
smallvec	MIT OR Apache-2.0
strsim	MIT
syn	MIT OR Apache-2.0
synom	MIT OR Apache-2.0
synstructure	MIT
tempfile	MIT OR Apache-2.0
term_grid	MIT
term_size	MIT
term_size	MIT OR Apache-2.0
thread_local	MIT OR Apache-2.0
typenum	MIT OR Apache-2.0
unix_socket	MIT OR Apache-2.0
vec_map	MIT OR Apache-2.0
wild	MIT OR Apache-2.0
winapi-util	Unlicense OR MIT
xattr	MIT OR Apache-2.0

## 230 Testing

231 In order to confirm the missing features/commands in the `utils-coreutils`  
 232 which are required by Apertis a testing needs to be performed. The steps  
 233 proposed are:

- 234 • Run initial tests on target images
  - 235 – Test booting standard target images
  - 236 – Test installing/removing packages
- 237 • Run current `coreutils-gplv2` test plan with `utils-coreutils`
- 238 • Run `utils-coreutils` as default on development environments
- 239 • Make `utils-coreutils` and all the Rust crates it depends on available in  
 240 Debian
- 241 • Provide long-term maintenance of the new packages in Debian as well  
 242 Note that some effort is being driven by `utils-coreutils` community to use  
 243 the `coreutils` test case to generate a report for the still missing features.  
 244 This will be a nice to have feature but it is more than it is actually required  
 245 for this stage.

## 246 Initial test and results

247 As part of an initial test using `utils-coreutils` the following steps have been  
248 taken

- 249 • Replace utilities from `coreutils-gplv2` with the ones provided by `utils-`  
250 `coreutils`
- 251 • Boot target image without issues
- 252 • Reinstall package `libc6` without issues

253 These initial results are promising, however more detailed tests should be  
254 planned and executed to spot potential issues.

## 255 Migration

256 As of `v2022dev3`, `utils-coreutils` is the default `coreutils` implementation  
257 shipped on the Apertis reference images for devices, while GNU `coreutils`  
258 remains in use in the package building pipelines and on the SDK images.  
259 This work involved importing Debian's `rust-coreutils`<sup>4</sup> package along with its  
260 dependencies.

261 The upstream Debian package isn't aimed at replacing `coreutils` yet, meaning  
262 some packaging changes were necessary to get Apertis images to build using  
263 `rust-coreutils` only:

- 264 • Change the package's priority from `optional` to `required`: this ensures  
265 `debootstrap` will pick up this package when bootstrapping the system
- 266 • Add `Conflicts/Breaks/Provides/Replaces` relationships so all dependencies  
267 are satisfied and conflicting packages (such as `coreutils-gplv2`) cannot be  
268 installed
- 269 • Install the binaries to `/bin` and `/usr/bin` instead of `/usr/libexec/rust-`  
270 `coreutils`

271 Additionally, a few patches were necessary to implement missing command-line  
272 options and most of them are in the process of being upstreamed. Some of our  
273 patches still require more work, especially regarding SELinux-related options:  
274 for now, we made sure the corresponding command-line options would be recog-  
275 nized, but the associated behavior isn't implemented yet. Since SELinux is not  
276 used in Apertis this was enough to ensure images could be built successfully,  
277 but these patches are not suitable for upstreaming. Implementing full SELinux  
278 support is not currently in the scope of Apertis, maintaining the downstream  
279 patches until upstream implements proper support is not going to be particu-  
280 larly problematic.

281 `coreutils-gplv2` have been consequently removed from Apertis `v2022dev3` and  
282 later repository.

---

<sup>4</sup><https://tracker.debian.org/pkg/rust-coreutils>