



GNU coreutils

Red Hat

Ondřej Vašík and Kamil Dudka

2009-09-11

Abstract

Set of core utilities for shell script usage and command line operations. At the moment there is approximately 100 utilities and the number is still rising.

History of GNU coreutils

Project started in **September 2002** by merging of **Fileutils**, **Shellutils** and **Textutils** projects.

fileutils		
cp	mv	ls
rm	install	
du	dd	df
mkdir	shred	

shellutils	
date	echo
pwd	sleep
id	printf
tee	whoami

textutils		
cat	tac	tr
join	sort	
wc	md5sum	
head	uniq	



coreutils
all 108 utilities together

GNU coreutils and gnuilib

GNU coreutils tarball consists from two separate projects:

1 GNU coreutils

- higher level source code of common utilities
- <http://www.gnu.org/software/coreutils>

2 gnuilib

- GNU portability library
- used by other GNU projects as well (e.g. findutils)
- <http://www.gnu.org/software/gnuilib>



Section 1

Development of GNU coreutils

Basic characteristics of the coreutils project

- **Conservative**
 - some things kept because of historical reasons
 - hard to justify new options in common utilities
- **Strict**
 - solution has to be as portable as possible
 - otherwise very likely rejected
- **Progressive**
 - 1st project to require automake 1.11 for build of stable release
 - 1st project to distribute tarballs in lzma format (and now in xz) instead of bzip2

Development of GNU coreutils

- **Maintainers**
 - Jim Meyering, Paul Eggert, Pádraig Brady, Eric Blake, ...
 - relatively small amount of contributors (so far only 65)
 - only 7 contributors with more than 20 commits
- **Finished?**
 - **No!** Approx. 30 commits per month...

Opened tasks

- better xattrs, ACL and SELinux support
- multibyte (i18n) support
- documentation improvement
- performance improvement
- memory consumption improvement
- new features
- few segfaults, nothing is perfect
- obscure date formats (simple Bison parser resolves date)
- obscure situations/options are not always handled correctly



Section 2

Development of gnulib

Development of gnulib

- **Maintainers**
 - Jim Meyering, Bruno Haible, Paul Eggert, Karl Berry, Simon Josefsson, Eric Blake, ...
 - relatively small amount of contributors (so far only 78)
 - only 15 contributors with more than 20 commits
- **Still very active**
 - approx. 100 commits per month
 - new modules, factorization
 - performance improvements, portability enhancements, ...



Section 3

Common mistakes

Man pages and info documentation

1 Man pages

- generated from `--help` output at build time
- intended to be compact

2 Info documentation

- many corner cases, examples and details
- often helpful for getting correct and expected result
- also available online:
`http://www.gnu.org/software/coreutils/manual`

Locales

- should be used only when necessary
- can be turned off by using `LC_ALL=C`
- affects e.g. sorting order of the `sort` utility
- good to try various locales before reporting a bug
- using locales can lead to lower performance (up to 20×)
- using locales can trigger a bug (not known to upstream)

Not a bug of coreutils

- **Shell built-ins**

- shell may have its own version of certain utilities (test, kill, ...)
- shipped with coreutils mostly for better portability
- you can specify full path to avoid use of the built-in:
`/usr/bin/[`

- **Utilities from another packages**

- some commonly used shell script utilities are not in coreutils
- they could be either in util-linux-ng or in separate packages (findutils, sed, grep, ...)

Other FAQ

- How do I change the ownership or permissions of a symlink?
- I used `rm` to remove a file. How can I get it back now?
- `ls -R *.py` does not work...
- `expr 2 * 3` does not work...

- and more at
<http://www.gnu.org/software/coreutils/faq/coreutils-faq.html>



Section 4

How to contribute to GNU coreutils

Step 1 - discuss intended contribution

- **check email archives**
 - read previous discussion about similar topics
 - it is quite hard to get new options/features to “core” utilities like ls, cp, ...
- **write to bug-coreutils@gnu.org or bug-gnulib@gnu.org**
 - non-trivial contributions should be announced before working on it
 - it is required to sign Free Software Foundation papers before submitting patches

Step 2 - write a patch

Once your idea is accepted you are ready to write a patch:

1 Download the appropriate git tree:

```
git clone git://git.sv.gnu.org/coreutils
git clone git://git.sv.gnu.org/gnulib
```

2 Read HACKING file

- very good how-to for first commits
- be prepared for amending your patch
(larger patches usually needs a few amendments before they get accepted into upstream git tree)

New features

New features need to:

- Follow POSIX rules:

`http://www.opengroup.org/onlinepubs/009695399/utilities/<utility>.html`

- Be portable/should not break portability
- Not be already provided by other utilities or easily done by few basic commands connected via pipes
- Patch should have a test case coverage (if possible, to prevent regressions)



Section 5

GNU coreutils options

GNU coreutils options

- if `POSIXLY_CORRECT` environment variable is set, options must appear before operands
- short/long options
- long options abbreviation
- module `getopt` from gnuilib

Standard options

- `--help`
- `--version`
- `--`

EXAMPLE: How to remove a file named '-v'?

```
$ rm -v
rm: missing operand
Try 'rm --help' for more information.
```

```
$ rm ./-v
$ rm -- -v
```

Backup options

- `--backup=mode`
 - simple/numbered, can be selected automatically (`-b`)
 - default can be set by the `VERSION_CONTROL` environment variable

EXAMPLE: Numbered backups with `cp`

```
$ cp -v --backup=numbered DIR_COLORS /etc/DIR_COLORS
'DIR_COLORS' -> '/etc/DIR_COLORS' (backup:
'/etc/DIR_COLORS.~1~')
```

```
$ cp -v --backup=numbered DIR_COLORS /etc/DIR_COLORS
'DIR_COLORS' -> '/etc/DIR_COLORS' (backup:
'/etc/DIR_COLORS.~2~')
```

Target options

- `-t`, `--target-directory=dir`
- `-T`, `--no-target-directory`

EXAMPLE: `--target-directory`

```
$ find /etc -name \*.conf \  
    | xargs --replace=ARG cp -v ARG .
```

```
$ find /etc -name \*.conf \  
    | xargs cp -v --target-directory=.
```

Target options

EXAMPLE: `--no-target-directory`

```
$ ln -s /etc/httpd/conf.d
$ stat -c%N *
'conf.d' -> '/etc/httpd/conf.d'

$ ln -sf /etc/fonts/conf.d conf.d
ln: creating symbolic link 'conf.d/conf.d':
Permission denied

$ ln -sf --no-target-directory /etc/fonts/conf.d \
    conf.d

$ stat -c%N *
'conf.d' -> '/etc/fonts/conf.d'
```


Reference option

- `--reference=REF_FILE`
- `chgrp`, `chmod`, `chown` and `chcon`
- `touch` and `date` (file's mtime)
- `truncate` (file's size)

Traversing symlinks

`[-H|-L|-P]`

- `-H` with `--recursive (-R)` traverse each symlink given as command line argument
- `-L` with `--recursive (-R)` traverse each encountered symlink to a directory
- `-P` `[default]` do not traverse any symlinks

Protection against user

EXAMPLE: Removing /

```
$ rm -fr /  
rm: cannot remove root directory '/'
```

- `--preserve-root`
 - default for `rm` to avoid typo like `'rm -fr / tmp/junk'`
 - can be used in aliases for `chmod`, `chown` and `chgrp`
- `--no-preserve-root`
 - default for `chmod`, `chown` and `chgrp`
 - can be used to override default behavior of `rm`



Section 6

GNU coreutils – examples

ls

- `ls -l`
- `ls -l --time-style=full-iso`
- `.` following the file mode bits \implies SELinux, but nothing else
- `+` following the file mode bits \implies alternate access method
- `ls -v`

EXAMPLE: `ls -v`

```
$ ls -l
vmlinuz-2.6.9-5.0.3.EL
vmlinuz-2.6.9-5.0.5.EL
vmlinuz-2.6.9-5.EL
vmlinuz-2.6.9-34.0.1.EL
vmlinuz-2.6.9-34.EL
```

```
$ ls -lv
vmlinuz-2.6.9-5.EL
vmlinuz-2.6.9-5.0.3.EL
vmlinuz-2.6.9-5.0.5.EL
vmlinuz-2.6.9-34.EL
vmlinuz-2.6.9-34.0.1.EL
```

stat

- various information about file (not available in `ls`)
- based on `stat(2)` and `stat` system call respectively
- `stat --printf=...`
- `find --printf=...` [not part of coreutils]

EXAMPLE: `stat --printf`

```
$ stat --printf="%d\t%i\t%N\n" *
```

13	5225	'controlC0'
13	5218	'hwCOD0'
13	5211	'hwCOD1'
13	9553	'seq'
13	5027	'timer'

basename, dirname

- handy when working with path names
- easy suffix cutting

EXAMPLE: basename, dirname

```
$ basename /etc/resolv.conf  
resolv.conf
```

```
$ basename /etc/resolv.conf .conf  
resolv
```

```
$ dirname /etc/resolv.conf  
/etc
```

readlink

- displays value of a symbolic link

EXAMPLE: readlink

```
$ cd /usr/lib
$ stat -c%N libcrypt.so
'libcrypt.so' -> '../..../lib/libcrypt.so.1'

$ readlink libcrypt.so
../..../lib/libcrypt.so.1
```


readlink

- How to obtain (canonicalized) absolute path of a file?
(from e.g. ../../lib/libcrypt.so.1)

```
readlink -e|-f|-m          (--canonicalize...)
```

EXAMPLE: readlink -f

```
$ readlink -f libcrypt.so  
/lib/libcrypt-2.10.1.so
```

```
$ readlink -f ../include  
/usr/include
```

cp

- `cp -a` (`--archive`)
- `cp --preserve=xattr,...`
- `cp --no-preserve=mode,...`

- `cp --sparse=always`
- `cp --sparse=never`

- `cp --one-filesystem`

- `cp -n` (`--no-clobber`)
- `yes | cp ...`

mkdir, rmdir

- `mkdir -p` (`--parents`)
- `rmdir -p` (`--parents`)

EXAMPLE: mkdir

```
$ mkdir a
$ mkdir a/b
$ mkdir a/b/c
```

EXAMPLE: mkdir -p, rmdir -p

```
$ mkdir -p a/b/c
$ rmdir -p a/b/c
```

install

- used to copy files and set attributes
- does **not** install RPM (neither DEB) packages!
- `install -d` (`--directory`)
- `install -Z` (`--context`)
(useful to boost performance of 'make install')

EXAMPLE: `install -d`

```
$ install -m 0700 -o kdudka -g root -d /home/kdudka
$ stat --printf="%A %U %G %n\n" /home/kdudka
drwx----- kdudka root /home/kdudka
```

dd

- does **not** use getopt module for options (an exception)
- `dd conv=ucase` `(tr [:lower:] [:upper:])`
- `dd iflag=fullblock`
- displays current I/O statistics on SIGUSR1

EXAMPLE: dd and SIGUSR1

```
$ dd if=/dev/zero of=./vm.img bs=1024 count=4194304 &  
[1] 29422
```

```
$ /bin/kill -s SIGUSR1 $!  
298309+0 records in  
298309+0 records out  
305468416 bytes (305 MB) copied, 5,41902 s, 56,4 MB/s
```

split, truncate

EXAMPLE: split

```
$ split --bytes=1G --numeric-suffixes --verbose \  
    vm.img vm.img.
```

```
creating file 'vm.img.00'  
creating file 'vm.img.01'  
...
```

EXAMPLE: truncate

```
$ truncate --reference=vm.img.00 vm.img
```

Hanging on a dead NFS mount point

- `alias ls='ls --color=auto'`
- `ls` can get stuck on symlink colorizing
- `eval 'dircolors | sed s/or=[^:]*:/or=0:/^'`
- `df --local`
- `timeout`
- `timeout --signal=...`

sort

- `sort`
- `sort --key=...`
- `sort --unique` (!= `sort|uniq`)
- `nl`
- `sort --human-numeric-sort`

EXAMPLE: `sort --human-numeric-sort`

```
$ du -sh `ls -A` | sort -rh | head -n 5
23G      vm
3.9G     git
3.1G     svn
867M     .ccache
865M     cvs
```


cut, fold

```
$ dmesg | tail -n 1  
[2116891.624879] <0>Kernel panic - not syncing: Fatal excep
```

EXAMPLE: cut

```
$ dmesg | tail -n 1 | cut -c -50  
[2116891.624879] <0>Kernel panic - not syncing: Fa
```

EXAMPLE: fold

```
$ dmesg | tail -n 1 | fold --spaces --width 50  
[2116891.624879] <0>Kernel panic - not  
syncing: Fatal exception: panic_on_oops
```

cut, tr

```
$ cut --delimiter='#' --fields=1 /etc/fstab \  
    | tr --squeeze-repeats [:space:] \  
    | cut --delimiter=' ' --fields=2
```

```
/\  
/home\  
swap\  
/dev/shm\  
/dev/pts\  
/sys\  
/proc\  
/mnt/mirror
```

printf, od, seq, tee, tail -f

- `printf` – shell equivalent of `printf(3)`
- `od` (Octal Dump)
- `od -x`

- `seq`
- `seq --equal-width 1000 | xargs touch`

- `tee`
- `tee --append`

- `tail -f` (not `tailf`)

wc, expand, unexpand, tac

- `wc` (Word Count)
- `wc --lines`
- `wc --words`
- `wc --chars`

- `expand --tabs=...`
- `unexpand --tabs=...`

- `tac` – prints lines in reverse order

md5sum, sha*sum, base64

- `md5sum`
- `md5sum --check`

- `sha1sum --check`
- `sha224sum --check`
- `sha256sum --check`
- `sha384sum --check`
- `sha512sum --check`

- `base64`
- `base64 --decode`

id, who, uptime, uname

- id
- id --group
- id --context

- who
- who am i (who mum likes)

- uptime
- uname – only wrapper around uname(2)

stty, shred

```
# save terminal settings
stty --file /dev/ttyS0 --save \
    | tee ttyS0-settings.txt \
    | fold --width=43
500:5:bf:8a3b:3:1c:7f:15:4:0:1:0:11:13:1a:0
:12:f:17:16:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0
# restore terminal settings
stty --file /dev/ttyS0 \
    `cat ttyS0-settings.txt`
```

EXAMPLE: shred

```
$ shred ~/.fedora.cert
```