

## 1. The gzip utility:

### 1. What URL can you download it from?

<http://ftp.gnu.org/gnu/gzip/>

**What's the latest version?**

gzip-1.8

**What version is currently on chuck?**

gzip 1.5

**What would be an advantage of downloading and installing it yourself?**

- The older version is present on chuck. To use the features of the latest version, now we can use the version installed.
- I got to learn what it is, what are all the version, etc.

### 2. List a couple of differences in the log of changes (hint) between the system version and the one you've downloaded.

Below is the list of dates, the person committing change and the changes:

- |        |            |               |   |
|--------|------------|---------------|---|
| i.     | 2016-04-20 | Paul Eggert   | gzip: fix bug with -l output to pipes       |
| ii.    | 2016-03-28 | Paul Eggert   | Port to Oracle Solaris Studio 12.4          |
| iii.   | 2016-03-26 | Paul Eggert   | Port to NetBSD 7.0                          |
| iv.    | 2016-03-26 | Paul Eggert   | * tests/init.sh: Sync from GnuLib.          |
| v.     | 2016-03-26 | Paul Eggert   | Port to Alpine Linux which uses Busybox     |
| vi.    | 2016-03-21 | Paul Eggert   | Fix typo in previous patch                  |
| vii.   | 2016-03-07 | Paul Eggert   | gzip: pacify clang                          |
| viii.  | 2016-03-06 | Paul Eggert   | gzip: remove --__bindir                     |
| ix.    | 2016-03-02 | Rusty Russell | gzip: support the --rsyncable option        |
| x.     | 2016-02-23 | Paul Eggert   | fsync output file before closing            |
| xi.    | 2015-03-17 | Paul Eggert   | gzip: make the GZIP env var obsolescent     |
| xii.   | 2014-11-10 | Jim Meyering  | maint: enable more syntax checks            |
| xiii.  | 2014-11-10 | Jim Meyering  | gnulib+bootstrap: update to latest          |
| xiv.   | 2014-10-10 | Paul Eggert   | tests: use local dir for output             |
| xv.    | 2014-08-10 | Paul Eggert   | gzip: fix --suffix=z bug (Bug#18239)        |
| xvi.   | 2013-10-04 | Paul Eggert   | znew: avoid denial-of-service issue         |
| xvii.  | 2013-06-09 | Jim Meyering  | maint: post-release administrivia           |
| xviii. | 2013-05-28 | Jim Meyering  | tests: exercise the new --keep option       |
| xix.   | 2012-10-24 | Paul Eggert   | tests: exercise the grep -e portability fix |
| xx.    | 2012-08-15 | Paul Eggert   | zgrep: do not assume GNU expr               |
| xxi.   | 2012-06-17 | Jim Meyering  | maint: add .mailmap                         |

### 3. How do you find out how to compile it - which file(s) should you read? Why in CAPS?

INSTALL contains the direction on installing and compiling the file  
Because in small lettere there is an existing command

### 4. What happens if you just type 'make' and skip the first step?

The first step is responsible for creating the Makefile, without the same make command won't be able to find the file and thus make and install commands fail to do anything.

**Why do we need the './' in the first step?**

The same runs the script name configure in the directory, which makes sure that the computer configurations are proper for the software installation

### 5. What are the typical configuration, compilation and installation steps?

./configure --prefix=\$HOME/5621

make  
make check  
make install

**6. Where would it try to install by default?**

Home directory, or the directory in which it's been downloaded

**Why won't that work for you on chuck?**

That would require write access on the same

**How do you change the installation prefix to somewhere in your homedir?**

--prefix=/HOME/5621

**Why is setting the prefix to just \$HOME a bad idea; what prefix did you use?**

The would require access and if downloaded then be alterable by all

**7. How long does the compilation step take?**

Less than half minute

**If you do a 'make clean' followed by 'make -j 2', how does that impact on the time?**

**Why?**

make clean removes the compiled files created by Makefile.

make -j 2 tells the number of jobs to be run at one time. Here, we are directing it to run 2 commands at a time and thus making the process serialized.

**8. Are there any self-test targets available? Does it pass?**

Yes, there are 18 such tests(gzip-env, helin-segv, help-version, hufts,  
keep, list, memcpy-abuse,mixed, null-suffix-clobber, stdin, trailing-nul,  
unpack-invalid, z-suffix,zdiff, zgrep-f,zgrep-context, zgrep-signal, znew-k).  
It passes all of them.

**2. The wget utility:**

**1. What URL can you download it from?**

<https://ftp.gnu.org/gnu/wget/>

**What's the latest version?**

Wget-1.19

**What version is currently on chuck?**

Wget-1.15

**2. What are the compilation steps?**

./configure --prefix=\$HOME/5621

make

make check

make install

**3. Why does the first step fail? How would you fix it?**

Missing gnutls package

./configure --prefix=\$HOME/5621 --with-ssl={gnutls,openssl}

**4. Does it pass the self-tests?**

Out of total 85 tests, it passed 71 and skipped 14

**What command did you use?**

make -k

**5. How long does the compilation step take?**

~2 minutes

**If you do a 'make clean' followed by 'make -j 2', how does that impact on the time? Why?**

The time was reduced.

make -j 2 tells the number of jobs to be run at one time. Here, we are directing it to run 2 commands at a time and thus making the process serialized

**6. What prefix did you set?**

--prefix=\$HOME/5621

**How would you make sure that you use the custom wget utility instead of the system one?**

Access the same from the /bin

cd bin

./gzip --version

**3. The openssl encryption suite:**

**1. What URL can you download it from?**

<https://www.openssl.org/source/>

**What's the latest version in the 1.0.1 branch?**

1.0.2

**What version is currently on chuck?**

OpenSSL 1.0.2k-fips

**2. What are the compilation steps?**

./config --prefix=\$HOME/5621

\$ make

\$ make test

\$ make install

**Are there any differences to the GNU software?**

Shared library is updated

**3. How long does the compilation step take?**

~1 minute

**If you do a 'make clean' followed by 'make -j 2', does that impact on the time? Why not? (hint: any warnings at the top of the make output?)**

No, because, the version change is from 1.0.1 to 1.0.2

**4. What is the latest version?**

OpenSSL 1.1.1

**Does 'make -j 2' make a difference to compiling for this version?**

~5 minutes, yes time was reduced