

Name: Priyanka, Student-id: 20179, Week-5➔HW1(Install command line bitcoin client to access bitcoin)

Installing Git on Ubuntu

Sudo apt install git

```
/root/.hushlogin file.  
root@LAPTOP-ORPFG98H:~# sudo apt install git  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
git is already the newest version (1:2.34.1-1ubuntu1.11).  
git set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
root@LAPTOP-ORPFG98H:~# |
```

Create Clone of the source code

<https://github.com/bitcoin/bitcoin.git>

```

0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@LAPTOP-ORPFG98H:~# git clone https://github.com/bitcoin/bitcoin.git
Cloning into 'bitcoin'...
remote: Enumerating objects: 289722, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 289722 (delta 8), reused 13 (delta 3), pack-reused 289691 (from 1)
Receiving objects: 100% (289722/289722), 263.47 MiB | 2.61 MiB/s, done.
Resolving deltas: 100% (209261/209261), done.
root@LAPTOP-ORPFG98H:~# cd bitcoin/
root@LAPTOP-ORPFG98H:~/bitcoin# ls -lrt
total 112
drwxr-xr-x  4 root root  4096 Oct 15 13:32 cmake
drwxr-xr-x  5 root root  4096 Oct 15 13:32 ci
-rw-r--r--  1 root root   871 Oct 15 13:32 SECURITY.md
-rw-r--r--  1 root root  3438 Oct 15 13:32 README.md
-rw-r--r--  1 root root    27 Oct 15 13:32 INSTALL.md
-rw-r--r--  1 root root  1142 Oct 15 13:32 COPYING
-rw-r--r--  1 root root 21610 Oct 15 13:32 CONTRIBUTING.md
-rw-r--r--  1 root root  3048 Oct 15 13:32 CMakePresets.json
-rw-r--r--  1 root root 26531 Oct 15 13:32 CMakeLists.txt
drwxr-xr-x  6 root root  4096 Oct 15 13:32 depends
drwxr-xr-x 21 root root  4096 Oct 15 13:32 contrib
-rw-r--r--  1 root root   377 Oct 15 13:32 libbitcoinkernel.pc.in
drwxr-xr-x  6 root root  4096 Oct 15 13:32 doc
drwxr-xr-x  6 root root  4096 Oct 15 13:32 share
drwxr-xr-x 30 root root  4096 Oct 15 13:32 src
drwxr-xr-x  7 root root  4096 Oct 15 13:32 test
-rw-r--r--  1 root root  1294 Oct 15 13:32 vcpkg.json

```

Git checkout

git checkout v28.0

Git status

```
1711111100c100c1294 Oct 19 19:52 vcpkg.json
root@LAPTOP-ORPFG98H:~/bitcoin# git checkout v0.19.0
Note: switching to 'v0.19.0'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable `advice.detachedHead` to false

```
HEAD is now at 42414bfd05 build: set CLIENT_VERSION_RC to 0 pre-final
```

```
root@LAPTOP-ORPFG98H:~/bitcoin# git status
```

```
HEAD detached at v0.19.0
```

```
nothing to commit, working tree clean
```

```
sudo apt install autoconf autogen libtool
```

```

nothing to commit, working tree clean
root@LAPTOP-ORPFG98H:~/bitcoin# sudo apt install autoconf autogen libtool
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  autogen-doc automake autotools-dev guile-2.2-libs libgc1 libltdl-dev libltdl7 libopts25 libopts25-dev
Suggested packages:
  autoconf-archive gnu-standards autoconf-doc gettext libtool-doc gfortran | fortran95-compiler gcj-jdk
The following NEW packages will be installed:
  autoconf autogen autogen-doc automake autotools-dev guile-2.2-libs libgc1 libltdl-dev libltdl7 libop
0 upgraded, 13 newly installed, 0 to remove and 2 not upgraded.
Need to get 9133 kB of archives.
After this operation, 57.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 m4 amd64 1.4.18-5ubuntu2 [199 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 autoconf all 2.71-2 [338 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 libgc1 amd64 1:8.0.6-1.1build1 [96.8 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 libltdl7 amd64 2.4.6-15build2 [39.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 guile-2.2-libs amd64 2.2.7+1-6build2 [5810 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libopts25 amd64 1:5.18.16-4 [59.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libopts25-dev amd64 1:5.18.16-4 [95.9 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/universe amd64 autogen amd64 1:5.18.16-4 [558 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 autogen-doc all 1:5.18.16-4 [1000 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/main amd64 autotools-dev all 20220109.1 [44.9 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/main amd64 automake all 1:1.16.5-1.3 [558 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libltdl-dev amd64 2.4.6-15build2 [169 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/main amd64 libtool all 2.4.6-15build2 [164 kB]
Fetched 9133 kB in 6s (1654 kB/s)
Selecting previously unselected package m4.

```

Prepare the Build Environment

./autogen.sh

```

Processing triggers for install-info (0.8-4build1) ...
root@LAPTOP-ORPFG98H:~/bitcoin# ./autogen.sh
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'build-aux'.
libtoolize: copying file 'build-aux/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'build-aux/m4'.
libtoolize: copying file 'build-aux/m4/libtool.m4'
libtoolize: copying file 'build-aux/m4/ltoptions.m4'
libtoolize: copying file 'build-aux/m4/ltsugar.m4'
libtoolize: copying file 'build-aux/m4/ltversion.m4'
libtoolize: copying file 'build-aux/m4/lt~obsolete.m4'
configure.ac:40: installing 'build-aux/ar-lib'
configure.ac:35: installing 'build-aux/compile'
configure.ac:24: installing 'build-aux/config.guess'
configure.ac:24: installing 'build-aux/config.sub'
configure.ac:30: installing 'build-aux/install-sh'
configure.ac:30: installing 'build-aux/missing'
Makefile.am: installing 'build-aux/depcomp'
parallel-tests: installing 'build-aux/test-driver'
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, 'build-aux'.
libtoolize: copying file 'build-aux/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'build-aux/m4'.
libtoolize: copying file 'build-aux/m4/libtool.m4'
libtoolize: copying file 'build-aux/m4/ltoptions.m4'
libtoolize: copying file 'build-aux/m4/ltsugar.m4'
libtoolize: copying file 'build-aux/m4/ltversion.m4'
libtoolize: copying file 'build-aux/m4/lt~obsolete.m4'
configure.ac:116: installing 'build-aux/compile'
configure.ac:51: installing 'build-aux/config.guess'
configure.ac:51: installing 'build-aux/config.sub'
configure.ac:58: installing 'build-aux/install-sh'
configure.ac:58: installing 'build-aux/missing'
src/Makefile.am: installing 'build-aux/depcomp'
parallel-tests: installing 'build-aux/test-driver'
root@LAPTOP-ORPFG98H:~/bitcoin# ./configure
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes

```

Configure the Build

./configure

```

11  libevent-dev 2.1.12-stable-1build3
root@LAPTOP-ORPFG98H:~/bitcoin# ./configure
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking whether make supports nested variables... (cached) yes
checking for g++... g++
checking whether the C++ compiler works... yes
checking for C++ compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C++... yes
checking whether g++ accepts -g... yes
checking for g++ option to enable C++11 features... none needed
checking whether make supports the include directive... yes (GNU style)
checking dependency style of g++... gcc3
checking whether g++ supports C++17 features with -std=c++17... yes
checking whether std::filesystem can be used without link library... yes
checking whether the compiler supports GNU Objective C++... no
checking whether g++ -std=c++17 accepts -g... no
checking dependency style of g++ -std=c++17... gcc3
checking how to print strings... printf
checking for gcc... gcc
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether gcc understands -c and -o together... yes
checking dependency style of gcc... gcc3
checking for a sed that does not truncate output... /usr/bin/sed

```

\$./configure CPPFLAGS="-I\${BDB_PREFIX}/include/ -O2" LDFLAGS="-L\${BDB_PREFIX}/lib/" --with-gui

```

root@LAPTOP-ORPFG98H:~/bitcoin# export BDB_PREFIX=/usr
root@LAPTOP-ORPFG98H:~/bitcoin# ./configure CPPFLAGS="-I${BDB_PREFIX}/include/ -O2" LDFLAGS="-L${BDB_PREFIX}/lib/" --with-gui
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether to enable maintainer-specific portions of Makefiles... yes
checking whether make supports nested variables... (cached) yes
checking for g++... g++
checking whether the C++ compiler works... yes
checking for C++ compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C++... yes
checking whether g++ accepts -g... yes
checking for g++ option to enable C++11 features... none needed
checking whether make supports the include directive... yes (GNU style)
checking dependency style of g++... gcc3
checking whether g++ supports C++11 features with -std=c++11... yes
checking whether std::atomic can be used without link library... yes
checking whether the compiler supports GNU Objective C++... no
checking whether g++ -std=c++11 accepts -g... no
checking dependency style of g++ -std=c++11... gcc3
checking how to print strings... printf
checking for gcc... gcc
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether gcc understands -c and -o together... yes

```

Compile the Source Code

“make” command


```
root@LAPTOP-ORPFG98H:~/bitcoin# make
Making all in src
make[1]: Entering directory '/root/bitcoin/src'
make[2]: Entering directory '/root/bitcoin/src'
CXX      bitcoind-bitcoind.o
CXX      init/bitcoind-bitcoind.o
CXX      libbitcoin_node_a-addrdb.o
CXX      libbitcoin_node_a-addrman.o
CXX      libbitcoin_node_a-banman.o
CXX      libbitcoin_node_a-blockencodings.o
CXX      libbitcoin_node_a-blockfilter.o
CXX      libbitcoin_node_a-chain.o
CXX      consensus/libbitcoin_node_a-tx_verify.o
CXX      libbitcoin_node_a-dbwrapper.o
CXX      libbitcoin_node_a-deploymentstatus.o
CXX      libbitcoin_node_a-flatfile.o
CXX      libbitcoin_node_a-headerssync.o
CXX      libbitcoin_node_a-httprpc.o
CXX      libbitcoin_node_a-httpserver.o
CXX      libbitcoin_node_a-i2p.o
CXX      index/libbitcoin_node_a-base.o
CXX      index/libbitcoin_node_a-blockfilterindex.o
CXX      index/libbitcoin_node_a-coinstatsindex.o
CXX      index/libbitcoin_node_a-txindex.o
CXX      libbitcoin_node_a-init.o
CXX      kernel/libbitcoin_node_a-chain.o
CXX      kernel/libbitcoin_node_a-checks.o
CXX      kernel/libbitcoin_node_a-coinstats.o
CXX      kernel/libbitcoin_node_a-context.o
CXX      kernel/libbitcoin_node_a-mempool_persist.o
CXX      libbitcoin_node_a-mapport.o
CXX      libbitcoin_node_a-net.o
CXX      libbitcoin_node_a-net_processing.o
CXX      libbitcoin_node_a-netgroup.o
CXX      node/libbitcoin_node_a-blockstorage.o
CXX      node/libbitcoin_node_a-caches.o
CXX      node/libbitcoin_node_a-chainstate.o
CXX      node/libbitcoin_node_a-coin.o
CXX      node/libbitcoin_node_a-connection_types.o
```



```
CXXLD    libbitcoinconsensus_la_scriptencodings.o
make[2]: Leaving directory '/root/bitcoin/src'
make[1]: Leaving directory '/root/bitcoin/src'
Making all in doc/man
make[1]: Entering directory '/root/bitcoin/doc/man'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/root/bitcoin/doc/man'
make[1]: Entering directory '/root/bitcoin'
make[1]: Nothing to be done for 'all-am'.
make[1]: Leaving directory '/root/bitcoin'
root@LAPTOP-ORPFG98H:~/bitcoin#
```

Install bitcoind

Sudo make install

```

root@LAPTOP-ORPFG98H:~/bitcoin# sudo make install
Making install in src
make[1]: Entering directory '/root/bitcoin/src'
make[2]: Entering directory '/root/bitcoin/src'
make[3]: Entering directory '/root/bitcoin/src'
/usr/bin/mkdir -p '/usr/local/lib'
/bin/bash ../libtool --preserve-dup-deps --mode=install /usr/bin/install -c libbitcoincons
libtool: install: /usr/bin/install -c .libs/libbitcoinconsensus.so.0.0.0 /usr/local/lib/libbit
libtool: install: (cd /usr/local/lib && { ln -s -f libbitcoinconsensus.so.0.0.0 libbitcoincons
coinconsensus.so.0.0.0 libbitcoinconsensus.so.0; }; })
libtool: install: (cd /usr/local/lib && { ln -s -f libbitcoinconsensus.so.0.0.0 libbitcoincons
consensus.so.0.0.0 libbitcoinconsensus.so; }; })
libtool: install: /usr/bin/install -c .libs/libbitcoinconsensus.lai /usr/local/lib/libbitcoinc
libtool: install: /usr/bin/install -c .libs/libbitcoinconsensus.a /usr/local/lib/libbitcoincon
libtool: install: chmod 644 /usr/local/lib/libbitcoinconsensus.a
libtool: install: ranlib /usr/local/lib/libbitcoinconsensus.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:
-----
Libraries have been installed in:
  /usr/local/lib

If you ever happen to want to link against installed libraries
in a given directory, LIBDIR, you must either use libtool, and
specify the full pathname of the library, or use the '-LLIBDIR'
flag during linking and do at least one of the following:
- add LIBDIR to the 'LD_LIBRARY_PATH' environment variable
  during execution
- add LIBDIR to the 'LD_RUN_PATH' environment variable
  during linking
- use the '-Wl,-rpath -Wl,LIBDIR' linker flag
- have your system administrator add LIBDIR to '/etc/ld.so.conf'

See any operating system documentation about shared libraries for
more information, such as the ld(1) and ld.so(8) manual pages.
-----
/usr/bin/mkdir -p '/usr/local/bin'
/bin/bash ../libtool --preserve-dup-deps --mode=install /usr/bin/install -c bitcoind bitcoi
tcoin '/usr/local/bin'

```

in order to check a certain block information,

Run Bitcoin Daemon

bitcoind -daemon

```

root@LAPTOP-ORPFG98H:~/bitcoin# bitcoind -daemon
Bitcoin Core starting
root@LAPTOP-ORPFG98H:~/bitcoin# |

```

To check the information with:

```
bitcoin-cli -getinfo
```

```
bitcoin-cli getblockchaininfo
```

```
root@LAPTOP-ORPFG98H:~/bitcoin# bitcoin-cli getinfo
error code: -32601
error message:
Method not found
root@LAPTOP-ORPFG98H:~/bitcoin# bitcoin-cli -getinfo
Chain: main
Blocks: 76288
Headers: 865851
Verification progress: ████████████████████████████████ 0.0112%
Difficulty: 511.7735342566043

Network: in 0, out 10, total 10
Version: 240001
Time offset (s): -1
Proxies: n/a
Min tx relay fee rate (BTC/kvB): 0.00001000

Warnings: (none)
```

```

root@LAPTOP-ORPFG98H:~/bitcoin# bitcoin-cli getblockchaininfo
{
  "chain": "main",
  "blocks": 128834,
  "headers": 865851,
  "bestblockhash": "00000000000002640e5ca9f089dd4ebabb7452da610ce97b94a94fedf041e8fba",
  "difficulty": 434877.0455276276,
  "time": 1307302191,
  "mediantime": 1307300009,
  "verificationprogress": 0.0006596643062990147,
  "initialblockdownload": true,
  "chainwork": "0000000000000000000000000000000000000000000000000000000000000000af2fabbb9b0b2f26",
  "size_on_disk": 248657447,
  "pruned": false,
  "warnings": ""
}
root@LAPTOP-ORPFG98H:~/bitcoin#

```

Now using bitcoin-cli,

we can get any block of bitcoin. –

First, we need to get the hash value of the block by running:

- bitcoin-cli getblockhash 76288

Once block hash value is obtained, you can get that block information

And i got the block hash:

```
000000000017233840390388190a00106321f39e6d2cc47dc0011adaefd011f3
```

Get Block Information Now, replace <block hash value> with the actual hash you received.

```
bitcoin-cli getblock
```

```
000000000017233840390388190a00106321f39e6d2cc47dc0011adaefd011f3
```

[illegible]

