

Bitcoin Daemon for CentOS 5.5

Michael Erwin Steurer
michael.steurer@iicm.tugraz.at
Institute for Information Systems and Computer Media
Graz University of Technology, Austria

April 26, 2011

1 Introduction

This manual bases on “Build notes for Linux type systems - tested on amd64”¹ by Laszlo Hanyecz with minor changes for Bitcoin version 0.3.21-beta. Further, we also include a detailed description for 32 bit operating systems.

We keep everything in our home directory under `~/Bitcoin/`. We do not make any changes to system parameters or system libraries and therefore no root privileges are required.

```
msteurer@sophie:~$ mkdir Bitcoin
msteurer@sophie:~$ cd Bitcoin
msteurer@sophie:Bitcoin$ mkdir Libraries
msteurer@sophie:Bitcoin$ mkdir Trunk
msteurer@sophie:Bitcoin$ mkdir Deps
```

2 Libraries

- Boost. Get the Library², put it into the directory `~/bitcoin/Libraries/` and execute the following commands (write the parameters for the `bjam` command in one line).

```
msteurer@sophie:Bitcoin$ cd Libararies
msteurer@sophie:Libraries$ tar -xzvf boost_1_46_1.tar.gz
msteurer@sophie:Libraries$ cd boost_1_46_1
msteurer@sophie:boost_1_46_1$ ./bootstrap.sh
msteurer@sophie:boost_1_46_1$ ./bjam --prefix=~/Bitcoin/Deps
                                link=static
                                runtime-link=static
                                install
```

- OpenSSL. Get the Library³, put it into the directory `~/bitcoin/Libraries/`, extract it and determine the system architecture.

```
msteurer@sophie:Libraries$ tar -xzvf openssl-1.0.0d.tar.gz
msteurer@sophie:Libraries$ cd openssl-1.0.0d
msteurer@sophie:openssl-1.0.0d$ uname -a
```

¹<http://heliacal.net/~solar/bitcoin/bitcoin-linuxbuild.pdf>

²http://sourceforge.net/projects/boost/files/boost/1.46.1/boost_1_46_1.tar.gz

³<http://www.openssl.org/source/openssl-1.0.0d.tar.gz>

If the output of the last command contains

```
x86_64 x86_64 x86_64 GNU/Linux
```

you have a 64-bit computer architecture and if it contains

```
i686 i686 i386 GNU/Linux
```

you have a 32-bit computer architecture.

Execute **one** of the following *./Configure* commands according to the architecture

– 64-bit architecture

```
msteurer@sophie:openssl-1.0.0d$ ./Configure --prefix=~/.Bitcoin/Deps
--openssldir=~/.Bitcoin/Deps/openssl
linux-x86_64
```

– 32-bit architecture

```
msteurer@sophie:openssl-1.0.0d$ ./Configure --prefix=~/.Bitcoin/Deps
--openssldir=~/.Bitcoin/Deps/openssl
linux-generic32
```

– Get a list with all supported architectures

```
msteurer@sophie:openssl-1.0.0d$ ./Configure --prefix=~/.Bitcoin/Deps
--openssldir=~/.Bitcoin/Deps/openssl
```

Finally, compile and install.

```
make && make install
```

- Berkeley DB. Get the Library⁴, put it into the directory `~/Bitcoin/Libraries/` and compile it.

```
msteurer@sophie:Libraries$ tar -xvzf db-5.1.19.tar.gz
msteurer@sophie:Libraries$ cd db-5.1.19
msteurer@sophie:db-5.1.19$ cd build_unix
msteurer@sophie:db-5.1.19$ ../dist/configure --prefix=~/.Bitcoin/Deps/ --enable-cxx
msteurer@sophie:db-5.1.19$ make && make install
```

3 Bitcoin

Get the Bitcoin sources from the webpage⁵ or check it out from the Git repository⁶ and put it into the directory `~/Bitcoin/Trunk/`

```
msteurer@sophie:Bitcoin$ git clone https://github.com/bitcoin/bitcoin.git Trunk
```

Create a new file `~/Bitcoin/Trunk/makefile.centos` and add the lines as follows.

⁴<http://http://freshmeat.net/projects/berkeleydb>

⁵<http://www.bitcoin.org/>

⁶<https://github.com/bitcoin/bitcoin>

```

#begin

# Copyright (c) 2009-2010 Satoshi Nakamoto
# Distributed under the MIT/X11 software license, see the accompanying
# file license.txt or http://www.opensource.org/licenses/mit-license.php.

CXX=g++

DEPSDIR=/home/msteurer/Bitcoin/Deps

INCLUDEPATHS= -I"${DEPSDIR}/include"
LIBPATHS= -L"${DEPSDIR}/lib"

WXINCLUDEPATHS=$(shell wx-config --cxxflags)
WXLIBS=$(shell wx-config --libs)

USE_UPNP=

DEFS= -DNOPCH -DFOURWAYSSE2 -DUSE_SSL

LIBS= -dead_strip \
      -Wl,-Bstatic \
      ${DEPSDIR}/lib/libdb_cxx-5.1.a \
      -mtl ${DEPSDIR}/lib/libboost_system.a \
      -mtl ${DEPSDIR}/lib/libboost_thread.a \
      -mtl ${DEPSDIR}/lib/libboost_filesystem.a \
      -mtl ${DEPSDIR}/lib/libboost_program_options.a \
      ${DEPSDIR}/lib/libdb_cxx.a \
      ${DEPSDIR}/lib/libssl.a \
      ${DEPSDIR}/lib/libcrypto.a \
      -l ssl \
      -l crypto \
      -Wl,-Bdynamic \
      -l gthread-2.0 \
      -l z \
      -l dl

ifdef USE_UPNP
  LIBS += -l miniupnpc
  DEFS += -DUSE_UPNP=${USE_UPNP}
endif

DEBUGFLAGS=-g -D__WXDEBUG__
CXXFLAGS=-O2 -Wno-invalid-offsetof -Wformat ${DEBUGFLAGS} ${DEFS} ${INCLUDEPATHS}
HEADERS=headers.h strlcpy.h serialize.h uint256.h \
          util.h key.h bignum.h base58.h script.h \
          db.h net.h irc.h main.h rpc.h uibase.h \
          ui.h noui.h init.h

OBJS= obj/util.o \
      obj/script.o \
      obj/db.o \
      obj/net.o \
      obj/irc.o \
      obj/main.o \
      obj/rpc.o \

```

```

    obj/init.o \
    cryptopp/obj/sha.o \
    cryptopp/obj/cpu.o

all: bitcoind

obj/%.o: %.cpp $(HEADERS)
    $(CXX) -c $(CXXFLAGS) $(WXINCLUDEPATHS) -DGUI -o $@ $<

cryptopp/obj/%.o: cryptopp/%.cpp
    $(CXX) -c $(CXXFLAGS) -O3 -o $@ $<

obj/sha256.o: sha256.cpp
    $(CXX) -c $(CXXFLAGS) -msse2 -O3 -march=amdfam10 -o $@ $<

bitcoin: $(OBJS) obj/ui.o obj/uibase.o obj/sha256.o
    $(CXX) $(CXXFLAGS) -o $@ $(LIBPATHS) $^ $(WXLIBS) $(LIBS)

obj/nogui/%.o: %.cpp $(HEADERS)
    $(CXX) -c $(CXXFLAGS) -o $@ $<

bitcoind: $(OBJS:obj/%=obj/nogui/%) obj/sha256.o
    $(CXX) $(CXXFLAGS) -o $@ $^ $(LIBS)

clean:
    -rm -f obj/*.o
    -rm -f obj/nogui/*.o
    -rm -f cryptopp/obj/*.o
    -rm -f headers.h.gch
    -rm -f bitcoin
    -rm -f bitcoind

#end

```

Change the Value *DEPSDIR* according to your *Deps/* directory. Keep in mind that are tabs instead of spaces in front of makefile-commands. Finally, compile the sources.

```
make -f makefile.centos bitcoind
```