Build notes for Linux type systems - tested on amd64
Laszlo Hanyecz (solar@heliacal.net)

It is assumed you know how to unpack the stuff after downloading, these are just the commands I used to build each thing on my 64 bit linux system.

I'm keeping everything in ~/bitcoin where ~ is /home/solar (solar being my unix id)
The dependencies are under ~/bitcoin/deps, the bitcoin source from svn is in ~/bitcoin/trunk

This process should be the same on i386 linux but you'll have to change a couple of small things when building the deps (should be obvious).

Boost
http://www.boost.org/users/download/

```
./bootstrap.sh
./bjam --prefix=/home/solar/bitcoin/deps link=static runtime-link=static install
```

OpenSSL
Download from http://www.openssl.org/source/

```
./Configure --prefix=/home/solar/bitcoin/deps --openssldir=/home/solar/bitcoin/deps/openssl linux-x86_64
make && make install
```

Berkeley DB
Download from http://freshmeat.net/projects/berkeleydb/

```
cd build_unix
../dist/configure --prefix=/home/solar/bitcoin/deps --enable-cxx
&& make && make install
```

wxWidgets
http://prdownloads.sourceforge.net/wxwindows/wxWidgets-2.9.0.tar.bz2

```
sudo apt-get install libgtk2.0-dev (for debian)

mkdir buildgtk
cd buildgtk
../configure --prefix=/home/solar/bitcoin/deps --with-gtk --
disable-shared --disable-debug_flag --enable-stl --enable-utf8
--with-libjpeg=builtin --with-libpng=builtin --with-
regex=builtin --with-libtiff=builtin --with-zlib=builtin --with-
expat=builtin
```

make && make install

Bitcoin

```
svn checkout https://bitcoin.svn.sourceforge.net/svnroot/
bitcoin/trunk
```

edit net.h and change line 15: 'htons(8333)' to 0x8d20

```
make -f makefile.unix
make -f makefile.unix bitcoind
```

My modified makefile.unix

```
# 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.

DEPSDIR=/home/solar/bitcoin/deps

INCLUDEPATHS= \
 -I"$(DEPSDIR)/include"

LIBPATHS= \
 -L"$(DEPSDIR)/lib"

WXLIBS=$(shell $(DEPSDIR)/bin/wx-config --libs --static)
WXBASELIBS=$(shell $(DEPSDIR)/bin/wx-config --libs base --
static)

LIBS= -dead_strip \
 $(DEPSDIR)/lib/libdb_cxx-4.8.a \
 $(DEPSDIR)/lib/libboost_system.a \
 $(DEPSDIR)/lib/libboost_filesystem.a \
 $(DEPSDIR)/lib64/libcrypto.a

WXDEFS=$(shell $(DEPSDIR)/bin/wx-config --cxxflags) -DNOPCH
DEBUGFLAGS=-g -DwxDEBUG_LEVEL=0
CFLAGS=-O2 -Wno-invalid-offsetof -Wformat $(DEBUGFLAGS) $
(WXDEFS) $(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 init.h
sha.h


all: bitcoin


headers.h.gch: headers.h $(HEADERS)
     g++ -c $(CFLAGS) -o $@ $<

obj/%.o: %.cpp $(HEADERS) headers.h.gch
     g++ -c $(CFLAGS) -o $@ $<

obj/sha.o: sha.cpp sha.h
     g++ -c $(CFLAGS) -O3 -o $@ $<

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

bitcoin: $(OBJS) obj/ui.o obj/uibase.o obj/sha.o
     g++ $(CFLAGS) -o $@ $(LIBPATHS) $^ $(WXLIBS) $(LIBS)


obj/nogui/%.o: %.cpp $(HEADERS)
     g++ -c $(CFLAGS) -DwxUSE_GUI=0 -o $@ $<

bitcoind: $(OBJS:obj/%=obj/nogui/%) obj/sha.o
     g++ $(CFLAGS) -o $@ $(LIBPATHS) $^ $(WXBASELIBS) $(LIBS)


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

# end
```