Installing OS/161 on Your Own Machine

1、You can get the complete version of this tutorial in

https://www.student.cs.uwaterloo.ca/~cs350/common/Install161NonCS.html

There are a few differences between those two. To be honest, I recommend this one, because it suits our class more.

2、Recommendations:

①OS Version: Ubuntu 14.04

②You had better operate in a virtual machine and record several snapshots。

3、NOTE:

①You had better use your own account on your machine instead of root.

②Those red words are some important information such as commands and filenames.

Step 1: Download

What	Download
Binutils for MIPS	http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-binutils.tar.gz
GCC MIPS Cross-Compiler	http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-gcc.tar.gz
GDB for Use with OS/161	http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-gdb.tar.gz
bmake for use with OS/161	http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-bmake.tar.gz
mk for use with OS/161	http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161-mk.tar.gz
sys/161	http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/sys161.tar.gz
OS/161	http://www.student.cs.uwaterloo.ca/~cs350/os161_repository/os161.tar.gz

NOTE: It may take some time to download some of these files and if you try to proceed (e.g., run tar before all of the file has been downloaded it will fail). Please check that your browser has finished downloading the file you are going to work with before working with it.

Step 2: Build and Install the Binary Utilities (Binutils)

Unpack the binutils archive:

```
cd ~
tar -xzf os161-binutils.tar.gz
```

Move into the newly-created directory:

```
cd binutils-2.17+os161-2.0.1
```

Configure binutils:

```
./configure --nfp --disable-werror --target=mips-harvard-os161 --prefix=$HOME/sys161/tools
```

Make binutils:

```
make
```

Note: If the make command fails because of a problem related to makeinfo, try running the following command:

```
find . -name '*.info' | xargs touch
```

and then re-run make.

Finally, once make has succeeded, install the binutils into their final location:

```
make install
```

This will create the directory $HOME/sys161/tools/ and populate it.

Step 3: Adjust Your Shell's Command Path

First, make the directory in which your shell will eventually find the toolchain binaries:

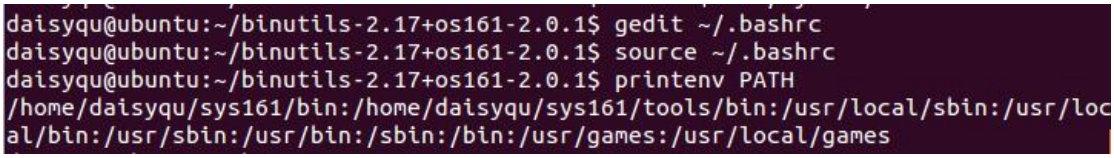
```
mkdir $HOME/sys161/bin
```

Next, add two directories ($HOME/sys161/bin and $HOME/sys161/tools/bin) to your shell's search path. Users of bash should do something like this in your .bash_profile and .bashrc files in your home directory: (You can use gedit ~/.bashrc)

```
export PATH=$HOME/sys161/bin:$HOME/sys161/tools/bin:$PATH
source ~/.bashrc
```

You can check the current setting of the PATH environment variable using the command

```
printenv PATH
```



Step 4: Install the GCC MIPS Cross-Compiler

Unpack the gcc archive:

```
cd ..
tar -xzf os161-gcc.tar.gz
```

Move into the newly-created directory:

```
cd gcc-4.1.2+os161-2.0
```

Configure gcc

```
./configure -nfp --disable-shared --disable-threads --disable-libmudflap --disable-libssp --target=mips-harvard-os161 --prefix=$HOME/sys161/tools
```

Make it and install it:

```
make
make install
```

Step 5：Install GDB

Unpack the gdb archive：

```
cd ..
tar -xzf os161-gdb.tar.gz
```

Move into the newly-created directory：

```
cd gdb-6.6+os161-2.0
```

Configure gdb

```
./configure --target=mips-harvard-os161 --prefix=$HOME/sys161/tools --disable-werror
```

Make it and install it：

```
make
```

Note：Compiling this version of GDB may fail when newer versions of the texinfo package are installed (version 5.2-1, for instance).

```
checking for library containing waddstr... no
checking for library containing tgetent... no
configure: error: no termcap library found
make[1]: *** [configure-gdb] Error 1
make[1]: Leaving directory `/home/daisyqu/gdb-6.6+os161-2.0'
make: *** [all] Error 2
daisyqu@ubuntu:~/gdb-6.6+os161-2.0$
```

If you encounter an issue with this step, you can either include MAKEINFO=missing when you run the make command, or you can install an older version of texinfo (such as 4.13-4) and try again. You may also need to install the libncurses-devel library. Just type：

```
sudo apt-get install libncurses-dev
make MAKEINFO=missing
make install
```

Step 6：Install bmake

Unpack the bmake archive：

```
cd ..
tar -xzf os161-bmake.tar.gz
```

Move into the newly-created directory：

```
cd bmake
```

Unpack mk within the bmake directory：

```
tar -xzf ../os161-mk.tar.gz
```

Run the bmake bootstrap script

```
./boot-strap --prefix=$HOME/sys161/tools
```

As the boot-strap script finishes, it should print a list of commands that you can run to install bmake under $HOME/sys161/tools. The list should look something like this：

```
Commands to install into /home/dasiyqu/sys161/tools/

mkdir -p /home/dasiyqu/sys161/tools/bin
cp /home/dasiyqu/bmake/Linux/bmake /home/dasiyqu/sys161/tools/bin/bmake-20101215
rm -f /home/dasiyqu/sys161/tools/bin/bmake
ln -s bmake-20101215 /home/dasiyqu/sys161/tools/bin/bmake
mkdir -p /home/dasiyqu/sys161/tools/share/man/cat1
cp /home/dasiyqu/bmake/bmake.cat1 /home/dasiyqu/sys161/tools/share/man/cat1/bmak
e.1
sh /home/dasiyqu/bmake/mk/install-mk /home/dasiyqu/sys161/tools/share/mk
```

Of course, your output will refer to your directories, not to /home/daisyqu.

Run the commands printed in the order in which they are listed.

Step 7: Set Up Links for Toolchain Binaries

```
mkdir $HOME/sys161/bin
cd $HOME/sys161/tools/bin
sh -c 'for i in mips-*; do ln -s $HOME/sys161/tools/bin/$i $HOME/sys161/bin/cs350-`echo
$i | cut -d- -f4-`; done'
ln -s $HOME/sys161/tools/bin/bmake $HOME/sys161/bin/bmake
```

When you are finished with these steps, a listing of the directory $HOME/sys161/tools/bin should look similar to this:

```
daisyqu@ubuntu:~/sys161/tools/bin$ ls
bmake                          mips-harvard-os161-gdbtui
bmake-20101215                 mips-harvard-os161-ld
mips-harvard-os161-addr2line   mips-harvard-os161-nm
mips-harvard-os161-ar          mips-harvard-os161-objcopy
mips-harvard-os161-as          mips-harvard-os161-objdump
mips-harvard-os161-c++filt     mips-harvard-os161-ranlib
mips-harvard-os161-cpp         mips-harvard-os161-readelf
mips-harvard-os161-gcc         mips-harvard-os161-run
mips-harvard-os161-gcc-4.1.2   mips-harvard-os161-size
mips-harvard-os161-gccbug      mips-harvard-os161-strings
mips-harvard-os161-gcov        mips-harvard-os161-strip
mips-harvard-os161-gdb
```

the directory $HOME/sys161/bin should look similar to this:

```
dasiyqu@ubuntu:~/sys161/bin$ ls
bmake            cs350-cpp        cs350-gdb        cs350-objdump   cs350-strings
cs350-addr2line  cs350-gcc        cs350-gdbtui     cs350-ranlib    cs350-strip
cs350-ar         cs350-gcc-4.1.2  cs350-ld         cs350-readelf
cs350-as         cs350-gccbug     cs350-nm         cs350-run
cs350-c++filt    cs350-gcov       cs350-objcopy    cs350-size
```

Step 8: Build and Install the sys161 Simulator

Unpack the sys161 archive:

```
cd $HOME
tar -xzf sys161.tar.gz
```

Move into the newly-created directory:

```
cd sys161-1.99.06
```

Next, configure sys161:

```
./configure --prefix=$HOME/sys161 mipseb
```

Build sys161 and install it:

```
make
make install
```

Finally, set up a link to a sample sys161 configuration file

```
cd $HOME/sys161
ln -s share/examples/sys161/sys161.conf.sample sys161.conf
```

Step 9: Install OS/161

First, create a directory to hold the OS/161 source code, your compiled OS/161 kernels, and related test programs.

```
cd $HOME
mkdir cs350-os161
```

Next, move the OS/161 archive into your new directory and unpack it:

```
mv os161.tar.gz cs350-os161
cd cs350-os161
tar -xzf os161.tar.gz
```

This will create a directory called os161-1.99 (under cs350-os161) containing the OS/161 source code. You should now be able build, install, and run an OS/161 kernel and related application and test programs