Part A: What is a toolchain?

In its simplest form, a toolchain is the set of programs used to develop software.
Use the following reference link to answer the questions below.
http://elinux.org/Toolchains
Questions:
1. Define binutils, as defined in the reference.

The GNU Binutils are the first component of a toolchain.

2. List the two major items that comprise binutils, as defined in the reference.

as, the assembler, that turns assembly code (generated by gcc) to binary

ld, the linker, that links several object code into a library, or an executable

3. List the first two pre-built toolchains defined in the reference.

CodeSourcery
Linaro

4. When talking about toolchains, what are the three machine distinctions (as defined in the reference)?

- the build machine, on which the toolchain is built
- the host machine, on which the toolchain is executed
- the target machine, for which the toolchain generates code

Part B: Do you know where your toolchain is?

The toolchain for this course has already been installed on 0xdeadbeef for your use. However, it is important to understand a little about how a toolchain is integrated into the linux environment that you are working with. In general, the integration of a toolchain simply involves placing a selection of programs in a folder that is linked into the mechanism for executing programs from a command line interface.
The following questions must be answered using 0xdeadbeef.boisestate.edu.
Questions:
5. From the command line, issue the which command with the native gcc command as an argument. What is the path displayed?

```
[cguo@0xdeadbeef ~]$ which gcc
/usr/bin/gcc
```

As an example, the following usage of the which command with another command as an argument

```
$ which grep
```

gives the following path:
/usr/bin/grep
6. Repeat for the ARM cross platform arm-none-eabi-gcc. What is the path displayed?

```
which arm-none-eabi-gcc
/usr/local/arm/toolchain/bin/arm-none-eabi-gcc
```

7. Use the ls command to list the other toolchain programs. How many of them are there? (hint: more than 20)

```
arm-none-eabi-addr2line    arm-none-eabi-gcc-4.9.3      arm-none-eabi-ld.bfd
arm-none-eabi-ar           arm-none-eabi-gcc-ar         arm-none-eabi-nm
arm-none-eabi-as           arm-none-eabi-gcc-nm         arm-none-eabi-objcopy
arm-none-eabi-c++          arm-none-eabi-gcc-ranlib     arm-none-eabi-objdump
arm-none-eabi-c++filt      arm-none-eabi-gcov           arm-none-eabi-ranlib
arm-none-eabi-cpp          arm-none-eabi-gdb            arm-none-eabi-readelf
arm-none-eabi-elfedit      arm-none-eabi-gdb-py         arm-none-eabi-size
arm-none-eabi-g++          arm-none-eabi-gprof          arm-none-eabi-strings
arm-none-eabi-gcc          arm-none-eabi-ld             arm-none-eabi-strip
```

8. Verify that the path for ARM cross platform tools is in the linux environment, using the linux echo command as follows.

```
$ echo $PATH
```

In the resulting list, in what position is the ARM cross platform path entry?
Toolchain

(Hint: The command grep in example 5 above, found in the directory "/usr/bin", is in the fifth position in the list.)

/usr/local/arm/toolchain//bin:/usr/local/Xilinx/Vivado/2016.2/bin:/usr/lib64/qt-3.3/bin:/usr/local/arm/toolchain/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/apps/MATLAB/R2016a/bin:/home/students/cguo/.local/bin:/home/students/cguo/bin

Part C: Toolchain versions

Toolchain versions are described on page 6-2 of the course reference.
Differences between versions, though seemingly small, can sometimes affect results. Therefore, it is important to establish the authoritative version for this course when different versions produce different results. In answering the following (and any other) questions in the lab or lecture, note that the version on 0xdeadbeef.boisestate.edu will be used as the official arbitrator for determining the correct answer.
Questions:
9. What is the version of the native assembler on 0xdeadbeef?

> GNU assembler version 2.23.52.0.1 (x86_64-redhat-linux) using BFD version version 2.23.52.0.1-el7 20130226

(Example: on my home installation (Ubuntu), the version is 2.24; the version identifier will be longer for 0xdeadbeef)
10. What is the version of the 0xdeadbeef ARM cross compiler assembler?

> using BFD version version 2.23.52.0.1-55.el7 20130226

Part D: Distributions
Though there are many commercially available (expensive) or proprietary toolchains, the open source gnu toolchain was specifically chosen for this course. This toolchain is relatively simple to use, and provides a powerful set of tools that can run across a wide range of target hardware platforms.
The version of the ARM toolchain used on 0xdeadbeef comes in a tarball with the following name:
gcc-arm-none-eabi-4_9-2014q4-20141203-linux.tar.bz2
Questions:
11. What is the URL for this tarball? (First hit when the string above is entered in google, The URL will contain launchpad). You will need this information during the lab.

https://launchpad.net/gcc-arm-embedded/+download

Part E: Tools studied in the lab

Using the references listed at the beginning of the pre-lab, investigate the following tools:

assembler (as)
compiler (gcc)
linker (ld)
objcopy
objdump

Questions:

12. What chapter in the course reference covers the assembler command?
Ch4

13. What chapter covers the linker command?
Ch6

Part F: Interpreting block diagrams for linux commands

Refer to the figures listed in the references at the beginning of this pre-lab. The lab will require that you create similar block diagrams for most of the commands used during the lab.

Questions:

14. What command is represented by the block diagram for the assembler (Figure 6-4 in the course reference)?

```
$ arm-none-eabi-as prog01.1 –o prog01.s.o –a=prog01.1st
```

15. What command is represented by the block diagram for the gcc compiler (Figure 16-1 in the course reference)?

```
$gcc –o hw hw.c
```