CS195 Assignment 1

Due on: April 15 (Thursday)

Objective
The objective of this assignment is to learn how to use a real-world testing tool to test a significant application on a significant testsuite. Specifically, we will use the DejaGNU testing tool to test the GCC (GNU Compiler Collection) C compiler on the C torture tests in the GCC testsuite.

Resources
For an overview of DejaGNU, see www.gnu.org/software/dejagnu/
For an overview of GCC, see gcc.gnu.org/
Download the following from www.stanford.edu/class/cs195/materials/asgns/asgn1/

1. dejagnu-1.4.4.tar.gz (DejaGNU testing tool)
2. expect.tar.gz (Expect, required by DejaGNU)
3. tcl8.4.2-src.tar.gz (Tcl, required by Expect)
4. gcc-core-3.3.tar.gz (GCC C compiler)
5. gcc-testsuite-3.3.tar.gz (GCC testsuite)

Setup

1. Create a directory (say ASGN1) and sub-directories dejagnu (for installing DejaGNU, Expect, and Tcl) and gcc (for installing the GCC C compiler and the GCC testsuite). Henceforth, ASGN1 refers to the full path (beginning from the root directory).

2. Uncompress the 5 above-mentioned files into the directory ASGN1. This should create sub-directories tcl8.4.2, expect-5.40, dejagnu-1.4.4, and gcc-3.3.

3. Configure, build, and install DejaGNU and the GCC C compiler:

```bash
cd ASGN1/tcl8.4.2/unix
./configure --prefix=ASGN1/dejagnu
make
make install
```
cd ASGN1/expect-5.40
./configure --prefix=ASGN1/dejagnu
make
make install

cd ASGN1/dejagnu-1.4.4
./configure --prefix=ASGN1/dejagnu
make
make install

cd ASGN1/gcc-3.3
./configure --prefix=ASGN1/gcc
make
make install

4. Modify the PATH environment variable to include the paths ASGN1/dejagnu/bin/ and ASGN1/gcc/bin/.

   **Note:** Your UNIX machine will have GCC already installed in a standard path. When you modify the PATH environment variable, ensure that the path ASGN1/gcc/bin/ is given precedence over that standard path. For instance, if you are using the bash shell, you should run the command `export PATH="ASGN1/dejagnu/bin/:ASGN1/gcc/bin:$PATH"`. Check that the command `which gcc` produces output ASGN1/gcc/bin/gcc.

Problems

1. What are the possible outcomes of running a testcase using DejaGNU? Also, briefly describe the meaning of each outcome.

2. The GCC testsuite is located in the directory ASGN1/gcc-3.3/gcc/testsuite/. The testsuite includes sub-testsuites for testing portable and architecture-specific features of various compilers in GCC. We will focus on the sub-testsuite for testing portable features of the C compiler: ASGN1/gcc-3.3/gcc/testsuite/gcc.c-torture/execute/*.c.

   Run DejaGNU on the above testsuite by executing the command:

   ```
   cd ASGN1/gcc-3.3
   make check-gcc RUNTESTFLAGS="execute.exp"
   ```

   This should produce files gcc.sum (summary of the testing) and gcc.log (details of the testing) in the directory ASGN1/gcc-3.3/gcc/testsuite/. Rename them to gcc.old.sum and gcc.old.log and submit them electronically.

3. Peruse the files gcc.old.sum and gcc.old.log. You will notice that DejaGNU invoked the GCC C compiler multiple times on each test case, each time with a different set of command-line options. Explain the rationale underlying which options are chosen for which test cases. [Hint: Read file ASGN1/gcc-3.3/gcc/testsuite/lib/c-torture.exp]
4. Inject a bug in the source code of the GCC C compiler (located in `ASGN1/gcc-3.3/gcc/*.c`) and build and install it again by running the command:

```bash
cd ASGN1/gcc-3.3
make
make install
```

Briefly describe the bug you introduced.

**Note:** The above build should succeed. Do not introduce a trivial bug that is triggered in every execution of the GCC C compiler! An example of such a bug would be dereferencing a NULL pointer at the entry to the main function in the file `ASGN1/gcc-3.3/gcc/main.c`. If you introduce a trivial bug, the above build itself will fail (because GCC bootstraps itself).

5. Add two new test cases named `my-pass.c` and `my-fail.c` to the directory `ASGN1/gcc-3.3/gcc/testsuite/gcc.c-torture/execute/`. Run DejaGNU on the extended testsuite by executing the command:

```bash
cd ASGN1/gcc-3.3
make check-gcc RUNTESTFLAGS="execute.exp"
```

This should produce files `gcc.sum` (summary of the testing) and `gcc.log` (details of the testing) in the directory `ASGN1/gcc-3.3/gcc/testsuite/`. Rename them to `gcc.new.sum` and `gcc.new.log` and submit them electronically.

The two new test cases should be designed such that `my-pass.c` passes and `my-fail.c` fails (because of the bug). Submit these two test cases either electronically or in text.

6. Does the outcome of each test case in `gcc.old.sum` match that of the corresponding test case in `gcc.new.sum`? For each test case for which the outcome does not match, briefly explain why it does not match.

**Submission Instructions**

- The written solutions are due on April 15 (Thursday) at the start of the class.
- The electronic solutions must be emailed to mhn@cs.stanford.edu at any time before the start of the class on April 15 (Thursday).