

gdb Practice Exercise; Project 1 Turn-in

CS 411

gdb Debugging Exercise

In addition to using gdb to debug a program, this exercise will also give you some practice creating a git/GitHub repository from scratch. I'd recommend using phoenix for this exercise.

```
# You're on another system/different account, so you'll need to repeat
# the following tooling configuration. Re-use your identity values.
```

```
git config --global user.name "Your Name"
git config --global user.email Your.Name@mail.goucher.edu
git config --global push.default simple
git config --global core.editor geany # or emacs, gedit, or vi
```

```
# Again, run one of these two commands:
```

```
# 1) Cache credentials in memory for one hour (units are seconds). Adjust
# the timeout as you see fit.
```

```
git config --global credential.helper "cache --timeout=3600"
```

```
# 2) Store credentials unencrypted on disk permanently
# Default storage file is ~/.git-credentials
```

```
git config --global credential.helper store
```

```
# First, create a new GitHub repository for this exercise.
```

```
#
```

```
# If you haven't already done so, go into GoucherLearn and follow the link
# you'll find there to create the GitHub repository for this lab.
```

```
# Replace <your-repository> below with the URL of your repo, which will look
# something like:
```

```
#
```

```
# https://github.com/GoucherCollegeCS411Spring2020/
# project1gdb-tpkelliher.git
```

```
#
```

```
# Except, well, it won't be split across two lines.
```

```
# Create a directory for your gdb exercise. Download fixme.c from the
# class web site and store it in this directory. While you're at it,
# create README.md in the directory and add a few lines to it. Now, from
```

inside this directory, initialize a new git local repository, add the two files to be staged for the initial commit, and make the initial commit:

```
git init
git add .
git commit -m "Initial commit."
```

Now, add your remote, empty GitHub repository. For ease of reference, it # will have "origin" as its name. (Otherwise, you'll be typing the full URL # every time you do a push or a pull.)

```
git remote add origin <your-repository>
```

Now, push your local repository to the remote, setting the default for # upstream to be the master branch of the origin remote repository:

```
git push -u origin master
```

From this point on, you can push to the remote using just

```
git push
```

and pull from the remote using

```
git pull
```

Carefully read the comments in `fixme.c` Using `gdb`, find and fix each of the five bugs in the program. As you fix each bug, leave brief comments in the program itself. In `README.md`, explain the nature of each bug and how you fixed it.

Project 1 Turn-in

By the project deadline, email to me the GitHub https URLs of your kernel repository, and your `gdb` exercise repository.

Make sure that the work that you want me to assess is in the (default) master branch of your repositories. For example, I will be using the following sequence to pull your kernel syscall work and set the branch that I will assess from my local copy of the kernel repo:

```
git remote add foo <your-repository>
git fetch foo
git checkout -b foo foo/master
```

This project will be assessed as follows:

- Kernel compile/syscall exercise: 30%
- `gdb` exercise: 60%
- Documentation: 10%

Document assistance you received from others. This should be done in your `README.md` files.