601.220 Intermediate Programming

Spring 2023, Day 3 (Jan 27th)

Today's agenda

- Academic ethics for individual homeworks
- Exercise 2 review
- Day 3 material
  - Git and Emacs
- Exercises 3-A and 3-B

Reminders

- HW0 is due Friday, Feb 3rd
    - You will need to be fully set up to use your private Git repo!
    - You will also need to have access to Gradescope
    - Let us know ASAP if either of these isn't the case

Academic ethics for individual homeworks

- The individual homeworks (HW0–HW7) must be completed *individually*
- Do not look at anyone else's work/code
- Do not allow anyone (other than course staff) to look at your work/code
- Do not use code from the internet, students who have taken the course previously, etc.
- Do not ask ChatGPT or other AI models to write your code for you
- We will run a similarity comparison on submissions
- Violations will be reported to the student conduct office

Exercise 2 review

Reading two integers:

```
int a, b;
printf("Enter two integers: ");
scanf("%d", &a);
scanf("%d", &b);
```

Another possibility:

```
int a, b;
printf("Enter two integers: ");
scanf("%d %d", &a, &b);
```

Exercise 2 review

The user might not enter a valid integer. You can detect this by checking the return value of `scanf`, which returns a count of how many values were read successfully.

```
int a;
printf("Enter an integer: ");
if (scanf("%d", &a) != 1) {
  printf("invalid input\n");
}
```

This technique will (likely) be useful for HW1.

Exercise 2 review

Computations on input values a and b:

```
int sum, diff, prod, quot, rem;
sum  = a + b;
diff = a - b;
prod = a * b;
quot = a / b;
rem  = a % b;
```

Exercise 2 review

Printing results of computations:

```
printf("%d + %d = %d\n", a, b, sum);
printf("%d - %d = %d\n", a, b, diff);
printf("%d * %d = %d\n", a, b, prod);
printf("%d / %d = %d\n", a, b, quot);
printf("%d %% %d = %d\n", a, b, rem);
```

Note %% in a `printf` format string means “print a literal % character.”

Don’t forget to add a newline (`\n`) when you want to end the current line of text (to avoid all of the output being on one line.)

Day 3 recap questions

1. Why do we use version control system like **git**?
2. Name six common **git** commands?
3. What are the files that must be included in your submission?
4. How do you save and quit on *emacs* editor?
5. How do you search and replace on *emacs*?

1. Why do we use version control system like **git**?

Version control systems record and archive a history of “snapshots” the files in a project. (Git calls these snapshots “commits”.)

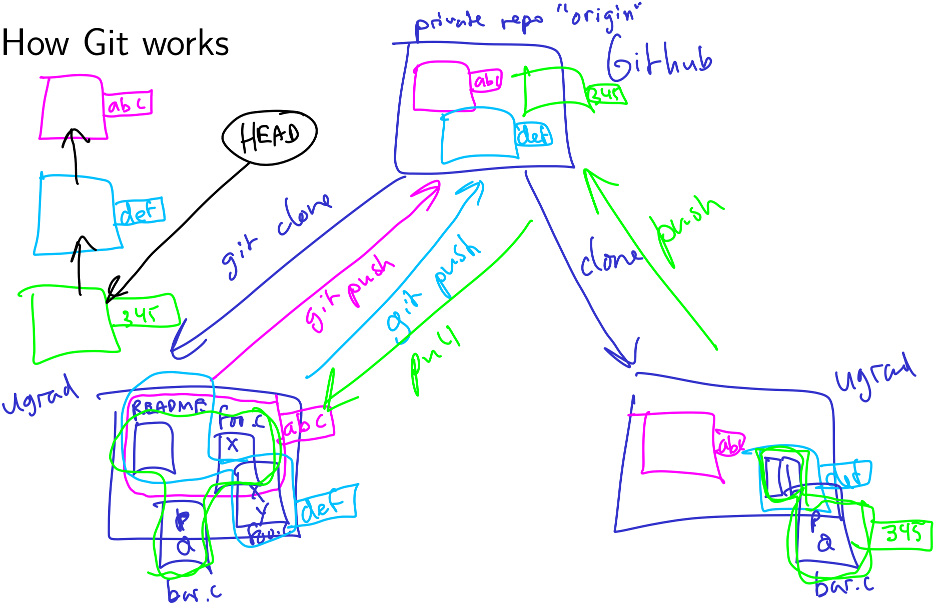
Advantages:

- Know exactly how code changed, be able to revert to working code if a bug is introduced or if a file is lost/corrupted
- Serves as a backup when you push commits to a remote repository (i.e., on Github)
- Team members can share changes with each other. (This is essential for team projects.)
- Synchronize files between multiple computers

2. Name six common **git** commands?

1. `git init`: initialize the current directory as a new Git repository
2. `git add`: specify file(s) that should be part of the next commit
3. `git commit`: create a new commit with all of the modifications staged using `git add`, with a meaningful log message
4. `git status`: show which files have been modified are haven't yet been added to the repository
5. `git diff`: show file modifications (relative to the previous commit)
6. `git push`: send commit(s) to the remote "origin" repository
7. `git pull`: receive commit(s) from the origin repository

How Git works



3. What are the files that must be included in your submission?

- All source files (e.g., .c, .h files)
- gitlog.txt: summarizes your Git commits
- Possibly: README
- Possibly: a Makefile (once we've covered make)

Create a zip file with all of these files, copy it to your local computer, then upload the zip file to Gradescope. E.g.:

```
cd ~/my220repo/homework/hw0
zip -9r my_hw0.zip *.c gitlog.txt
```

ugrad

On your local machine

```
scp USERNAME@ugradx.cs.jhu.edu:my220repo/homework/hw0/my_hw0.zip .
```

4. How do you save and quit on *emacs* editor?

Save: Control-X followed by Control-S

Quit: Control-X followed by Control-C

Pro tip: use multiple terminal windows! Keep your editor (e.g., Emacs) open in one terminal, then use another terminal to run compiler commands, run the program, etc.

5. How do you search and replace on *emacs*?

Meta-% (a.k.a. ESC followed by %)

Work on Exercises 3-a and 3-b

- Exercise 3-a: Clone your private repository
- Exercise 3-b: Clone the public repository, copy starter code, work on a program

Recommendation: When cloning repositories, use the **ssh** version of the Git URL

- This will work if you've created an SSH key and added it to your Github account
- We can help you set this up!

Notes

Notes

Notes

Notes

Notes