

CSE 220: Systems Fundamentals I

Stony Brook University

Fall 2024

GitHub

If you do not have a GitHub account, create one now. Otherwise, login to your account.

CodeSpaces

GitHub provides a virtual machine-like development environment known as a Docker, which can be easily customized to install whatever software, languages, tools, packages, etc. are needed. We will develop C software in Linux in Docker containers to simplify the distribution, collection and grading of assignments. Codespaces is a feature in GitHub that uses Docker environments and makes it simple to get started coding right away.

For the sake of practicing on the Linux command line, we will create an empty repository on GitHub and then create a Codespace for it.

1. At github.com, click the **New** button in the upper-left corner of the page. (If you don't see it, go to the landing page of github.com.)
2. For "Repository name", enter "cse220_playground".
3. Make the repository private.
4. Check the box for "Add a README file"
5. Click the **Create repository** button.
6. You should now see the repository, which contains only a README file. Press the **Code** button, then "Codespaces", then click the **Create codespace on main** button.

Important: this is **not** how you will create the repositories for the homework assignments. You will be given starter code that must be obtained by cloning an existing repository. Directions will be provided in the first homework assignment.

Linux Command Line Basics

Based loosely on the Ubuntu tutorial [Linux command line for beginners](#).

Working with Directories

1. `pwd` Shows you what directory you're currently looking at
2. `ls` Lists the files and directories of the current working directory (cwd)
3. `ls -l` Lists contents of the current working directory in "long listing format"
4. `mkdir src` Creates a new directory named "src"
5. `cd src` Enters the new directory
6. `ls -l`
7. `cd ..` Goes up to the parent directory
8. `pwd`

Copying, Renaming/Moving and Deleting Files & Directories

- `cp file1 file2` Makes a copy of `file1` named `file2`
- `mv file1 file3` Changes the name of `file1` to `file3`
- `rm file1` Deletes `file1`

Redirecting Input and Output; Pipes

1. Type a few names into a new file called `names.txt`
2. `cat names.txt`
Displays the entire contents of the file `names.txt`.
3. `less names.txt`
Displays the entire contents of the file `names.txt`, one page at a time.
4. `sort names.txt` Sorts the lines of the file and prints it.
5. `sort names.txt > sorted.txt`
Sorts the lines of the file and redirects (sends) the output to the file `sorted.txt` instead of the screen. `sorted.txt` is overwritten if it exists.

6. `cat names.txt >> sorted.txt`

Sorts the lines of the file and appends the output to the file `sorted.txt` instead of the screen.

7. `cat names.txt | grep` and then type a name that is in the file

`grep` searches the contents of `names.txt` for some text you provide.

gcc: CSE 220's C compiler

1. Create a simple `.c` file that prints a short message and save it in `prog.c`

```
#include <stdio.h>
int main() {
    printf("Hello world!\n");
    return 0;
}
```

2. `gcc -c prog.c`

Generates `prog.o` ("object" file)

3. `ls -l`

4. `rm prog.o`

5. `gcc prog.c`

Generates `a.out` from source code

6. `ls -l`

7. `./a.out`

8. `rm a.out`

9. `gcc prog.o -o prog`

Generates `a.out` from object file

10. `ls -l`

11. `./prog`

12. `gcc -E prog.c`

Shows preprocessor output.

13. `gcc -S prog.c`

Generates assembly code for the host CPU.

14. `ls -l`

15. `less prog.s`

Danger Zone

- `sudo`
- `rm -fr`

Miscellaneous

- `man`

Access manuals for terminal commands