

CS 162 HW 5

[Background](#) / [Code](#)

Code

NOTE

This might look like a lot of code, but much of it is already written for you. You will only need to understand and modify around 3 files to complete this assignment.

```
.
├── Makefile
├── app
│   ├── app.c
│   ├── app.h
│   ├── grep
│   │   ├── grep.c
│   │   └── grep.h
│   ├── vertex_degree
│   │   ├── vertex_degree.c
│   │   └── vertex_degree.h
│   └── wc
│       ├── wc.c
│       └── wc.h
├── bin
├── client
│   ├── client.c
│   └── client.h
├── codec
│   ├── codec.c
│   └── codec.h
├── coordinator
│   ├── coordinator.c
│   ├── coordinator.h
│   ├── job.c
│   └── job.h
├── data
│   ├── alphabet2
│   │   ├── letters1.txt
│   │   └── letters2.txt
│   ├── graph-edges-medium
│   │   ├── 00.txt
│   │   ├── 01.txt
│   │   ├── 02.txt
│   │   ├── 03.txt
│   │   └── 04.txt
│   └── gutenber
│       ├── p.txt
│       ├── q.txt
│       ├── r.txt
│       └── s.txt
```

```
├── t.txt
├── u.txt
├── lib
│   ├── lib.c
│   └── lib.h
├── rpc
│   └── rpc.x
├── worker
│   ├── worker.c
│   └── worker.h
```

app/

MapReduce applications such as word count and distributed grep. You may not modify any of the existing files in this folder, although you are free to test out your own MapReduce apps if you like.

bin/

Binaries that are generated by `make`.

client/

Code for the MapReduce client. You may not modify these files.

codec/

Helper functions for serialization, which is used by the worker implementation to read and write data. You may not modify the code in this folder as our tests depend on it.

coordinator/

Code for the MapReduce coordinator. All of your changes should be made in this folder. The majority of your changes will be likely be made in the `coordinator.c` file, but you may encapsulate some of your job-specific logic in `job.c` and `job.h`. You can find the definition of the struct used to hold coordinator state in `coordinator.h`.

data/

Data files for testing your MapReduce implementation. The `alphabet2` and `gutenberg` folders contain data that can be used with the `wc` (word count) and `grep` applications, while the `graph-edges-medium` folder contains data for the `vertex-degree` application.

lib/

Helper functions, types, and constants that may come in useful for your implementation. You may not modify this file, but you are encouraged to take a look at its contents.

rpc/

Protocol definitions and autogenerated code for the coordinator server. The definition file `rpc.x` contains the RPCs that you will need to implement, as well as their arguments and return types. You can look at the autogenerated `rpc.h` to see how the XDR definition language is translated to C.

worker/

Code for MapReduce workers. You may not modify these files, though it may be helpful to read through what the workers are doing at a high level.