

User Manual

The guide assumes that you have the ability to compile Rust code on your system.

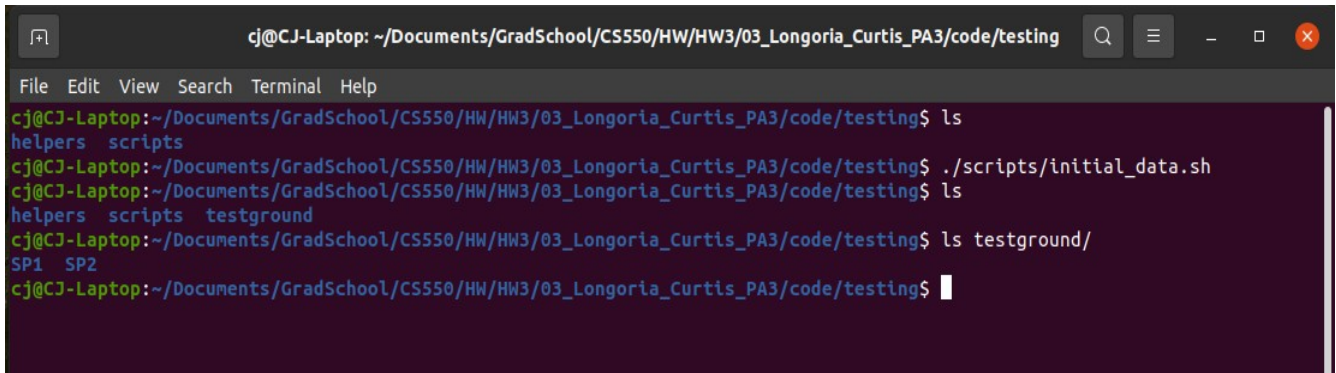
The file structure is depicted below:

```
docs
|-----Performance.pdf
|-----Design.pdf
|-----Tests.pdf
|-----Manual.pdf
|-----Output.pdf
code
|-----build.rs
|-----Cargo.toml
|-----Cargo.lock
|-----proto
|         |-----p2p-server.proto
|-----src
|         |-----client.rs
|         |-----server.rs
|-----testing
|         |-----helpers
|         |         |-----all-config.ini
|         |         |-----linear-config.ini
|         |-----scripts
|         |         |-----file_mod.sh
|         |         |-----initial_data.sh
|         |         |-----make_data.sh
|         |         |-----perf.sh
|-----target
```

Quick Start

Step 0

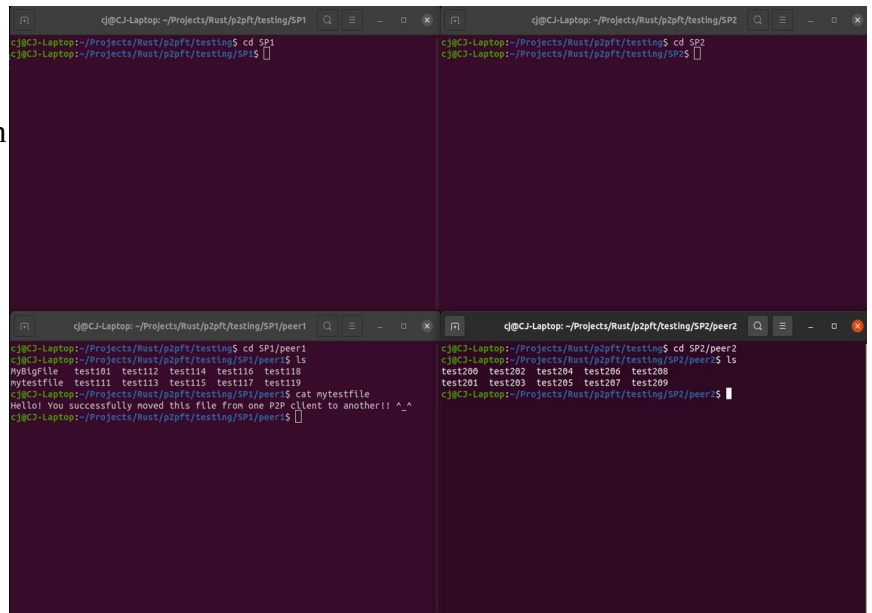
Begin by navigating to the testing folder (Code>testing) and running the initial_data script using the command “./script/initial_data.sh”. After running the command verify that you have a “testground” folder inside the “testing” folder. Additionally, verify that the “testground” folder has the three directories “SP1” and “SP2” inside it.



```
cj@CJ-Laptop: ~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing
File Edit View Search Terminal Help
cj@CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing$ ls
helpers  scripts
cj@CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing$ ./scripts/initial_data.sh
cj@CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing$ ls
helpers  scripts  testground
cj@CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing$ ls testground/
SP1  SP2
cj@CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing$
```

Step 1

Now open four terminals and navigating to the testing folder (Code>testing>testground) on each. Then navigate to the “SP1” (top-left) and “SP2” (top-right) directories in the top two terminals terminals. Navigate to SP1>peer1 (bottom-left) and SP2>peer2 (bottom-right) in the bottom two terminals. Notice that there is a test file “testfile” in the SP1>peer1>Owned directory. Go ahead and type “cat Owned/testfile” in the bottom-left terminal to see what it says. During this quick start we will move that test file from the SP1>peer1>Owned directory to the SP2>peer2>Downloaded directory using the P2P program. Refer to the picture on the right as the starting point for this quick start guide.



```
Top-left: cj@CJ-Laptop: ~/Projects/Rust/p2pft/testing/SP1
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP1$ cd SP1
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP1$

Top-right: cj@CJ-Laptop: ~/Projects/Rust/p2pft/testing/SP2
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP2$ cd SP2
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP2$

Bottom-left: cj@CJ-Laptop: ~/Projects/Rust/p2pft/testing/SP1/peer1
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP1/peer1$ ls
MyBigFile  test101  test112  test114  test116  test118
mytestfile test111  test113  test115  test117  test119
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP1/peer1$ cat mytestfile
Hello! You successfully moved this file from one P2P client to another!! ^_^
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP1/peer1$

Bottom-right: cj@CJ-Laptop: ~/Projects/Rust/p2pft/testing/SP2/peer2
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP2/peer2$ cd SP2/peer2
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP2/peer2$ ls
test200  test202  test204  test206  test208
test201  test203  test205  test207  test209
cj@CJ-Laptop:~/Projects/Rust/p2pft/testing/SP2/peer2$
```

Step 2

First run “cargo run --bin p2p-server” in the top terminals and “cargo run --bin p2p-client” in the bottom terminals. Notice that the server logs that new clients have registered to the P2P system and lists each client’s info, including what files they have available for download. Refer to the picture below for how this should look. *Note: if you get any errors about a thread panic exit (CTRL+C or type “exit”) and re-launch the binary.

```
id: 13515671703725017856,
state: 4215873836689564450,
files: [
  "test0",
  "test1",
  "test2",
  "test3",
  "test4",
  "test5",
  "test6",
  "test7",
  "test8",
  "test9",
  "testfile",
],
ip: "192.168.86.24:50160",
ip_qe: "192.168.86.24:50161",
}

-----Server Summary-----
Active clients: 1
Active files in index: 11
-----

c:\CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing$ cd testgr
ound/SP2/peer1
c:\CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing/testgrou
nd/SP2/peer1$ cargo run --bin p2p-client
    Finished dev [unoptimized + debuginfo] target(s) in 0.05s
    Running `./home/cj/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/target/de
bug/p2p-client`

Welcome to P2P File Transfer Client

Commands (case-insensitive):
  Help
  List
  Search <filename>
  Get <filename>
  Refresh
  Exit

p2pft>

state: 11348544933590540587,
files: [
  "test10",
  "test11",
  "test12",
  "test13",
  "test14",
  "test15",
  "test16",
  "test17",
  "test18",
  "test19",
],
ip: "192.168.86.24:50166",
ip_qe: "192.168.86.24:50167",
}

-----Server Summary-----
Active clients: 1
Active files in index: 10
-----

Reset list of sent queries to super-peer neighbors
Reset list of sent queries to super-peer neighbors

c:\CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing$ cd testgr
ound/SP2/peer1
c:\CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing/testgrou
nd/SP2/peer1$ cargo run --bin p2p-client
    Blocking waiting for file lock on build directory
    Finished dev [unoptimized + debuginfo] target(s) in 41.52s
    Running `./home/cj/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/target/de
bug/p2p-client`

Welcome to P2P File Transfer Client

Commands (case-insensitive):
  Help
  List
  Search <filename>
  Get <filename>
  Refresh
  Exit

p2pft>
```

Step 3

Now enter the command “list” then press enter in both bottom terminals. You can see that the client returns a list of all files available within the respective super-peer group. Notice that each client sees a different list of files. This is because one client is on super-peer 1 and the other is on super-peer 2. We will be able to move files via search function and super-peer broadcasting. The top terminals that are running the index servers (super-peers) might have a log entry

```
"test0",
"test1",
"test2",
"test3",
"test4",
"test5",
"test6",
"test7",
"test8",
"test9",
"testfile",
},
ip: "192.168.86.24:50160",
ip_qe: "192.168.86.24:50161",
}

-----Server Summary-----
Active clients: 1
Active files in index: 11
-----

Reset list of sent queries to super-peer neighbors
Reset list of sent queries to super-peer neighbors
Got a full list request from Client: 13515671703725017856

c:\CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing$ cd testgr
ound/SP2/peer1
c:\CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing/testgrou
nd/SP2/peer1$ cargo run --bin p2p-client
    Finished dev [unoptimized + debuginfo] target(s) in 0.05s
    Running `./home/cj/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/target/de
bug/p2p-client`

Welcome to P2P File Transfer Client

Commands (case-insensitive):
  Help
  List
  Search <filename>
  Get <filename>
  Refresh
  Exit

p2pft> list

Available Files:
[
  "test0",
  "test1",
  "test2",
  "test3",
  "test4",
  "test5",
  "test6",
  "test7",
  "test8",
  "test9",
  "testfile",
]

p2pft>

"test11",
"test12",
"test13",
"test14",
"test15",
"test16",
"test17",
"test18",
"test19",
},
ip: "192.168.86.24:50166",
ip_qe: "192.168.86.24:50167",
}

-----Server Summary-----
Active clients: 1
Active files in index: 10
-----

Reset list of sent queries to super-peer neighbors
Reset list of sent queries to super-peer neighbors
Reset list of sent queries to super-peer neighbors
Reset list of sent queries to super-peer neighbors
Got a full list request from Client: 15015141128355039683

c:\CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing$ cd testgr
ound/SP2/peer1
c:\CJ-Laptop:~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/testing/testgrou
nd/SP2/peer1$ cargo run --bin p2p-client
    Finished dev [unoptimized + debuginfo] target(s) in 0.05s
    Running `./home/cj/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis_PA3/code/target/de
bug/p2p-client`

Welcome to P2P File Transfer Client

Commands (case-insensitive):
  Help
  List
  Search <filename>
  Get <filename>
  Refresh
  Exit

p2pft> list

Available Files:
[
  "test10",
  "test11",
  "test12",
  "test13",
  "test14",
  "test15",
  "test16",
  "test17",
  "test18",
  "test19",
]

p2pft>
```

that the list of sent queries to super-peer neighbors has been reset. This is normal as the list is periodically cleared.

Step 4

Now enter “search testfile” and press enter in the bottom-right terminal. The program returns that the file is available from one other client. The server logs that a client made a search request. The top terminals will display a lot of messages on this step. You can see that since we didn’t provide any CLI arguments the servers attempt to broadcast to all super-peers assigned in our static configuration file. However, the only successfully ones are from the servers running in the terminals. The top-left terminal shows a search request entry followed by a “pushing a query response” this means the super-peer had a query hit and is back-propagating the results. The top-right terminal shows an entry that it is broadcasting a response to the final destination. This is just pushing the query results from another super-peer along.

The image displays three terminal windows from a Linux desktop environment, illustrating the configuration and operation of a peer-to-peer file transfer system (p2pft).

Top-Left Terminal Window: Shows the 'Server Summary' for a node named 'q@CJ-Laptop: ~/Documents/GradSchool/CS550/HW/HW3/03_Longoria_Curtis...'. It reports 11 active clients and 11 active files in the index. Below this, it lists several 'Reset list of sent queries to super-peer neighbors' and 'Pushed query onto qd engine from broadcast' messages, followed by a list of broadcasted queries to various IP addresses (e.g., 192.168.86.24:50056).

Top-Right Terminal Window: Shows the 'Active clients' list for the same node. It reports 10 active clients and 10 active files in the index. It also lists several 'Reset list of sent queries to super-peer neighbors' and 'Got a query request from client' messages, followed by a list of broadcasted queries to various IP addresses (e.g., 192.168.86.24:50051).

Bottom Terminal Window: Shows the 'p2pft' command-line interface. It displays a list of available files, including 'test0', 'test1', 'test2', 'test3', 'test4', 'test5', 'test6', 'test7', 'test8', 'test9', and 'testfile'. The user enters the command 'p2pft> list', which shows the list of available files. The user then enters the command 'p2pft> search testfile', which shows the search results.

Step 5

Now enter “get testfile” and press enter in the bottom-right terminal. The program returns that the file is available and references the client’s IP address that has the file. The program then transparently reaches out to the peer and downloads the file. The server logs that a client made a search request, but has no visibility on if the client downloaded a file from another peer. After the file is downloaded the auto-update mechanism notices a change in the client’s state and deregisters/re-registers the client with the indexing server.


```
j@CJ-Laptop: ~/Documents/Gradschool/CSS50/HW/HW3/03_Longoria_Curtis_P...  
Reset list of sent queries to super-peer neighbors  
Reset list of sent queries to super-peer neighbors  
Reset list of sent queries to super-peer neighbors  
Reset list of sent queries to super-peer neighbors  
Reset list of sent queries to super-peer neighbors  
Reset list of sent queries to super-peer neighbors  
Got a query request from Client: 13515671703725017856  
broadcasting to http://192.168.86.24:50054  
Successfully broadcasted to peer  
Broadcasting to http://192.168.86.24:50056  
Broadcasting to http://192.168.86.24:50060  
Broadcasting to http://192.168.86.24:50062  
Broadcasting to http://192.168.86.24:50064  
Broadcasting to http://192.168.86.24:50066  
Pushed query to neighbors  
Got a search request from Client: 13515671703725017856  
pushing a invalid message  
Reset list of sent queries to super-peer neighbors  
Reset list of sent queries to super-peer neighbors  
Reset list of sent queries to super-peer neighbors  
Reset list of sent queries to super-peer neighbors  
Reset list of sent queries to super-peer neighbors  
  
j@CJ-Laptop: ~/Documents/Gradschool/CSS50/HW/HW3/03_Longoria_Curtis_P...  
File Edit View Search Terminal Help  
List  
Search <filename>  
Get <filename>  
Refresh  
Exit  
  
p2pf> list  
  
Available Files:  
  
[  
    "test0",  
    "test1",  
    "test2",  
    "test3",  
    "test4",  
    "test5",  
    "test6",  
    "test7",  
    "test8",  
    "test9",  
    "testfile",  
]  
  
j@CJ-Laptop: ~/Documents/Gradschool/CSS50/HW/HW3/03_Longoria_Curtis_P...  
p2pf> search testfile  
Searching...  
  
Available Clients: 1  
p2pf> get testfile  
Downloading from 192.168.86.24:50100  
File Downloading  
File Downloaded  
p2pf>  
Deleted file: testfile  
This file was removed from your Downloads folder since it was redundant.  
You can re-download all expired files with the 'refresh' command  
p2pf> refresh  
Refresh Complete!  
p2pf> exit  
Deregistered from Server  
Exiting  
cjs@CJ-Laptop: ~/Documents/Gradschool/CSS50/HW/HW3/03_Longoria_Curtis_PA3/code/testing/testground  
/$P2$/peer$ cat Downloaded/testfile  
Hello, World! This file is used for the demo.  
mow  
cjs@CJ-Laptop: ~/Documents/Gradschool/CSS50/HW/HW3/03_Longoria_Curtis_PA3/code/testing/testground  
/$P2$/peer$
```

User Notes:

- The super-peers default to a broadcast configuration (where every super-peer is connected to every other super-peer). This can be changed to a linear configuration by running “cargo run --bin p2p-server linear” instead of “cargo run --bin p2p-server”
- The client program run with a push based consistency model by default. This will broadcast out an invalidation message anytime the original file is modified. This can be switched to a pull based method with a configurable TTR value located in the config ini files in the helpers directory. In order to use the pull based method run the client binary with “cargo run --bin p2p-client pull”
- The super-peers and their associated clients (leaf-nodes) are pre-configured in a static configuration file under the “helpers” directory. This means you have to run the p2p-server binary from one of the SP directories and you have to run the p2p-client binary in one of the peer directories within one of the SP directories.
- Since the ports are static and pre-configured the thread will panic if the port is already in use. In this event you will have to re-run the program.
- TTL (time-to-live) is set to 3. So if you run the servers in a linear configuration you wouldn't have complete coverage of all super-peers.
- The servers can eat a lot of CPU if you run a lot of them at the same time. Each server consumes about 6% (while idle) of CPU on my machine

