Objectives:

1. Exposure to directory synchronization
2. Implementing replication transparency
3. Experience with multithreading

Project Specification:

This lab is intended to be built upon Lab #1. While it is not a requirement that this lab utilize your code from Lab #1, all of the functionality from Lab #1 must be included in Lab #2.

These will be **individual** projects. You may write the program in any language that is supported under any Integrated Development Environment (IDE). Keep in mind that available controls, objects, libraries, et cetera, may make some of these tasks easier in one language than in another. Finally, because of the lack of restrictions on IDEs, you will have to have that IDE available to demo to the TA (e.g., you will demo the program on your own laptop).

Lab #1 Infrastructure

You will write a program that will generate a composite directory listing from multiple servers. Your project will consist of a client process and two server processes and function as a command line instruction.

Each server, Server A and Server B, will feature a pre-designated directory named `directory_a` and `directory_b`, respectively. When executed, your client will establish a connection to Server A, which will generate a listing of the contents of `directory_a` (this listing is analogous to the `ls -l` command on Linux/Bash or `dir` on Windows). Server A will then establish a connection to Server B, which will generate a listing of the contents of `directory_b` and return the listing to Server A.

Server A should combine the listing of contents of `directory_a` and `directory_b` into a single list sorted by file name. The list should only include the file name, file size, and either the time the file was created or the time the file was last modified. Server A will return the composite list to the Client, which will print the data to the command line.

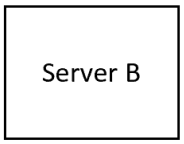
Lab #2 Additions

Server A and Server B will autonomously synchronize the contents of `directory_a` and `directory_b` during runtime, including both files and file metadata. During runtime, any change to the contents of a directory on one server, including adding, deleting, or modifying a file, should be applied at the other server. Files will be added, deleted, or modified with the host's native file manager (e.g., Windows Explorer for Windows or Finder on macOS), and for the purposes of this lab assignment neither directory will include subdirectories.

Upon startup, Servers A and B will generate an inventory of the contents of their designated directories and compare contents. Any content discrepancy should be addressed, with duplicated files being made consistent based on the most-recent modified-at time.

Any change to the contents of the directory during runtime should be recognized within five seconds. The user should be notified of the servers' actions in real-time, and the notifications should include which file is being synchronized when applicable. The mechanism by which the directory contents are made consistent is left to the developer's discretion.

Example:

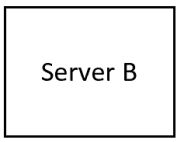



```
/directory_b/
      - a.jpg    575.1kB    20-Aug
      - c.pdf    64.4kB     14-Jul
      - e.txt    400B       28-Sep
```

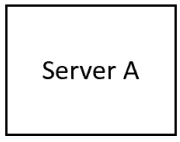

```
/directory_a/
      - b.txt    331B       01-Jun
      - d.gif    1.2MB      26-Jan
      - e.txt    400B       29-Sep
```

T_0: Prior to Startup

At time T_0, the contents of the designated directories are divergent. Server A and Server B both have a copy of `e.txt`, but the modified-at times are different.



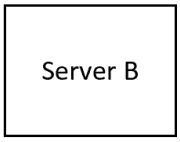

```
/directory_b/
      - a.jpg    575.1kB    20-Aug
      - b.txt    331B       01-Jun
      - c.pdf    64.4kB     14-Jul
      - d.gif    1.2MB      26-Jan
      - e.txt    400B       29-Sep
```




```
/directory_a/
      - a.jpg    575.1kB    20-Aug
      - b.txt    331B       01-Jun
      - c.pdf    64.4kB     14-Jul
      - d.gif    1.2MB      26-Jan
      - e.txt    400B       29-Sep
```

T_1: Runtime

At T_1, the contents of the shared directories have been synchronized and the system is consistent. The most recent version of `e.txt` has been applied across both servers.

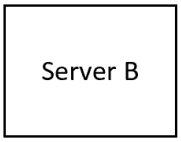



```
/directory_b/
      - a.jpg     575.1kB    20-Aug
      - aa.jpg    424.4kB    01-Oct
      - b.txt     331B       01-Jun
      - c.pdf     64.4kB     14-Jul
      - d.gif     1.2MB      26-Jan
      - e.txt     400B       29-Sep
```


```
/directory_a/
      - a.jpg     575.1kB    20-Aug
      - b.txt     331B       01-Jun
      - c.pdf     64.4kB     14-Jul
      - d.gif     1.2MB      26-Jan
      - e.txt     400B       29-Sep
```

T_2: File addition at Server B

At time T_2, file `aa.jpg` has been added to `directory_b` at Server B. The content of the directories is inconsistent.

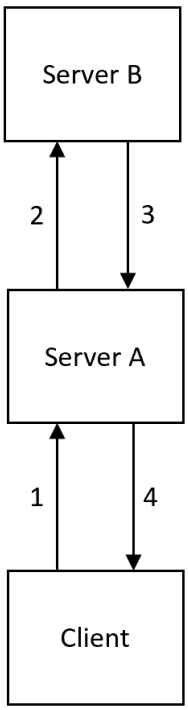



```
/directory_b/
      - a.jpg     575.1kB    20-Aug
      - aa.jpg    424.4kB    01-Oct
      - b.txt     331B       01-Jun
      - c.pdf     64.4kB     14-Jul
      - d.gif     1.2MB      26-Jan
      - e.txt     400B       29-Sep
```


```
/directory_a/
      - a.jpg     575.1kB    20-Aug
      - aa.jpg    424.4kB    01-Oct
      - b.txt     331B       01-Jun
      - c.pdf     64.4kB     14-Jul
      - d.gif     1.2MB      26-Jan
      - e.txt     400B       29-Sep
```

T_3: Contents synchronized across servers

At time T_3, the system has identified the divergence of the directory contents and `aa.jpg` has been copied to `directory_a` at Server A. The system is now consistent.




```
/directory_b/
      - a.jpg    575.1kB    20-Aug
      - aa.jpg   424.4kB    01-Oct
      - b.txt    331B       01-Jun
      - c.pdf    64.4kB     14-Jul
      - d.gif    1.2MB      26-Jan
      - e.txt    400B       29-Sep

/directory_a/
      - a.jpg    575.1kB    20-Aug
      - aa.jpg   424.4kB    01-Oct
      - b.txt    331B       01-Jun
      - c.pdf    64.4kB     14-Jul
      - d.gif    1.2MB      26-Jan
      - e.txt    400B       29-Sep
```

T_4: Client Query

```
user@Client:~$ ./lab2
a.jpg            575.1kB          20-Aug
aa.jpg           424.4kB          01-Oct
b.txt            331B             01-Jun
c.pdf            64.4kB           14-Jul
d.gif            1.2MB            26-Jan
e.txt            400B             29-Sep
```

At time T_4, the client has initiated a query of the file contents. The contents are returned to the client and the output ignores the replicated files. Servers A and B should continue to synchronize content and respond to content queries until manually terminated by the user.

Notes:

- All processes may run on the same physical machine.
- Server A and Server B may be run from different command line instances.
- The IP address and port number of Server A and Server B may be hardcoded.
- The formats of the dates and file sizes are developer's discretion.
- Files may be added, deleted, or modified at any time.
- The contents of the designated directories will be verified with the host's native file manager.
- The program must operate independently of a browser engine.

Citations:

You may use open source code found on the Internet in your program. When citing this code: