Informe de elaboración - Sistemas Operativos - Práctica #2

Integrantes:

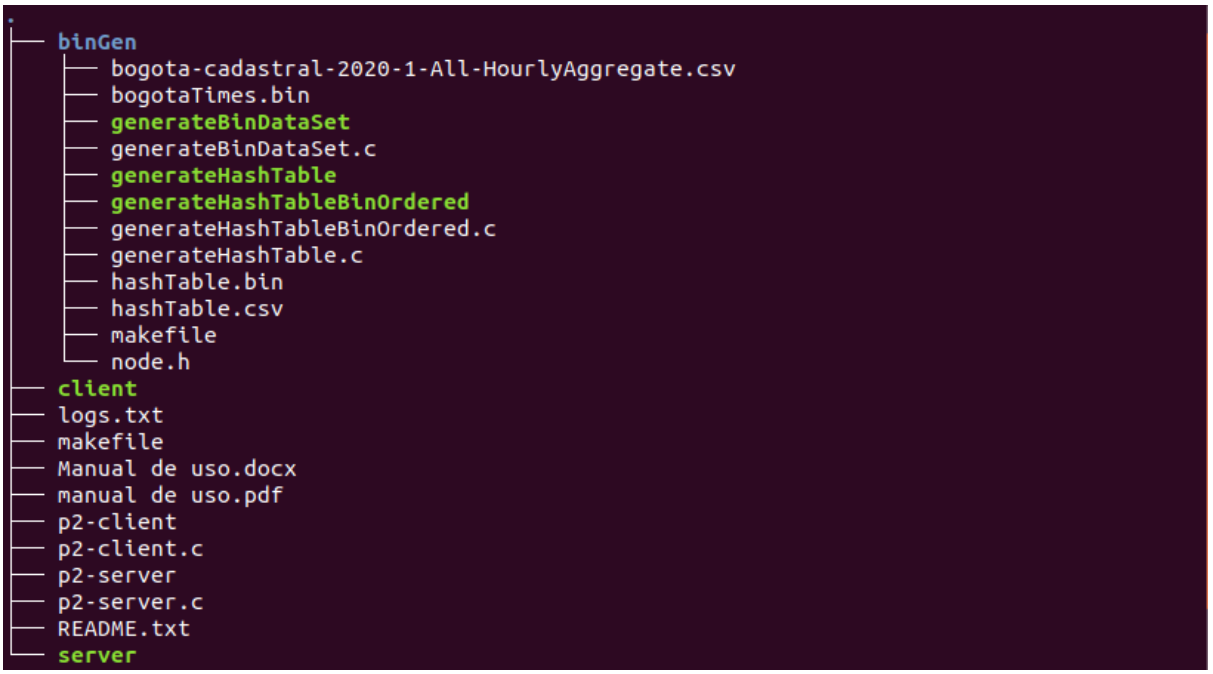
- Diego Fernando Bello Lopez.
- Andres Eduardo Rubiano Martinez.
- Santiago Tovar Mosquera.

Objetivo:

Diseñar dos programas cliente-servidor, que permitan acceder a los registros almacenados en disco.

Elaboración:

La siguiente imagen muestra la estructura del programa en el folder que se encuentra en github (https://github.com/aerubianoma/Operating-systems/Lab2):



A continuación se muestra un paso a paso de elaboración:

1. Generación de archivos binarios: Esto se realiza mediante generateHashTable.c, generateHashTableBinOrdered.c y generateBinDataSet.c los cuales implementan el esquema mostrado en la funcionalidad del servidor.
2. Búsqueda: Este proceso se realiza en el servidor mediante la función search() que usa el esquema del paso 1. la cual permite realizar búsquedas más rápidas.
3. Menú: Se implementa un menú básico para el ingreso de los parámetros de búsqueda.
4. Cliente-Servidor: Se implementan las variables y estructuras necesarias para la comunicación por medio de paso de mensajes dentro del servidor se realiza la

búsqueda luego de que el cliente despliegue el menú y envía los parámetros al servidor.

Funcionalidades:

Cliente: El cliente posee la funcionalidad de desplegar el menú, como el que se encuentra a continuación:

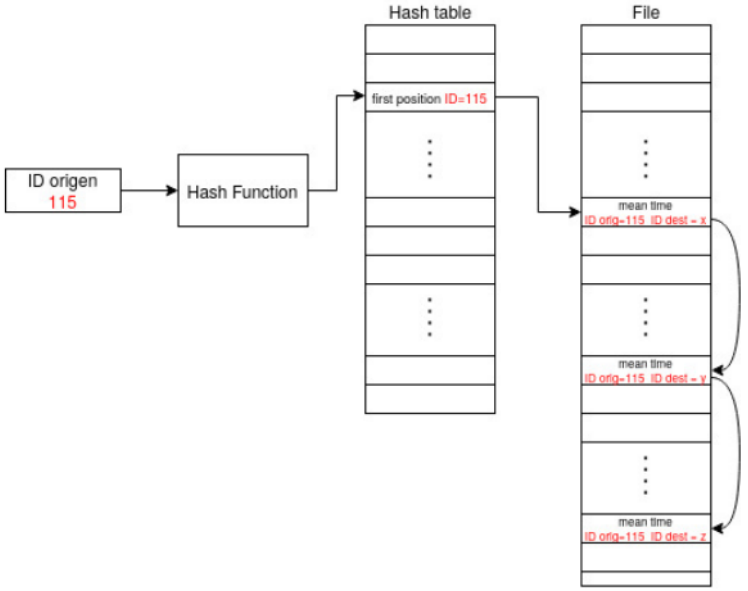
```
Bienvenido

1. Ingresar origen
2. Ingresar destino
3. Ingresar hora
4. Buscar tiempo de viaje medio
5. Salir
```

Este permite ingresar los parámetros necesarios para la búsqueda, los cuales quedan guardados en un array (opciones 1,2,3), finalmente la opción 4 permite realizar la conexión con el servidor y envía dicho array al servidor para que la búsqueda pueda ser realizada, por otro lado, la opción 5 termina la ejecución del cliente.

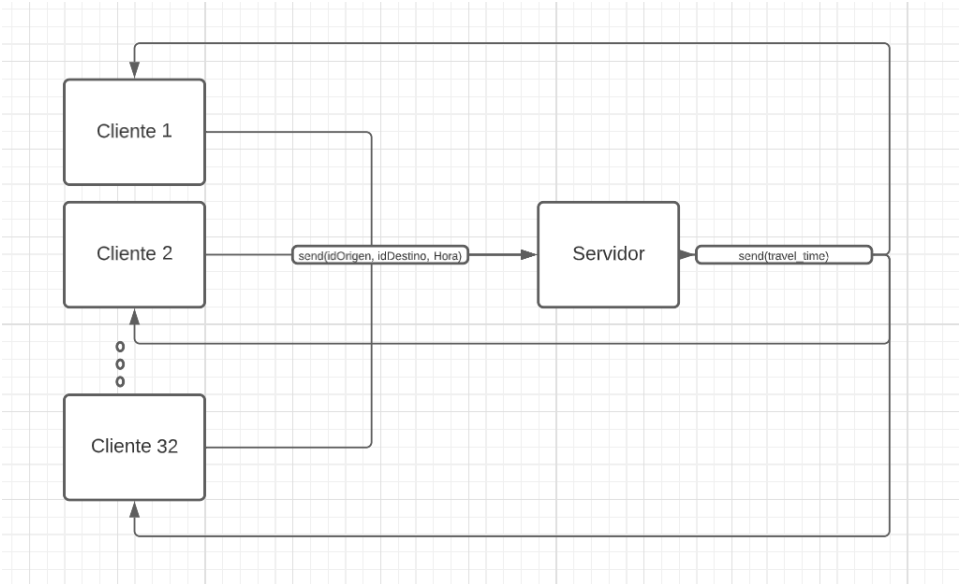
Servidor: El servidor posee acceso a los archivos binarios que se encuentran en la carpeta binGen, dichos binarios pasan previamente por un preprocesamiento (inciso 1 del manual de uso) el cual toma el archivo .csv de los tiempos de viaje y guarda en una columna adicional la información de dado un idOrigen donde puedo encontrar el siguiente con el mismo idOrigen, esto permite realizar saltos en el archivo y realizar una búsqueda muchísimo más rápida (bogotaTimes.bin). También hay otra tabla la cual me indica dado un idOrigen, donde encuentro el primero en el archivo bogotaTimes.bin (hashTable.bin). Nótese que al decir “donde” en un archivo binario nos referimos a su posición dentro del archivo.

El siguiente esquema muestra la funcionalidad de la búsqueda en el archivo .bin mostrando el paso a paso desde la función hash que indica en qué posición se encuentra la primera coincidencia del idOrigen con el que se realiza la búsqueda. Luego el archivo .bin adquiere una columna adicional que permite ubicar la siguiente coincidencia del idOrigen.

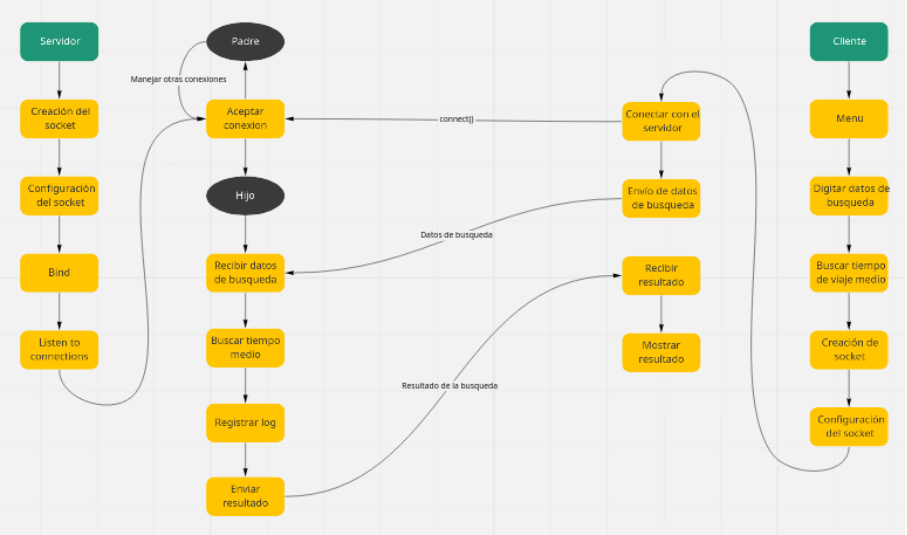


Finalmente, el servidor posee la funcionalidad search, la cual busca en el archivo bogotaTimes.bin el tiempo de viaje que desea el cliente y esta es enviada a dicho cliente.

A continuación, podemos encontrar un diagrama de comunicaciones que ilustra este proceso.



Por último, tenemos el diagrama de bloques:



Conclusiones:

- Esta práctica nos permitió desarrollar un sistema servidor multicliente en el cual se pueden conectar hasta 32 clientes simultáneamente por medio de paso de mensajes

(servidor-cliente) y a su vez la creación de tablas hash y manejo de archivos para similar de manera efectiva un sistema de búsqueda con datos reales.