**UNIVERSIDAD SIMÓN BOLÍVAR**

**DEPARTAMENTO DE COMPUTACIÓN Y TECNOLOGÍA DE LA INFORMACIÓN**

**CI3825 – SISTEMAS DE OPERACIÓN I**

**Trimestre:** Enero - Marzo 2023

**Profesor:** Fernando Torre

**Informe del Proyecto #1**

Transporte USB

**Estudiantes:**

José González 15-10627

Ana Santos. 17-10602

Sartenejas, Julio del 2022

**Introducción**

En la búsqueda que los estudiantes de Sistemas de Operación I aprendan sobre el funcionamiento de los procesos, semáforo, los hilos y la comunicación de estos a través del uso de pipes en el Lenguaje de Programación C, se plantea la implementación de un programa que simule las necesidades del servicio de transporte diario de la Universidad Simón Bolívar. El objetivo de este programa es lograr establecer si la capacidad de las unidades de transporte es suficiente para complacer la demanda estudiantil.

En el siguiente informe encontrará, los detalles de implementación, las decisiones de diseño, las evaluaciones de las configuraciones y el cómo se construyó la solución propuesta para el desarrollo del programa sobre la demanda del servicio de este transporte universitario.

1. **Decisiones de diseño:**
   1. Estructura de Datos y Optimizaciones:

Para optimizar el programa se decidió realizar dos archivos, uno destinado para funciones que facilitan la lectura de los archivos y otro en el que está contenido el programa principal del proyecto, *readFile.c* y *simutransusb.c,* respectivamente.

En *simutransusb.c*, se usan como estructuras principales un float para guardar el tiempo que dura la simulación, donde se toma por defecto 0,25 segundos; dos apuntadores tipo char para guardar los archivos que almacenan los datos de la carga de autobuses y las demandas de las paradas, donde para la carga se toma por defecto *carga.csv.*

En *readFile.c*, se definen los procedimientos *trimWord, readCsv, readTxt* y *getStop* para llevar a cabo las lecturas de los archivos que contienen los datos:

* El procedimiento *trimWordk* toma como entrada dos apuntadores tipo char con la finalidad de eliminar los espacios en blanco de una de las variables para guardarla en la otra.
* El procedimiento *readCsv* toma como entrada un apuntador tipo char, crear una variable apuntador tipo FILE, un apuntador tipo char , un arreglo tipo char, un arreglo inicial tipo int de 25 casillas y un int iterador. El objetivo de este procedimiento es leer el archivo .csv para guardar las horas de servicio de la parada en el arreglo y almacenar los demás datos en una estructura stop que representa las paradas de autobuses. El procedimiento *readTxt* funciona de manera similar sólo que el objetivo es leer un archivo txt, donde se realiza una metodología interna para leer la capacidad en el archivo dado el formato Indicado en las instrucciones.
* El procedimiento *getStop* es propio de las estructuras stop, tomando como entrada un apuntador tipo char que almacena el código referente a la parada. El objetivo de esta función es conseguir todos los datos en cuanto a la carga según el código asociado.

Además, se cuenta con un archivo *readFile.h,* donde se definen las estructuras necesarias para almacenar y trabajar los datos:

* La estructura *passengerLoad* es utilizada para almacenar el tiempo y la carga de un transporte/viaje de una parada en una hora determinada.
* La estructura *bus* es utilizada para representar a los autobuses en las paradas. Se guarda en ellos la cantidad de espacio que poseen, la parada a la que se dirigen/pertenecen, su hora de salida y la duración del recorrido.
* La estructura *stop* representa a la parada Se guarda en ellos la cantidad de espacio que poseen, la parada a la que se dirigen/pertenecen, su hora de salida y la duración del recorrido.
  1. Dificultades encontradas y soluciones realizadas:

Las dificultades encontradas en este proyecto estuvieron asociadas hacia el almacenamiento de los datos recibidos como entrada para poder realizar el estudio de la suficiencia del servicio de transporte. Los estudiantes no consiguieron con éxito encontrar unas estructuras con las que trabajar asociando los datos entre ambos archivos.

1. **Evaluación de las configuraciones:**
   1. Resultados obtenidos:

Para los siguientes cuatro casos de combinaciones de los archivos, se logró leer, separar y guardar los datos en las respectivas y mencionadas estructuras, en el apartado anterior.

* + 1. Combinación *servicio.txt* y *carga.csv*:
    2. Combinación *servicio2007.txt* y *carga2007.csv*:
    3. Combinación *servicio2007.txt* y *carga.csv*:
    4. Combinación *servicio.txt* y *carga2007.csv*:

**Conclusiones**

Al realizar este trabajo los estudiantes lograron investigar a profundidad sobre las funciones y herramientas de C para trabajar pipes y los hilos de procedimiento, aunque no se lograra el éxito de este proyecto. Se realizó el uso de funciones como *fork, semget, semctl, semop* yde las librerías como *pthread.h, unistd.h, semaphore.h,* para entender cómo se deben manejar los temas objetivos, pipes, hilos y semáforo, respectivamente.

Finalmente, no se pudo comprobar si el servicio de autobuses de la USB es suficiente o no con los cuatro archivos dados por el profesor para la medición de resultados, aunque se logró hacer la lectura de la información y almacenarlas en distintas estructuras.

Como recomendación para próximos trabajas, se debe intentar tener un mayor entendimiento del enunciado y del problema a resolver para poder completar con éxito los objetivos planteados.