

Fundamentos de los Sistemas Operativos

Ficha de entrega de práctica 2

Grupo de prácticas*:

Miembro 1: Romen Adama Caetano Ramirez

Número de la práctica*: 2/Revisión

Fecha de entrega*: 24/05/2023

Descripción del trabajo realizado*

Descripción del trabajo realizado

El presente informe describe las modificaciones realizadas en la práctica 2 previamente entregada.

En primer lugar, se solicitó la ampliación del archivo sala.c proporcionado en la entrega inicial. Dicho programa, denominado sala.c, es una aplicación que permite crear, gestionar y almacenar información sobre una sala (API). La estructura Sala almacena datos como los asientos, la capacidad, los asientos libres y los ocupados. Para esta segunda entrega, se han incorporado las siguientes funciones que permiten acceder al archivo, escribirlo y guardarlo:

- 1. guarda_estado_sala: Esta función se encarga de guardar el estado actual de la sala en un archivo especificado. En primer lugar, verifica si el archivo existe y lo abre en modo lectura/escritura, creándolo en caso de que no exista. A continuación, reserva memoria para un búfer de tamaño 10. Escribe la capacidad de la sala en el archivo y luego escribe el ID de cada cliente en los asientos correspondientes.*
- 2. recupera_estado_sala: Esta función recupera el estado de la sala a partir de la información almacenada en un archivo especificado. Verifica la existencia del archivo y lo abre en modo lectura. Lee la capacidad de la sala desde el archivo y crea una sala con dicha capacidad. A continuación, lee los ID de los clientes desde el archivo y los guarda en el array de asientos de la sala.*
- 3. guarda_estadoparcial_sala: Esta función guarda el estado de un conjunto de asientos de la sala en un archivo preexistente. Verifica si el archivo existe y lo abre en modo lectura/escritura. Luego, reserva memoria para un búfer de tamaño 10 y salta 10 bytes en el archivo. Recorre los asientos de la sala y, si el asiento se encuentra en el conjunto de asientos especificado, escribe el ID del cliente en el archivo.*
- 4. recupera_estadoparcial_sala: Esta función recupera el estado de un conjunto de asientos de la sala desde un archivo preexistente. Verifica la existencia del archivo y lo abre en modo lectura. Luego, reserva memoria para un búfer de tamaño 10 y salta 10 bytes en el archivo. Recorre los asientos de la sala y, si el asiento se encuentra en el conjunto de asientos especificado, lee el ID del cliente desde el archivo y actualiza el estado del asiento en la sala.*

En segundo lugar, se solicitó el desarrollo del programa misala.c.

El programa "misala" permite realizar diversas operaciones sobre los asientos de una sala. Se invoca con la siguiente sintaxis general: "misala orden argumentos".

El punto central del programa es la función "main()", que desempeña un papel fundamental en la ejecución y control del flujo del programa. En esta función, se declaran y se inicializan las variables necesarias para el funcionamiento del programa, como contadores, flags o estructuras de datos.

Además, se utiliza la función "getopt()" para procesar los argumentos de línea de comandos proporcionados por el usuario al momento de ejecutar el programa. Esta función permite analizar y

| | |
|--|--------------------------------------|
| Grupo de prácticas*: Miembro 1: Romen Adama Caetano Ramirez | |
| Número de la práctica*: 2/Revisión | Fecha de entrega*: 24/05/2023 |
| <p><i>extraer las opciones y argumentos ingresados, lo que facilita el establecimiento de los modos de operación del programa y la toma de decisiones correspondientes.</i></p> <p><i>El programa admite varios modos de operación, como la creación de una sala con una capacidad especificada, la reserva o anulación de asientos, y la obtención del estado parcial o completo de la sala, entre otros. Estos modos se activan mediante opciones de línea de comandos, como "-c" (crear sala), "-f" (fichero), "-n" (reserva), "-a" (anulación) y " " (estado).</i></p> | |
| Horas de trabajo invertidas* Miembro 1: 1 semana. | |
| Cómo probar el trabajo* <ul style="list-style-type: none"> Descomprimir él .zip y mediante comandos de terminal (cd) desplazarse hasta el directorio: .../pract2 (seguramente sea el directorio por defecto al descomprimir) <p>Ejecutar el script desarrollado para facilitar la manipulación del programa: <small>(se encarga de compilar los archivos en caso de alguna modificación en los mismos)</small></p> <ul style="list-style-type: none"> bash script.sh <p>En el directorio /pract2/fuentes/ y ejecutar el programa misala. Existe varios modos:</p> <ul style="list-style-type: none"> <code>./misala crea -c 10 -f sala.txt</code> (crea un fichero de una sala con 10 asientos) <code>./misala crea -c 15 -f sala.txt -o</code> (modifica la sala previamente creada con el comando -o) <code>./misala reserva -f sala.txt -n 3 1 2 2</code> (error no se puede reservar asientos con el mismo identificador) <code>./misala reserva -f sala.txt -n 3 1 2 3</code> (comando de reserva correcto) <code>./misala estado -f sala.txt</code> (muestra el estado de la sala) <code>./misala anula -f sala.txt -a 1 2</code> (anula la reserva de los asientos 1 y 2) <p>En el directorio /pract2/fuentes/ y ejecutar el programa de funciones parciales. Este programa es para el testeo de las funciones parciales.</p> <ul style="list-style-type: none"> <code>./parciales</code> | |
| Incidencias <ul style="list-style-type: none"> Como primer intento y como se comentó en mi primera entrega, estaba trabajando con un fichero que trabajaba con caracteres, esto complico el desarrollo de la práctica, y por tanto migue a un "buffer de enteros". No podía aplicar un modo_parcial en misala para el testeo de las funciones parciales, no encontré el motivo, por tanto, desarrollé una clase y un programa aparte. | |
| Comentarios <p>Las capturas, se encuentran dentro del .zip, en una carpeta llamada MEDIA, en ella se adjuntan las capturas del funcionamiento de los diferentes modos que se encuentra para misala y parciales.</p> <p>Link del repositorio para ver la evolución del proyecto:</p> <p>https://github.com/Romen-Adama-Dev/FSO.git</p> | |