

Fundamentos de los Sistemas Operativos

Ficha de entrega de práctica

*: campo obligatorio

IMPORTANTE: esta ficha no debe superar las DOS PÁGINAS de extensión

Grupo de prácticas*:

Miembro 1: Romen Adama Caetano Ramirez

Número de la práctica*: 2

Fecha de entrega*: 20 de Abril

Descripción del trabajo realizado*

En esta práctica, se debe evolucionar una API previamente desarrollada, para que la información sobre el estado de ocupación de una sala de teatro persista en un archivo. Para ello, se deben agregar operaciones para guardar y recuperar el estado de la sala. El programa debe permitir guardar el estado actual de la sala en un archivo y recuperarlo posteriormente, así como también guardar el estado parcial de un conjunto de asientos de la sala en un archivo y recuperarlo. Además, se debe crear un programa llamado "misala" que permita realizar diferentes operaciones sobre los asientos de una sala, como crear una sala con una capacidad determinada y guardar el estado inicial en un archivo, reservar un número de asientos para personas específicas y anular reservas de asientos. El programa debe validar los argumentos de entrada y manejar errores apropiadamente.

En sala.c se desarrolló:

- La función guarda_estado_sala guarda la capacidad de la sala y el estado de cada uno de sus asientos en un archivo. Para hacer esto, se abre el archivo con permisos de lectura y escritura y se escribe la capacidad de la sala seguida de cada uno de los asientos. Para escribir los asientos, se utiliza un ciclo que recorre el arreglo de asientos y escribe cada uno de ellos en el archivo. Finalmente, se libera la memoria utilizada por el puntero auxiliar y se cierra el archivo.
- La función recupera_estado_sala recupera el estado de la sala a partir del archivo guardado anteriormente. Para hacer esto, se abre el archivo con permisos de solo lectura y se lee la capacidad de la sala. Después, se crea una sala con esa capacidad y se leen los asientos de la sala. Para leer los asientos, se utiliza un ciclo que recorre el arreglo de asientos y lee cada uno de ellos del archivo. Finalmente, se libera la memoria utilizada por el puntero auxiliar y se cierra el archivo.

Las funciones siguientes están a medio desarrollar, pendientes de la posibilidad de extender la entrega.

- Las funciones guarda_estadoparcial_sala y recupera_estadoparcial_sala guardan y recuperan, respectivamente, el estado parcial de la sala en un archivo. Para hacer esto, se abre el archivo con permisos de escritura o lectura, según sea el caso, y se escribe o lee la cantidad de asientos especificados en la variable num_asientos del arreglo.

Programa misala.c:

Este código en C implementa un programa que permite la creación y gestión de una sala con asientos numerados y permite la reserva y anulación de dichos asientos por parte de los usuarios. El programa toma argumentos de línea de comandos para especificar las acciones a realizar y las opciones correspondientes, tales como la capacidad de la sala, la ruta del archivo donde se almacenará el estado de la sala, y el número de asientos y personas involucradas en una reserva o anulación. Además, el programa utiliza la biblioteca "sala.h" para definir la estructura (sala.c) y funciones necesarias para manipular la sala y su estado.

La primera parte del programa establece varias constantes y la definición de un enum con los valores de las opciones válidas que el usuario puede utilizar. También encontramos dos funciones, "comprueba_error" e "imprime_ayuda", que se encargan de mostrar los errores producidos por el programa y de mostrar la ayuda del programa en caso de introducir una opción no válida.

La función principal, "main", comienza por declarar una serie de variables y flags para cada modo de funcionamiento del programa. Estos son:

1. "modo_crea_sala"
2. "modo_fichero"
3. "modo_sobreescribir"
4. "modo_reserva"
5. "modo_anula" (pendiente de desarrollar)
6. "modo_estado"

Grupo de prácticas*: Miembro 1: Romen Adama Caetano Ramirez	
Número de la práctica*: 2	Fecha de entrega*: 20 de Abril
<p>7. "cap (<i>capacidad, pero no se puede declarar con ese nombre para evitar conflictos con la funcion definida en sala.c</i>)"</p> <p>8. "ruta_fichero"</p> <p>9. "num_asientos"</p> <p>10. "num_personas"</p> <p>La mayoría de estas variables se utilizan para almacenar información sobre la sala, los asientos y los archivos.</p> <p>El programa comprueba si se ha introducido una opción válida para determinar qué acción se va a realizar. En caso contrario, se muestra un mensaje de ayuda al usuario.</p> <p>El programa utiliza una serie de flags para determinar qué acción se va a realizar y, posteriormente, se utilizan las opciones del usuario (mediante getopt) para establecer la capacidad de la sala, la ruta del fichero, el número de asientos y las personas que han reservado dichos asientos.</p>	
Horas de trabajo invertidas* Miembro 1: 20 horas	
<p>Cómo probar el trabajo*</p> <p>Descomprimir el .zip y mediante comandos de terminal (cd) desplazarse hasta el directorio:</p> <p>.../pract2 (seguramente sea el directorio por defecto al descomprimir)</p> <p>Ejecutar el script desarrollado para facilitar la manipulación del programa:</p> <p>(se encarga de compilar los archivos sala.c, sala.h y misala.c en caso de alguna modificación en los mismos)</p> <p>bash script.sh = Contenido del script:</p> <ul style="list-style-type: none"> • #!/bin/bash • gcc -c lib/salas/sala.c -o lib/salas/sala.o • ar -crs lib/lib sala.a lib/salas/sala.o • #gcc fuentes/test_sala.c -lsala -Llib -o fuentes/test_sala • gcc fuentes/misala.c -lsala -Llib -o fuentes/misala • #fuentes/misala <p>Dirigirse hacia el programa misala.c:</p> <p>cd .../pract2/fuentes</p> <p>Una vez ahí ejecutar en funcion de lo que se desee:</p> <p>Para crear un fichero:</p> <p><code>./misala crea -c 10 -f misala.txt</code></p> <p>En Caso que exista, añadir -o al final para overwrite:</p> <p><code>./misala crea -c 15 -f misala.txt -o</code></p> <p>Para reservar un asiento (modificar el archivo creado):</p> <p><code>./misala reserva -f misala.txt -n 1 20</code></p> <p>Para mostrar los asientos reservados (visualizar el contenido del archivo creado):</p> <p><code>./misala estado -f misala.txt</code></p>	
<p>Incidencias</p> <p>Errores al ejecutar el código en el Macbook Air, incompatibilidad con ciertos procesos de las librerías incluidas. Migración a un sistema Linux basado en Ubuntu.</p> <p>No suficiente tiempo debido al trabajo de las prácticas externas, y resto de las 4 asignaturas que curso, espero poder optar por un "no apto" con opción de seguir con el desarrollo de la práctica.</p> <p>Tareas pendientes:</p> <p>Modo "anula" y las funciones parciales.</p>	
<p>Comentarios</p> <p>Para ver el contenido completo y el desarrollo evolutivo ver el gitub personal.</p> <p>https://github.com/Romen-Adama-Dev/FSQ</p>	