

# Fundamentos de los Sistemas Operativos

## Ficha de entrega de práctica

\*: campo obligatorio

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

<b>Grupo de prácticas*: 42</b>	
<b>Miembro 1:</b> M <sup>a</sup> Elena Navarro Santana	
<b>Número de la práctica*: 2</b>	<b>Fecha de entrega*: 05/05/2023</b>
<b>Descripción del trabajo realizado*</b>	
<p><b><i>guarda_estado_sala</i></b> → se encarga de guardar el estado actual de la sala en un archivo especificado por la ruta. Si el archivo ya existe, su contenido se sobrescribe. Si no existe, se crea uno nuevo con los permisos 00666. La función devuelve 0 si todo se realiza correctamente y -1 en caso de que se produzca algún error.</p> <p>Acto seguido, se escribe la capacidad de la sala en el archivo. Para ello se utiliza la función "contar_digitos" para contar el número de dígitos del número de asientos de la sala y guardarlo en un búfer mediante la función "sprintf". Luego se usa la función "write" para escribir la capacidad en el archivo.</p> <p>A continuación, el estado de cada asiento se escribe en el archivo. Para hacer esto, se ejecuta un bucle desde 0 hasta el número de asientos en la sala y guarda el estado de cada asiento en un búfer. Se utiliza la función "write" para escribir el búfer en un archivo. Por último, se cierra el fichero y si hay algún error se devuelve un -1, si todo sale bien devuelve un 0.</p> <p><b><i>recupera_estado_sala</i></b> → se encarga de recuperar el estado de la sala a partir de la información guardada en un fichero. Esta función abre el archivo, lee su contenido y lo procesa para establecer la capacidad de la sala y la capacidad de cada uno de sus asientos.</p> <p>Después de abrir el archivo, la función comienza a leer su contenido carácter por carácter utilizando la función read(). Si el carácter leído no es un salto de línea o una coma, se trata como un número y se almacena en la variable tempNumber. Si se encuentra un salto de línea después del primer número, se establece la capacidad de la sala mediante la función crea_sala(). Si se encuentra una coma, se guarda el valor actual de tempNumber en la variable auxIndex y se reinicia tempNumber. Si se encuentra un salto de línea después de una coma, se almacena el valor actual de tempNumber en el índice auxIndex del arreglo capacidad_sala, se reinician tempNumber y auxIndex, y se incrementa num_lineas.</p> <p>Después de que el archivo se haya leído completamente, la función verifica si el número de líneas en el archivo coincide con la capacidad actual de la sala. Si no coinciden, la función devuelve un error. Finalmente, la función cierra el archivo y devuelve 0 si todo ha ido bien.</p> <p>Es importante destacar que esta función asume que el archivo de entrada contiene información válida y coherente con el estado de la sala actual. Si el archivo contiene información incorrecta, la función puede fallar o producir resultados inesperados.</p> <p><b><i>guarda_estadoparcial_sala</i></b> → esta función se encarga de cargar los datos actuales de la sala desde el archivo. Luego, se abre el archivo en modo escritura con permisos 0666. Después, se escribe la capacidad de la sala en el archivo.</p> <p>A continuación, se itera sobre todos los asientos de la sala y se verifica si cada uno de ellos se encuentra en el conjunto de identificadores de asientos a guardar. Si un asiento se encuentra en dicho conjunto, se escribe su estado actual (la cantidad de personas que lo ocupan) en el archivo. En caso contrario, se escribe el estado previamente almacenado en el archivo.</p> <p>Por último, se cierra el archivo y se comprueba si se produce algún error al hacerlo. La función devuelve 0 si</p>	

<b>Grupo de prácticas*: 42</b> <b>Miembro 1: M<sup>a</sup> Elena Navarro Santana</b>	
<b>Número de la práctica*: 2</b>	<b>Fecha de entrega*: 05/05/2023</b>
<p>todo sale bien, y -1 en caso de producirse algún error al abrir, escribir o cerrar el archivo.</p> <p><b>recupera_estadoparcial_sala</b> → esta función recupera el estado de un conjunto de asientos de la sala a partir de la información guardada en un fichero.</p> <p>En primer lugar, la función abre el archivo en modo lectura y si se produce algún error, devuelve -1. A continuación, se recupera la capacidad de la sala del archivo leyendo los caracteres y almacenando el número leído en una variable temporal. Cuando se encuentra un salto de línea, se crea la sala con la capacidad recuperada.</p> <p>Después, se recupera la capacidad parcial de cada uno de los asientos especificados en el array de identificadores. Si el identificador del asiento se encuentra en el array, se marca la existencia del mismo. Cuando se encuentra un salto de línea y se tiene un índice de asiento válido, se actualiza la capacidad de la sala para el asiento correspondiente.</p> <p>Finalmente, si se produce algún error al leer o cerrar el archivo, se envía el error por el canal correspondiente. Si todo sale bien, se devuelve 0.</p> <p>Es importante mencionar que la función asume que la capacidad de la sala y la capacidad parcial de los asientos están almacenados en el archivo en un formato determinado y que los identificadores de los asientos especificados en el array están ordenados en el mismo orden en el que aparecen en el archivo.</p>	
<b>Horas de trabajo invertidas* Miembro 1: 20 horas</b>	
<p><b>Cómo probar el trabajo*</b></p> <p>Se tienen 3 directorios dentro del trabajo: lib, fuente y cabeceras.</p> <p>Dentro de cabeceras esta sala.h</p> <p>Dentro de /lib está el directorio /sala y dentro de este directorio estará sala.h, sala.o y sala.c</p> <p>Compilamos: <b>gcc -c sala.c -o sala.o</b></p> <p>En /lib compilamos la biblioteca: <b>ar -crs libsala.a sala/sala.o</b></p> <p>En fuentes tendremos las pruebas: misala.c, para compilarlo simplemente: <b>gcc misala.c -o misala -lsala -L ../lib</b></p> <p>Y ejecutamos de diferentes formas:</p> <p><b>./misala crea -f archivo.txt -c capacidad</b> → Este comando crea la sala, se puede poner el nombre que se quiera al archivo.txt, por ejemplo prueba.txt y en capacidad se especifica un número positivo.</p> <p><b>./misala crea -f archivo.txt -c capacidad -o</b> → En este comando se añade -o que es para sobrescribir el archivo en caso de que se quiera más capacidad en el mismo archivo, por ejemplo.</p> <p><b>./misala reserva -f archivo.txt -n número id_asiento...</b> → Este comando reserva los asientos de la sala, si se pone un número que no coincide con el número de id_asiento salta un error. Si la sala no tiene asientos disponibles se notificará que no se pueden reservar los asientos.</p> <p><b>./misala anula -f archivo.txt -a id_asiento...</b> → Este comando anula los asientos que se pasan por el parámetro -a, los inicializa a 0.</p> <p><b>./misala estado -f archivo.txt</b> → Este comando muestra el estado actual de la sala.</p> <p><b>./misala confirmaciones -f archivo.txt -a id_asientos...</b> → Este comando se le pasa los id_asientos que han confirmado asistencia y los demás se ponen a 0.</p>	
<p><b>Incidencias</b></p> <p>Ninguna.</p>	
<p><b>Comentarios</b></p> <p>No se realiza reto.</p>	