

Ejercicios Integradores de Estructuras Enlazadas

Ejercicio 1

Una empresa desea realizar el resumen mensual de las horas trabajadas y horas de ausencia de sus empleados. Para realizar el cálculo cuenta con un archivo binario "HORAS_DIARIAS.dat" donde cada registro se compone de los siguientes datos:

- Legajo del empleado: 10 caracteres
- Día del mes: número de 1 a 31
- Horas: número entero de 1 a 24
- Ausencia: 1 o 0 (si es 1: las Horas son horas de ausencia, si es 0: las Horas son horas trabajadas)

El archivo se encuentra desordenado y por haber distintos turnos puede haber más de un registro por día de un mismo empleado. Se solicita:

- Definir la estrategia y estructuras necesarias para la lectura del archivo y realización de los puntos b. y c.
- Imprimir por pantalla el listado de las horas imputadas por cada empleado para el mes, ordenado por legajo en forma creciente. imprimiendo:

| Legajo | Hs. Trabajadas | Hs. Ausencia | % Ausentismo (HA / HT + HA) |
|--------|----------------|--------------|-----------------------------|
|--------|----------------|--------------|-----------------------------|

- Indicar el día del mes con mayor cantidad de Horas de Ausencia
- Indicar el legajo con mayor Ausentismo

Ejercicio 2

Se desea implementar una aplicación que permita realizar el balanceo de carga de varios servidores web. Para ello se cuenta con N nodos, todos iguales, que atienden los pedidos que le deriva el balanceador de carga.

Cuando llega un pedido el balanceador deberá asignar el mismo al nodo que posea menos pedidos en espera.

Cada nodo tiene una capacidad M para encolar pedidos. Si al momento de asignar un pedido, todos los nodos se encuentran al máximo de su capacidad, deberá generarse un nuevo nodo.

En caso de encontrar varios nodos con la misma cantidad de pedidos, aplicar una política aleatoria para seleccionar uno de los nodos.

Los nodos utilizan una política FIFO para procesar los pedidos.

Cada pedido posee:

- Id
- Cantidad de ciclos de procesador a utilizar.
- Tipo: CPU | E/S

Se pide:

- Definir estrategia y estructuras para soportar las definiciones anteriores
- Desarrollar el procedimiento para agregar un nuevo proceso
- Desarrollar el procedimiento que permita “procesar” uno de los procesos que se encuentran en el balanceador de carga. El procedimiento debe recibir la cantidad de ciclos que pudo aplicar
- Desarrollar un procedimiento que de un reporte de status informando:
 - % de procesos CPU y E/S de cada nodo
 - El nodo que más procesos de tipo CPU procesó
 - El nodo que procesó el proceso de E/S más largo

Ejercicio 3

Una empresa ferroviaria desea procesar los archivos históricos de venta de pasajes de 2 estaciones que producto de una remodelación fueron unificadas.

Para esto cuenta con 2 archivos binarios, uno por cada estación, donde cada registro describe una venta de un pasaje. Cada registro contiene

- Número de boleto (long int)
- Fecha en formato "DD-MM-YYYY HH:MM:SS"
- Id de estación (short int)
- Precio (float)

Los archivos se encuentran ordenados por fecha.

Se pide:

1. Generar un nuevo archivo de texto, que contenga los registros de los 2 archivos binarios, ordenado por fecha. La particularidad es que del campo fecha solo se requiere la fecha DD-MM-YYYY
2. A partir del archivo de texto generado en el punto 1, emitir un listado con la siguiente forma:

Estación: xxx

Año: 9999. Mes:99

Total mensual: 999.99

Año: 9999. Mes: 99

Total mensual: 999.99

Estación: yyy

Año: 9999. Mes:99

Total mensual: 999.99

Año: 9999. Mes: 99

Total mensual: 999.99

Ejercicio 4

Dados 2 archivos binarios, donde cada registro posee una única palabra de 256 caracteres como máximo. Se pide generar un archivo de texto con las siguientes características:

1. Debe contener las palabras que existan en ambos archivos y que además sean palíndromos
2. El archivo no debe contener duplicados
3. El archivo debe estar ordenado de forma alfabética ascendente

Codificar además la función **esPalindromo** utilizando sólo sintaxis de punteros

Ejercicio 5

Una facultad requiere de una pieza de software que le permita gestionar a oos alumnos inscriptos.

La universidad cuenta actualmente con 1000 alumnos, de cada uno posee los siguientes datos:

- Legajo, 7 caracteres
- Carrera: 1 caracter
- Nombre, 30 caracteres
- Apellido, 30 caracteres
- Email, 30 caracteres
- Teléfono, 10 caracteres

El legajo es un número entero entre 1000 y 2000.

Las carreras están nombradas según el siguiente listado

- K, ingeniería en sistemas
- C, ingeniería civil
- P ingeniería electrónica
- D ingeniería mecánica

Los datos antes mencionados se encuentran almacenados en el archivo alumnos.dat

Los datos de las carreras se encuentran almacenados en el archivo carreras.txt, donde además del código se encuentra el nombre y descripción de la carrera.

Por otra parte, la facultad posee las notas de cada materia de los alumnos en distintos archivos, uno por cada materia. El nombre de cada archivo indica la materia.

Estos archivos poseen el número, fecha de cada parcial y el legajo de cada alumno. Tener en cuenta que estos archivos se encuentran desordenados y que sólo hay 2 parciales por cada alumno.

Se pide:

1. Imprimir el archivo de notas de forma ordenada, utilizando el legajo como criterio. Restricción: sólo puede ser recorrido secuencialmente 1 vez.
2. Calcular y guardar el promedio de las notas de los alumnos, ordenado por legajo de forma ascendente. Restricción: sólo posee 8000 bytes de espacio en disco.

Ejercicio 6

Desarrollar una aplicación que permita simular la atención de clientes en la caja de un banco.

Para ello suponga que el banco cuenta con 2 colas, 1 para clientes premium y otra para clientes comunes.

Cada vez que un cliente ingresa al banco, ingresa sus datos y la operación que desea realizar; a continuación el sistema le indica en qué cola debe esperar. En caso de que el cliente sea una mujer y esté embarazada, deberá pasar al principio de la fila.

Cada vez que el cajero del banco termina de atender a un cliente y llama a otro, el sistema deberá asignarle el turno según las siguientes reglas:

- Si la cola de clientes premium tiene más de 2 clientes en espera, deberá asignar el turno a un cliente premium
- Cuando la cola de clientes premium posea menos de 3, la asignación del turno deberá alternar entre ambas colas

- Si en cualquiera de las 2 colas se encuentra una mujer embarazada, deberá tener prioridad por sobre todo el resto de los clientes.

Se pide:

- Cargar ambas colas contemplando los casos descritos
- Desarrollar un procedimiento que permita simular la atención del cajero a los clientes. Por cada cliente atendido se deberá generar una entrada en un archivo de texto con los datos del cliente.