FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Algoritmos y Programación I
Curso Mendez

TP3 - Hogwadmin

Fecha Presentacion	12/06/2018
Fecha Entrega	24/06/2018

Historia

En Hogwarts, es sabido que al comenzar el año, los alumnos nuevos son asignados a una de las cuatro casas según sus atributos intrínsecos. Por otro lado, al terminar el año, la escuela entera ofrece el banquete de fin de año a la casa que mayor puntaje haya acumulado durante el año, según su comportamiento, sus logros y su participación en distintas actividades de la escuela.

La administración de la escuela es conocida por su intrincado sistema de gestión de la información, completamente digitalizado, que previene ser modificado por medios mágicos, y en esta ocasión, nos ha pedido que les echemos una mano para manejar los datos de sus inscriptos.

Objetivo

Que el alumno analice el problema dado, determine los requerimientos pedidos y diseñe una solución acorde.

Que el alumno se familiarice con el manejo de archivos binarios y de texto, valiéndose de todos los elementos aprendidos en el curso.

Que el alumno profundice aún más las bases de las buenas prácticas de programación.

Especificaciones Tecnicas

La administración de la escuela lleva cuenta de cuántos puntos ganó o perdió cada alumno en cada momento en que le fueron asignados en un archivo de bitácora, en formato texto.

Por otro lado, cuenta con el listado de alumnos inscriptos para el año lectivo, un archivo binario ordenado por casa, año y padrón de cada alumno que se crea al comenzar el año, a partir de modificar el del año anterior.

Requerimientos

Se debe desarrollar un programa llamado *hogwadmin*. Su propósito principal será el de asistir al personal administrativo de la escuela en sus labores de inicio y fin de año y deberá proporcionar tres prestaciones elementales:

- Generar el archivo de alumnos para el año que comienza
- Actualizar el archivo del año actual
- Contabilizar los puntos de cada alumno al final del año
- Calcular la casa ganadora al final del año.

A tal efecto, el programa se ejecutará el programa por línea de comando enviando los parámetros necesarios para cada modo de uso.

Prestaciones

El primer parámetro enviado a este programa debe ser el identificador de operación según se define a continuación:

- Generación del archivo de inicio de año
  - Comando identificador: **limpiar**
  - Argumentos
    - Archivo de datos del año anterior
    - Archivo a generar: ruta al archivo que se quiere generar para el año
  - Objetivo: Remover a los alumnos del año egresado (7), incrementar en 1 el año de todos los demás alumnos en carrera. Además deja en cero los puntajes acumulados del año anterior.
  - Ejemplos

```
$ ./hogwadmin limpiar /data/2017_alumnos.dat /data/2018_alumnos.dat
$ ./hogwadmin limpiar ~/admin/2017.dat ~/admin/2018.dat
$ ./hogwadmin limpiar alumnos-2017.dat alumnos-2018.dat
```

- Incorporación de ingresantes
  - Comando identificador: **actualizar**
  - Argumentos
    - Archivo de alumnos del año: el que se usará durante el corriente año
    - Archivo de nuevos ingresos: contendrá los nuevos estudiantes a incorporarse. Los nuevos estudiantes podrán ser alumnos de primer año, o de años mayores, en caso de que hubieran sido transferidos.
  - Objetivo: incorporar a los ingresantes al registro de alumnos. Recordar que los alumnos nuevos ingresarán con puntaje 0 (cero).
  - Ejemplos

```
$ ./hogwadmin actualizar /data/2018_alumnos.dat /tmp/2018_ingresos.txt
$ ./hogwadmin actualizar ~/admin/2018.dat ~/admin/nuevos-2018.txt
$ ./hogwadmin actualizar alumnos-2018.dat ingresos-2018.txt
```

- Aplicar puntos
  - Comando identificador: **aplicar**
  - Argumentos
    - Archivo de alumnos del año: el que se usará durante el corriente año
    - Bitácora de puntos: archivo de modificaciones de puntos
  - Objetivo: Modificar el archivo de alumnos con los puntos acumulados por cada alumno
  - Ejemplos

```
$ ./hogwadmin aplicar /data/2018_alumnos.dat /tmp/2018_puntos.txt
```

```
$ ./hogwadmin aplicar ~/admin/2018.dat ./puntos2018.txt
$ ./hogwadmin aplicar alumnos-2018.dat puntos.txt
```

- Calcular casa ganadora
    - Comando identificador: **resumir**
    - Argumentos
        - Archivo de alumnos del año: el que se usará durante el corriente año
    - Objetivo: Mostrar por año, y por casa, los puntos obtenidos, y un resumen con el nombre de la casa ganadora, su puntaje obtenido, y el alumno que más puntos sumó para la casa.
    - Ejemplos

```
$ ./hogwadmin resumir /data/2018_alumnos.dat
$ ./hogwadmin resumir ~/admin/2018.dat
$ ./hogwadmin resumir alumnos-2018.dat
```

- Ayuda
    - Comando identificador: **ayuda**
    - Argumentos: ninguno
    - Objetivo: Mostrar una ayuda indicando los comandos que puede recibir el programa, cómo se los invoca, argumentos que requieren, y su propósito u objetivo.
    - Ejemplo

```
$ ./hogwadmin ayuda
```

Archivos

alumnos.dat

Un archivo binario que contiene la información personal de los alumnos de la escuela. Cada registro de este archivo contiene la siguiente información.

1. Padrón: cadena de hasta 8 caracteres alfanuméricos;
2. Nombre: cadena de hasta 32 caracteres alfabéticos incluyendo el espacio ‘ ‘;
3. Casa: cadena de hasta 16 caracteres alfanuméricos indicando el nombre de la casa. Contendrá uno de los siguientes valores: **Gryffindor**, **Ravenclaw**, **Slytherin**, **Hufflepuff**.
4. Año: El año que cursa el alumno.
5. Puntos: la cantidad de puntos obtenidos por ese alumno para el año lectivo.

El archivo se encuentra ordenado ascendentemente por casa, año, y padrón y toda operación que se efectúe sobre el mismo debe mantener dicho orden. En caso de no existir, se deberá crear el archivo. Debe considerar la posibilidad de que inicialmente esté vacío. Si no existe, se deberá crear.

- bitacora.txt

Archivo de texto que contiene la información de los movimientos de puntos durante el año. Cada línea de este archivo contiene la siguiente información en campos separados por punto y coma (;):

1. Padrón: cadena de hasta 8 caracteres alfanuméricos.
2. Fecha: la fecha de esta ocurrencia representada en una cadena de texto con formato 'aaaa-MM-dd'
3. Puntos: un número entero (positivo o negativo) indicando los puntos sumados o restados por el alumno.
4. Comentarios: un texto libre de hasta 256 caracteres alfanuméricos donde el formador explica el motivo del incremento o decremento de puntos. No contiene el caracter separador (;).

El orden de este archivo es cronológico. Las entradas se registraron según fueron sucediendo.

- ingresos.txt

Archivo de texto que contiene sólo los datos de los alumnos que ingresan este año a la escuela. Cada línea de este archivo contiene la siguiente información separada por punto y coma (;):

1. Padrón: cadena de hasta 8 caracteres alfanuméricos.
2. Nombre: cadena de hasta 32 caracteres alfabéticos incluyendo el espacio ' '.
3. Casa: cadena de hasta 16 caracteres alfanuméricos indicando el nombre de la casa. Contendrá uno de los siguientes valores: **Gryffindor**, **Ravenclaw**, **Slytherin**, **Hufflepuff**.
4. Año: El año que cursará el alumno.

Este archivo está ordenado por casa, año y padrón, igual que alumnos.dat

Biblioteca admin

Se debe desarrollar una biblioteca admin, para resolver el núcleo de lo que se pide al programa. Esta biblioteca deberá contar con las siguientes funciones.

```
/**
 * Toma el archivo de un año, elimina los alumnos del último año (año 7)
 * e incrementa en 1 el año de todos los alumnos de años anteriores.
 * Devuelve la cantidad de alumnos que quedaron en el archivo destino.
 * Queda a criterio del alumno el uso de valores negativos para informar
 * errores.
 */
int pasar_de_anio(char* ruta_origen, char* ruta_destino);

/**
 * Aplica las actualizaciones del archivo de actualizaciones al archivo
 * de alumnos, incorporando a dicho padrón a todos los ingresantes.
 * Devuelve la cantidad de alumnos ingresados al padrón de alumnos.
```

```
 * Queda a criterio del alumno el uso de valores negativos para informar
 * errores.
 */
int actualizar_alumnos(char* ruta_alumnos, char* ruta_actualizaciones);

/**
 * Para cada entrada del archivo de bitácora, modifica la cantidad de
 * puntos acumulada durante el año por el alumno indicado en dicha
 * entrada.
 * Devuelve la cantidad de entradas de la bitácora procesadas.
 * Queda a criterio del alumno el uso de valores negativos para informar
 * errores.
 */
int aplicar(char* ruta_alumnos, char* ruta_bitácora);

/**
 * Muestra por pantalla un resumen de cantidad de puntos por cada año
 * dentro de una casa, y por cada casa. Incluye al final del resumen la
 * casa con la máxima cantidad de puntos (el nombre), los puntos
 * acumulados, y para el integrante de la casa que más puntos ganó para
 * esa casa ese año y los puntos que ese alumno sumó.
 * Devuelve una estructura con la información final del resumen:
 *    - Casa ganadora
 *    - Puntos
 *    - Padrón de alumno con más puntos
 *    - Puntos obtenidos por el alumno
 *
 * Se puede asumir que no hay empates.
 */
typedef struct resumen_casa {
    char casa[/* Defina la constante adecuada */];
    unsigned long puntos_casa;
    char padron[/* Defina la constante adecuada */];
    unsigned long puntos_alumno;
} resumen_casa_t;

resumen_casa_t resumir_puntos(char* ruta_alumnos, FILE* canal_impresion);
```

Aclaraciones complementarias

Los archivos que ingresa el usuario pueden no existir o no poderse acceder por cualquier motivo y esto deberá validarse. Pero los archivos que existan deberán estar bien formados.

Los registros de archivo binario tendrán que tener los campos indicados **en el orden** en que se indican. Los tipos deberán inferirse correctamente de su descripción.

Compilacion

Para compilar el trabajo se debe poder ejecutar la siguiente línea:

```
gcc hogwadmin.c admin.c -o hogwadmin -std=c99 -Wall -Wconversion -Werror
```

Condiciones de Aprobacion

El trabajo debe poder ser compilado sin problemas, deben estar desarrolladas todas las funciones y comportarse lo más cercano posible a los requerimientos que figuran en el enunciado.
La legibilidad y extensibilidad del código también serán tenidas en cuenta como en los trabajos anteriores. Un trabajo que funciona perfecto pero que es imposible de leer puede ir a reentrega directa.

.

Entrega

La entrega deberá incluir los archivos componentes del programa comprimidas en un archivo ZIP sin carpetas internas:

- hogwadmin.c
- admin.h
- admin.c

El zip puede tener cualquier nombre, pero no debe tener ninguna carpeta que contenga el código. Este archivo ZIP se deberá subir a su cuenta de la plataforma Kwyjibo en http://tps.algoritmos7540mendez.com.ar y quedar con estado **successful** y en la salida capturada de la ejecución debe decir Pruebas exitosas. Entrega aceptada.