

Instituto Tecnológico de Costa Rica
Administración de Tecnologías de
Información
Lenguajes de Programación
Prof. Andréi Fuentes L.

Tarea Programada 3

Erick Vargas Victor
Luis Diego Coto Mata
Mariam Ramírez Cordero

2013

Contenido

Descripción del problema	2
Diseño del programa	3
Librerías usadas.....	5
Análisis de Resultados.....	6
Manual de usuario	7
Conclusión personal	9
Bibliografía	10

Descripción del problema

Esta tarea consiste en desarrollar un proyecto que debe estar escrito en SML/NJ (SML de New Jersey), y que tiene como objetivo principal extraer las fechas que tengan la forma dd/mm/yyyy o dd/mm/yy. El programa debe permitirle al usuario especificar el nombre del archivo de texto de donde se quieren extraer las fechas.

Después de extraer las fechas que siguen con el formato requerido, se deben guardar en una lista de tupla que tenga la forma `int*int*int`, siguiendo el mismo orden: el primer entero corresponde al día, el segundo al mes y el tercero al año. Posteriormente se debe recorrer esta lista, para revisar que todas las fechas que se extrajeron del archivo de texto son válidas. Si se encuentra alguna fecha que no cumple con los requisitos, se debe mostrar un mensaje de error al usuario.

Para las validaciones, se debe tomar en cuenta que las únicas fechas válidas son aquellas que tienen años positivos, los meses deben estar dentro del rango que va de 1 a 12, y para los días se debe validar que sean apropiadas de acuerdo al mes y considerando los años bisiestos (un año bisiesto es el que es divisible por 100 o por 4, pero no entre 100).

Una vez que las fechas estén validadas, se deben ordenar en forma ascendente, y después convertirlas a formato de hileras. Esto quiere decir, por ejemplo, la fecha 12/02/2013 deberá traducirse a "12 de Febrero del 2013". Por último se deben imprimir únicamente las fechas que sean válidas, en formato de hilera.

Diseño del programa

Para poder desarrollar este proyecto, se hizo una lista con aspectos que se debían cumplir:

- Leer archivo.
- Extraer fechas.
- Validar formato.
- Ingresar fechas a tupla.
- Buscar fechas inválidas.
- Ordenar ascendentemente.
- Convertir a formato de hilera.
- Imprimir fechas validas en formato de hilera.

La solución que le dimos fue la siguiente:

- Se empezó mandando un mensaje a consola: “Digite el nombre del archivo a utilizar” para que el usuario ingrese el nombre del archivo del cual desea extraer las fechas
- Para la lectura de archivos, usamos una función ‘readlist’ que leyera el archivo y a la vez lo guardara en una lista:

```
fun readlist (archivo : string) =  
  let val ins = TextIO.openIn archivo  
  fun loop ins = case TextIO.inputLine ins of SOME line =>  
    line :: loop ins | NONE =>  
    [] in loop ins before TextIO.closeIn ins end ;
```

- Después se decidió hacer una función que llamamos ‘leerarch’ la cual leer el archivo char por char. Esta función recibe como argumento una lista de strings y una tupla donde se van a ir insertando las fechas. La misma valida que la lista no este vacía, y va buscando los dígitos char por char. Si encontrara uno, lo manda a la tupla.
- Para validar la fecha se creó la función ‘validar fecha’ que tiene como argumento un elemento x de la tupla. Esta función toma en cuenta las restricciones para el día, el mes y el año.
- También se desarrolló una función para pasar la lista de fechas a lista de días

- Se implementó una función de ordenamiento de quicksort, conteniendo los meses.
- Además se desarrolló una función que obtiene el nombre correspondiente a un numero de mes
- Finalmente para la impresión de resultados, se desarrollaron un par de funciones para validar las fechas y posteriormente imprimirlas en el formato deseado.

Librerías usadas

No se utilizaron librerías para la elaboración de esta tarea.

Análisis de Resultados

- Leer archivo: se logró leer un archivo indicado por el usuario.
- Extraer fechas: se lograron extraer las fechas del archivo de texto previamente señalado.
- Validar formato: se logró hacer la validación de fechas con el formato dd/mm/yyyy o dd/mm/yy
- Ingresar fechas a tupla: las fechas se pudieron ingresar a la tupla
- Buscar fechas invalidas: usando ciertas restricciones, se pudieron descartar las fechas invalidas
- Ordenar ascendentemente: se pudieron ordenar las fechas ascendentemente mediante un algoritmo de transformación a dias
- Convertir a formato de hilera: se lograron convertir las fechas a formato de hilera para su posterior impresión
- Imprimir fechas validas en formato de hilera: se lograron imprimir los resultados esperados, dejando atrás las fechas invalidas y tomando en cuenta que se debían imprimir las fechas en formato de hilera

Manual de usuario

- Instalación

Compilador:

Colocar en la terminal: `sudo apt-get install mlton`

Para ejecutar un archivo desde terminal:

`mlton Prueba.sml ("Prueba" siendo el nombre del archivo)`

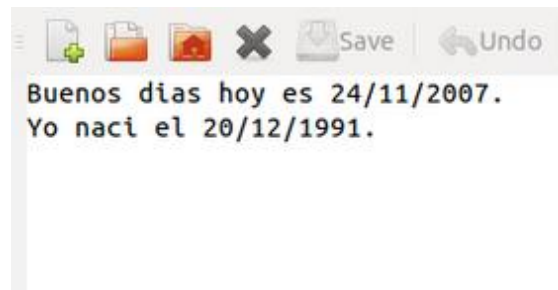
`./Prueba`

- Programa

1. Una vez que se corre el programa, se va a mostrar un mensaje indicando que el usuario debe ingresar el nombre del archivo del cual desea extraer fechas.
2. Cuando se indica esto, el programa empieza la lectura del archivo buscando fechas que sean válidas, de lo contrario le va a indicar al usuario que existen fechas inválidas:
3. Finalmente se imprimirán las fechas encontradas en formato de hilera:

- Ejemplo de corrida

Se tiene el siguiente archivo de texto llamado "lectura.txt":



Para correr el programa, se indica el folder donde se encuentra y se procede a abrirlo utilizando los comandos necesarios:

```
mariam@ubuntu:~$ cd Descargas
mariam@ubuntu:~/Descargas$ mlton tareaprogramada3.sml
mariam@ubuntu:~/Descargas$ ./tareaprogramada3
Digite el nombre del archivo a utilizar
█
```

Enseguida sale la indicación para el usuario, de que digite el archivo que desea utilizar. Una vez que se indica el nombre del archivo, se imprimen las fechas encontradas en el archivo de texto, en forma de hilera y ordenadas de manera ascendente:

```
mariam@ubuntu:~$ cd Documentos
mariam@ubuntu:~/Documentos$ mlton tareaprogramada3.sml
mariam@ubuntu:~/Documentos$ ./tareaprogramada3
Digite el nombre del archivo a utilizar
lectura.txt
20 de Diciembre de 1991
24 de Noviembre de 2007
mariam@ubuntu:~/Documentos$ █
```

Conclusión personal

Al realizar este proyecto, mediante la estructura de este programa desarrollado en SML para leer archivos de texto, se logró practicar la programación funcional, que es un paradigma de programación basado en el uso de funciones aritméticas que no maneja datos de mutables o de estado. Se logró adaptarse al nuevo lenguaje e instruirse sobre los estándares que requiere.

En cuanto al proyecto, se logró lo que se esperaba, se consiguieron todos los objetivos. Para conseguir los objetivos de esta tarea, fue necesario investigar sobre elementos que nos pudieran ayudar a lograr lo que se esperaba. Además se tuvieron que hacer muchas pruebas, para poco a poco eliminar todos los errores, ya que no estábamos muy familiarizados con el lenguaje. Se logró entender el funcionamiento del lenguaje, lo que requiere el entendimiento de la sintaxis y semántica del mismo, para poder desarrollar código que funcione adecuadamente.

Bibliografía

- Páginas de Internet
 - ✓ <http://www.smlnj.org/doc/literature.html#tutorials>
 - ✓ <http://www.cs.cornell.edu/riccardo/prog-smlnj/notes-011001.pdf>
- Presentaciones vistas en clase