

Administración de Tecnologías de Información

Lenguajes de Programación

Tarea Programa No. 3 Lenguaje SML

Integrantes:

Alexander Durán Martínez

Andrew Araya Vega

Benjamín Calvo de León

Tabla de contenido

Descripción del problema.....	3
Diseño del programa.....	3
Decisiones de diseño.....	3
Algoritmos usados.....	3
Librerías Usadas.	4
Análisis de Resultados.....	4
Manual de Usuario.....	4
Conclusión Personal.....	5

Descripción del problema.

Se solicita un programa en lenguaje SML, capaz de leer archivo de texto (.txt), que permita encontrar todas las fechas en el archivo de texto, y que permita validar si pueden darse o sino, por ejemplo, si en un archivo viniera “31 de febrero de 2013”, notificar que no es una fecha válida ya que febrero solo cuenta con 28 días y en el año bisiesto con 29, entonces, el sistema además de validar dichas fechas deberá acomodarlas según el año, de forma cronológica, es decir desde la fecha más antigua hasta la más actual, además deberá realizar una transformación de la fecha en números, a la fecha en palabras, es decir pasar “01/01/2013” a “1ero de enero de 2013”.

Además, al leer las fechas deberán ser guardadas en tuplas, después de ser validadas se imprimen todas las fechas válidas en forma de hileras (mediante una lista).

Diseño del programa.

Decisiones de diseño.

En realidad es complicado decidir una herramienta para realizar el programa, así que se descarga el programa Emacs, desde el sitio web: “<http://www.gnu.org/software/emacs/>”, luego de esto, se procede a instalarlo, el profesor asignó un manual de cómo instalar emacs, luego de esto es indispensable instalar el programa que permitirá el desarrollo sml, para esto se ingresa a “<http://www.smlnj.org/>”.

Ya con éstas herramientas instaladas mediante el manual que nos asignó el profesor, se decide crear un .txt con fechas, éstas son al azar, éstas pueden ser válidas o no. EL programa consiste en tomar dicho texto e ir leyendo línea por línea, guarda cada línea en una tupla que es guardada posteriormente en una lista, hasta terminar de leer el texto, después de esto se obtiene cada tupla y se toma como una nueva lista, se lee y se buscan posibles números, entonces, al encontrar los números con el formato de fecha, se toma, se realiza una validación para asegurarse de que sea correcta, se guarda en otra lista, en ésta otra lista se encuentran todas las fechas validadas, cada una en una nueva posición, se hace el proceso de acomodar, para esto se toman las que tienen el año menor, en caso de que tuvieran un mismo año, se mueve la posición del indicador de la lista al mes, para ver cuál es más antigua, y en caso de que tuvieran el mismo mes, se hace lo mismo a día, después se toman, se transforman a fechas con letras en los meses, para esto, se realiza una comparación, se revisa en el mes y se imprimen de la siguiente forma “día + de + mes en letras + de + año”, después de esto, es importante tomar cada fecha e ir imprimiendo según el orden de la lista, se va eliminando la cabeza de la lista según se vayan imprimiendo en un ciclo recursivo.

Algoritmos usados.

Como ya se mencionó, se usan algoritmos como ciclos recursivos en la búsqueda de fecha en el archivo, para esto se usa un algoritmo que consiste en buscar líneas hasta que no haya más, es decir, que el texto llegue a nulo, otro algoritmo importante es ir leyendo la lista extrayendo todas las tuplas con las líneas leídas, en éste caso lo importante fue tomar las tuplas y convertirlas a listas, esto para ir leyendo posibles números que coincidieran con el formato de fechas, entonces, se guardan para después realizar el ordenamiento, el ordenamiento, utiliza un contador, que selecciona cada fecha, la compara con la siguiente, si es menor queda así, sino, las cambia de posición (en efecto es un ordenamiento burbuja), no es tan eficaz pero es eficiente. Al final se imprime la cabeza de la lista y se retorna a la función de imprimir con la cola de la lista, de esta forma se asegura que las fechas menores se vayan imprimiendo primero.

Librerías Usadas.

No se usó ninguna librería específica, podría tomarse a Emacs como una especie de librería sin embargo lo consideramos más una herramienta para desarrollar el código, todo el código es realizado por métodos propios que no requieren de librería alguna del lenguaje.

Análisis de Resultados.

- Se logra instalar correctamente Emacs y SML.
- Se logra leer el archivo.
- Se logra tomar cada tupla y extraer fechas y validarlas.
- Se logra convertir la fecha de números a palabras (hileras).
- Faltó realizar el ordenamiento de fechas para que aparezcan desde la más antigua hasta la más actual.
- Se propone usar herramientas alternas como mlton que es otra herramienta de compilación en Linux en combinación con Geany, sin embargo se encuentra difícil de usar y se cuenta con poco tiempo para asesorarse sobre su correcto uso.
-

Manual de Usuario.

Para realizar la instalación de Emacs seguimos el manual asignado por el profesor:

Instalación de EMACS

Bajar última versión de EMACS

Linux: `sudo apt-get install emacs`

Mac OS X: <http://emacsformacosx.com/>

Windows: <http://www.gnu.org/software/emacs/>

Instalación de SML/NJ

Linux: Revisar si distribución de Linux tiene un paquete disponible para SML/NJ. En caso de que no tenga, seguir las instrucciones de instalación de <http://www.smlnj.org/dist/working/110.75/>

Windows: Bajar e instalar smlnj.msi <http://www.smlnj.org/dist/working/110.75/>

Mac OS X: Bajar el instalar el dmg de <http://www.smlnj.org/dist/working/110.75/>

Para revisar si SML se instaló bien, abrir la consola, digitar sml y se debería mostrar:

Standard ML of New Jersey ...

Instalación de SML Mode para Emacs

Dentro de Emacs, ejecutar el comando Meta-x (Alt-x), y digitar el comando listpackages, digitar Enter. Las versiones más modernas de Emacs deberían tener el listpackages.

Dentro de los paquetes listados, buscar "sml-mode", darle click, y darle install.

Salir y reiniciar Emacs

Para verificar la instalación del sml-mode, crear un nuevo archivo (C-x C-f), digitar el nombre del archivo (Ejemplo: archivo.sml), y dar Enter. En la parte inferior Emacs, en la barra de estado, debería aparecer SML.

Poner código de SML para verificar que funcione, puede ser algo como:

```
val x = 5;
```

Mientras se esté editando el archivo, digitar C-c C-s, digitar el comando sml, y dar Enter. Si ese comando da error, se tiene que editar el archivo de configuración de Emacs.

En Linux, digitar C-x C-f y luego digitar ~/.emacs para abrir el archivo de configuración, y escribir las siguientes líneas:

```
(setenv "PATH" (concat "/ruta_a_smlnj/smlnj-110.75/bin:" (getenv"PATH")))
```

```
(setq exec-path (cons "/ruta_a_smlnj/smlnj-110.75/bin" execpath))
```

Se debe reemplazar la parte de ruta_a_smlnj por la ruta en la que smlnj fue instalado, eso va a depender de cómo se instaló.

Siguiendo, la instalación de SML, deberá ingresar a la página <http://www.smlnj.org/dist/working/110.75/index.html> donde encontrará el link de descarga, seguidamente a <http://www.smlnj.org/dist/working/110.75/NOTES/INSTALL> que lo guiará en el procedimiento de realizar una instalación correcta.

Para ejecutar el programa es necesario abrir el archivo, click derecho, abrir con Emacs, seguidamente:

- Ctrl-c, ctrl-s
- Use "nombre del archivo".sml; (es necesario el punto y coma ya que es parte de la sintaxis)
- Ejecutar la función principal que se llama extraerFechas ("Escritorio/textoprueba.txt");

Conclusión Personal.

Como conclusión personal consideramos que dicha programada fue en realidad sencilla, sin embargo, lo difícil fue el lenguaje, ya que la herramienta "Emacs", es bastante complicada para usar su debugguedor que permita registrar problema de tipo lógico, en realidad, el mayor problema resulta ser el lenguaje en sí, ya que no hay mucha información que pueda ser válida, muchos programas que se encontraron son inválidos o no dan ningún resultado productivo. Otro problema que afecta fue el tiempo designado a dicha tarea, ya que aprender a usar la herramienta "Emacs" toma muchísimo tiempo.

Para la búsqueda de código buscamos en fuentes como www.stackoverflow.com, y la propia página de smlnj. Sin embargo, consideramos que es muy poca la información brindada y que la asistencia a clases y el uso de las presentaciones vistas en clase son fundamentales para entender el lenguaje y tratar de diseñar algoritmos funcionales que cumplan con las expectativas y realizar el programa de la forma más adecuada.

Nos afectó como grupo la unión, ya que los horarios son muy incompatibles y no siempre se puede trabajar los tres al mismo tiempo, esto conlleva a un proceso de programación más lento que tiene más dificultades y menos cabezas que puedan pensar en una solución alternativa.