 Costa Rica Institute of Technology

Años Bisiestos.

Computer Engineering

San José

Sleyter Angulo Chavarría.

2013388958

Ing. Esteban Arias Méndez.

Introducción

La programación en el lenguaje ensamblador, es una de las mas tediosas, ya que debes aprender a realizar la lógica de una manera muy distinta. Generalmente debemos dejar de lado el alto nivel de otros lenguajes como JAVA, Python, entre otros; para adentrarnos en el mundo de “Assembly Language”.

En ensamblador la lógica recae principalmente en instrucciones maquina, sin embargo; el nivel de ensamblador esta por encima del mismo lenguaje de maquinas. Una vez que se obtiene la lógica o practica del lenguaje, es mas fácil el diseño de las diferentes aplicaciones a realizar.

Para este caso, la aplicación que se desea desarrollar consiste en que el usuario digite una fecha y como resultado obtendremos, el día correspondiente a esa fecha y si el año que se digito es bisiesto o no.

Conforme se vaya avanzando en la solución de dicho problema, se ira explicando la forma como se implementaron las distintas funciones para obtener dichos resultados. Además se analizará los pro y contras de utilizar ensamblador como lenguaje principal.

Especificación del Problema

Según la especificación dado por el profesor tenemos:

“Como se comento en clase, esta tarea en ensamblador consiste en realizar un programa que reciba una hilera de una fecha por parte del usuario, en formato de dígitos para los datos: día, mes y año. Para los datos recibidos su programa deberá calcular a qué día de la semana corresponde dicha fecha: Lunes, Martes, Miércoles, ..., Domingo. Además el programa debe indicar si el año de la fecha indicada es un año bisiesto o no.

La fecha ingresada debe ser un solo string y no varios datos leídos por separado. Por ejemplo debe ingresarse desde el teclado: "12 Octubre 1502", "20 05 3000", "7 7 777". El orden esperado es día, mes y año. Nótese que hay fechas que corresponden a años en el pasado como a años en el futuro. El rango de fechas aceptado será a partir del año 0 y posteriores, denominados DC. El aceptar años antes del año 0, es decir AC, será valorado como puntos extra, por ejemplo "1 3 120000 AC" que corresponde a una fecha de hace 118 mil años.

La salida del programa indicando el día debe ser un banner en grande con el dato del día y luego un mensaje indicando si el año es bisiesto o no.

El banner debe lucir como :

L    U   U  N   N  EEEE    SSSS

L    U   U  NN  N  E      S

L    U   U  N N N  EEE     SSS

L    U   U  N   N  E          S

LLLL  UUU   N   N  EEEE   SSSS

El diseño de cada letra puede cambiar, pero cada letra debe ser formada por las letras en mayúscula de la misma letra, como en el ejemplo.

Además en el programa, debe almacenarse una única copia de cada letra, es decir, por ejemplo la letra E es usada en varios días: lunes, miércoles, viernes, etc.. pero la letra E debe ser almacenada en el programa una única vez. Esto para hacer más pequeño y eficiente el programa.”

Descripción

Una vez entendido el problema, debemos empezar las posibles soluciones que nos van a conducir a la solución definitiva y final. Básicamente la implementación de las funciones utilizadas fue la siguiente:

* Section .data:

En esta sección declaro todas las variables que voy a utilizar y tienen un valor fijo (constante).

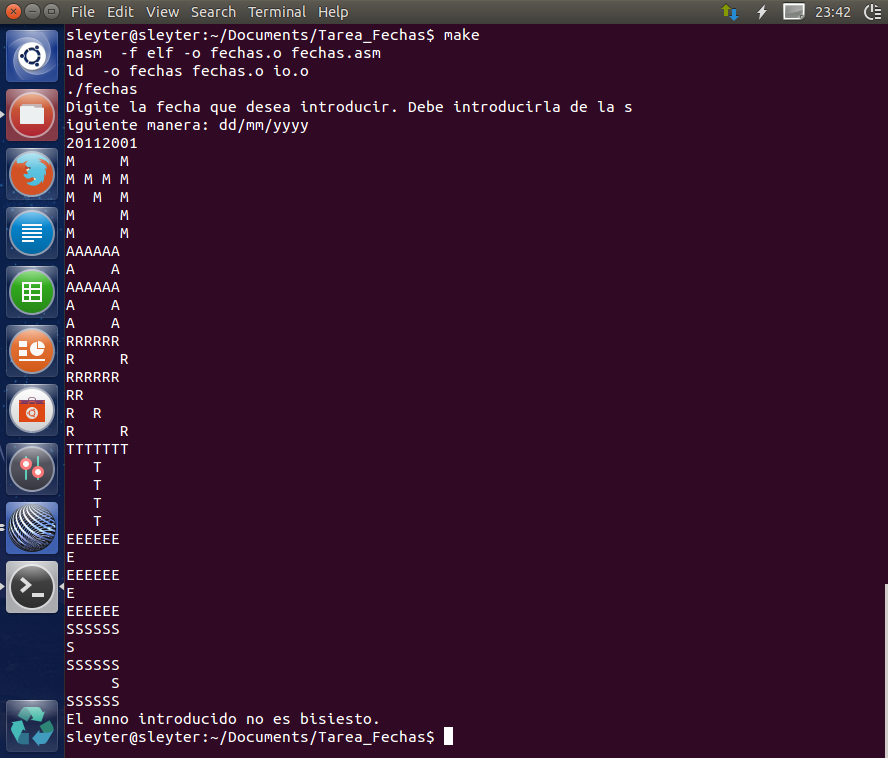
* Section .bss:

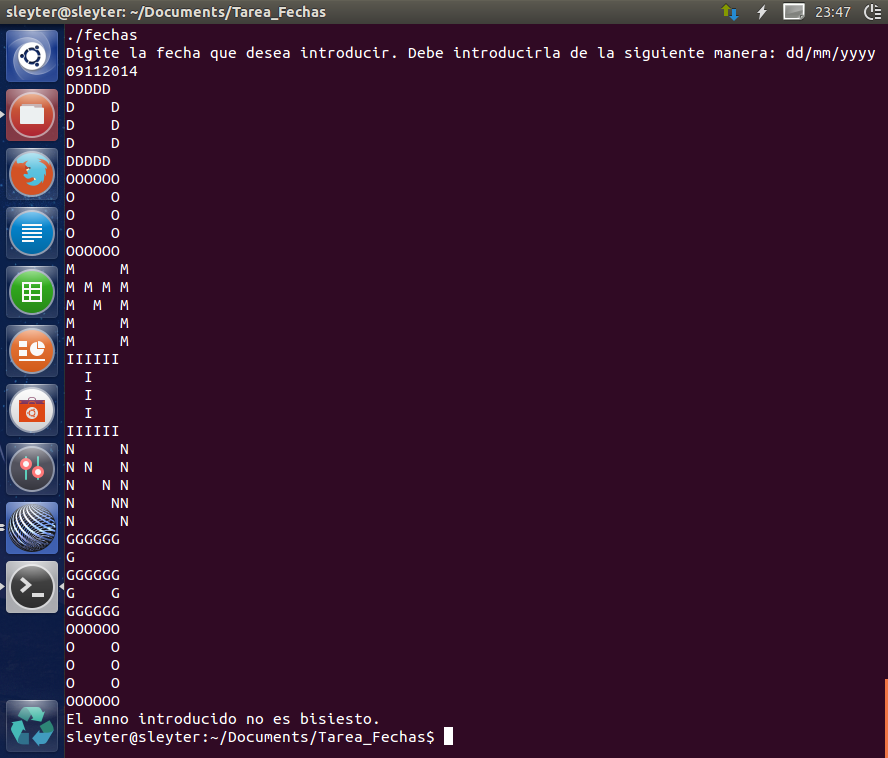
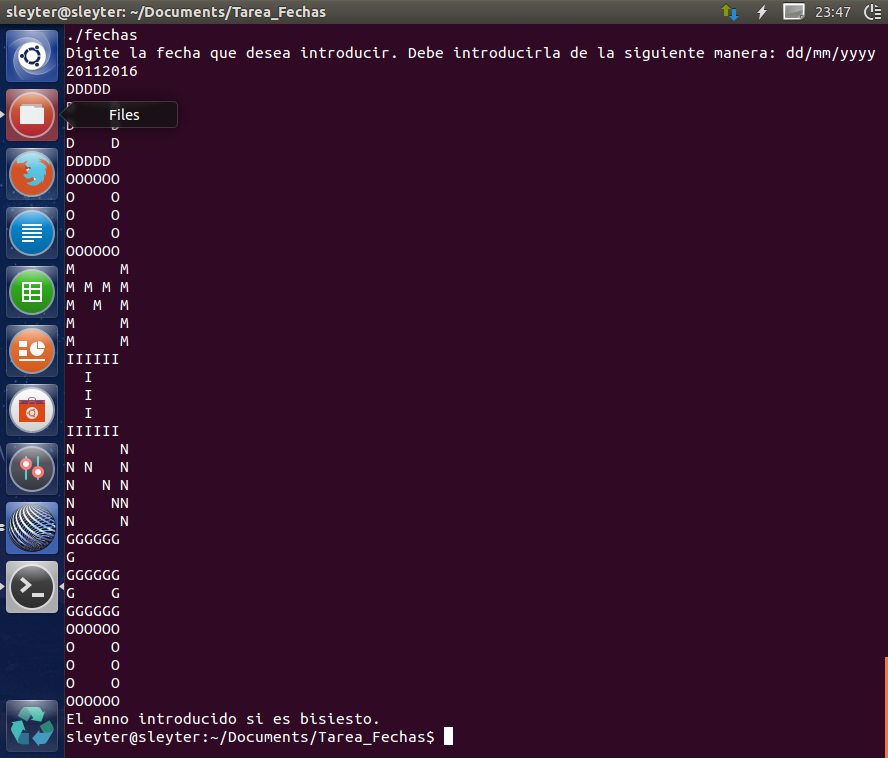
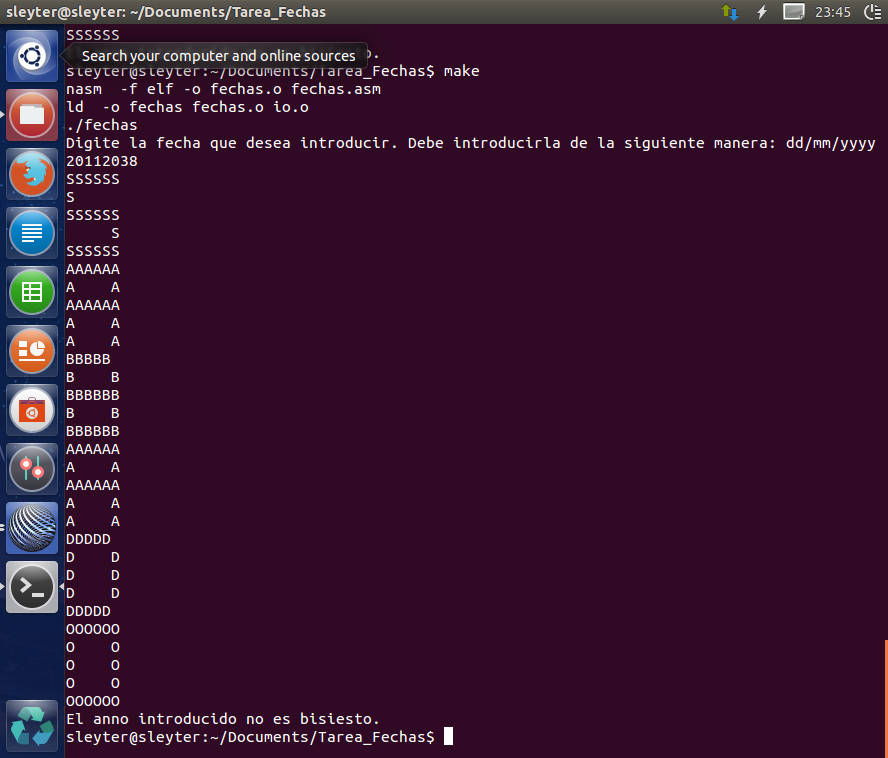
En esta sección declaro las variables que utilizo, pero que aun no tienen un valor, y este debe ser dado por el usuario. Por ejemplo: la fecha que va a ser dada por el usuario.

* Section .text:

En esta sección corresponde a la parte donde se va a escribir el código y contiene todas las funciones, dentro de ellas tenemos:

* \_start: es la función mas importante ya que sin ella ensamblador no es capaz de iniciar y lanzaría un error.
* Imprimir\_Bienvenida: es una función, la cual le dará al usuario la bienvenida al sistema.
* Sacar\_Dia: esta función tomara lo dos primeros dígitos, que son obtenidos a través de la fecha que digita el usuario, y son almacenados en la variable “dia”.
* ATOI\_Dia: lo que realiza esta función, es que toma los caracteres ASCII de la variable “dia” y los pasa a un INTEGER.
* Sacar\_Mes: toma los dígitos 2 y 3 de la fecha introducida por el usuario y los guarda en la variable “mes”.
* ATOI\_Mes: toma los caracteres ASCII de la variable “mes” y los convierte en INTEGER.
* Sacar\_Anno: toma los dígitos 4 hasta 7 de la fecha introducida por el usuario y los guarda en la variable “anno”.
* ATOI\_Anno: convierte de ASCII a INTEGER la variable “anno”.
* Determinar\_Mes: esta función determino el mes que el usuario digito como fecha y lo comparo para obtener el mes. Principalmente para el caso especial de Febrero.
* Obtener\_Dia: es una de las funciones mas importantes, ya que con esta se obtiene el dia de la semana siguiendo la siguiente formula N = D + M + A + E [A/4] + S.
* Imprimir\_Dia: esta función lo que realiza, es que toma la variable “cont\_dia” y compara con los distintos valores asignados a los días y si corresponde a ese día, llama a la función que imprime ese día.
* Imprimir\_“Cualquier día”: llama a las distintas letras que se van a utilizar para el dia e imprime una por una.
* Es\_Bisiesto: para saber si un año debe ser divido por 4, 100 y 400, y si cumple las anteriores condiciones, ese año corresponde a uno bisiesto.
* Imprimir\_EsBisiesto: es una función que imprime un buffer que contiene un mensaje indicando que ese año es bisiesto.
* Imprimir\_NoesBisiesto: es una función que imprime un buffer que contiene un mensaje indicando que ese año no es bisiesto.





Conclusiones

El programar en ensamblador, requiere de una habilidad especial a la hora de diseñar la solución al problema. El principal reto, como en otros lenguajes es evitar los fallos.

En ensamblador el error que mas cuesta en cuando no lanza ningún error, pero el resultado que se obtuvo no es el correcto. Eso sucede muy frecuentemente en ensamblador. Pero no solo aquí sino también en otros lenguajes. El error mas común, que obtuve durante la elaboración de la tarea fue “segmentation fault”.

Ese error consiste o radica, en que se accedió a una parte de la memoria que estaba ocupada o que sobrepaso. Cuesta mucho saber donde ocurre eso ya que no sabes que parte esta provocando dicho error.

Lo mas importante al final es la tarea y el conocimiento aprendido, y que sin el no avanzamos como profesionales. La idea en cada curso es ir fomenta el aprendizaje de los contenidos que brindan los profesores como la experiencia que ellos tienen.

Bibliografía

Método para determinar si un año es un año bisiesto, Microsoft, recuperado 08 de noviembre de 2014, http://support.microsoft.com/kb/214019/es

Como calcular el dia que cae una fecha?, Taringa, recuperado 06 de noviembre de 2014, http://www.taringa.net/post/info/928508/Como-calcular-el-dia-que-cae-una-fecha.html