Organización de computadoras

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Segundo Cuatrimestre de 2017

Proyecto N° 2
Programación en Lenguaje Ensamblador

Propósito

El objetivo principal del proyecto es implementar en lenguaje ensamblador, un programa que enumere cada una de las línea de un archivo de texto parametrizado.

En función a la invocación, la numeración línea a línea será visualizada por consola, o bien, será almacenada en un nuevo archivo también parametrizado. En ambos casos, el programa agregará una última línea con la cantidad total de líneas numeradas.

La implementación debe realizarse utilizando el ensamblador *Yasm*, sobre una arquitectura *Intel x86*, haciendo uso de las llamadas al sistema provistas por el sistema operativo *GNU/Linux*.

Sintaxis

El programa implementado debe permitir ser invocado desde la línea de comandos mediante la siguiente sintaxis:

```
enum [ -h ] | archivo_entrada [ archivo_salida ]
```

Las opciones separadas por una barra vertical denotan *posibles alternativas* a llamadas por línea de comandos, y los parámetros entre corchetes denotan *parámetros opcionales*. El significado de las diferentes opciones de invocación es el siguiente:

- Si se hace uso de la primer alternativa especificando únicamente el parámetro `-h`, el programa debe ofrecer una pequeña ayuda por pantalla la cual debe reflejar un breve resumen del propósito general del programa junto con la especificación de las diferentes posibilidades que ofrece para ser invocado.
- Si se hace uso de la segunda alternativa especificando únicamente el parámetro `archivo_entrada`, el programa debe leer las líneas (línea por línea) contenidas en el `archivo_entrada`, y mostrar cada una de estas líneas de forma numerada en la consola.
- Si se hace uso de la segunda alternativa especificando tanto el nombre del `archivo_entrada` como el nombre del `archivo_salida`, el programa debe leer las líneas (línea por línea) contenidas en el `archivo_entrada`, y escribir cada una de estas líneas de forma numerada en el `archivo_salida`.

Se considera que una línea está numerada, cuando antes de mostrar/escribir la misma, se agrega el número de línea seguido de ": "(dos puntos y un espacio). Por ejemplo, si el `archivo_entrada` mantiene el contenido indicado en la Figura 1, ya sea en la consola o en el `archivo_salida`, se deberá mostrar/escribir el contenido indicado en la Figura 2.

```
#include <stdio.h>
int main(){
  int i = 0;
  printf("%i\n", i);
  return 0;
}
```

Figura I

```
1: #include <stdio.h>
2: int main(){
3:   int i = 0;
4:   printf("%i\n", i);
5:   return 0;
6:  }

Cantidad de líneas: 6.
```

Figura II

Toda vez que el programa termine su ejecución, se debe informar sobre la situación de terminación a quien haya invocado al mismo. Para esto se debe utilizar la llamada al sistema `sys_exit`, respetando la siguiente convención:

EBX	**Detalle**
0	Terminación normal.
1	Terminación anormal por error en el archivo de entrada.
2	Terminación anormal por error en el archivo de salida.
3	Terminación anormal por otras causas.

Sobre la implementación

- El archivo fuente principal se debe denominar **enum.asm**.
- La copia o plagio del proyecto es una falta grave. Quien incurra en estos actos de deshonestidad académica, desaprobará automáticamente el proyecto.

Sobre el estilo de programación

- El código implementado debe reflejar la aplicación de las técnicas de programación modular estudiadas a lo largo de la carrera.
- En el código, entre eficiencia y claridad, se debe optar por la claridad. Toda decisión en este sentido debe constar en la documentación que acompaña al programa implementado.
- El código debe estar indentado, fuertemente comentado, y debe reflejar el uso adecuado de nombres significativos para la definición de variables, funciones y parámetros.

Sobre la documentación

Los proyectos que no incluyan documentación estarán automáticamente desaprobados. La misma debe:

- Estar dirigida a desarrolladores.
- Explicar detalladamente los programas realizados, incluyendo el diseño de la aplicación y el modelo de datos utilizado, así como toda decisión de diseño tomada y toda observación que se considere pertinente.

- Incluir explicación de **todas** las funciones, rutinas o algoritmos implementados, indicando su prototipo y el uso de los parámetros de entrada y de salida (tanto sean registros, valores almacenados en la pila, etc). Se espera que la explicación esté dada en términos de diagramas, pseudocódigos, o cualquier representación que considere adecuada, la cual asista al desarrollador a comprender cada una de las líneas del código implementado.

Sobre la entrega

Toda comisión que no cumpla con los requerimientos, estará automáticamente desaprobada. Los mismos son:

- Las comisiones estarán conformadas por 2 alumnos, y serán las que oportunamente registró y notificó la cátedra.
- La entrega del código fuente y la documentación se realizará a través de un archivo comprimido **zip** o **rar**, denominado ***PR2-Apellido1-Apellido2***, que debe incluir las siguientes carpetas:
  - **Fuentes**, donde se debe incorporar el archivo fuente "enum.asm" (ningún otro).
  - **Documentación**, donde se debe incorporar el informe del proyecto en formato PDF (ningún otro).
- El archivo comprimido debe enviarse por e-mail, respetando el siguiente formato:
  - **Para:** *federico.joaquin@cs.uns.edu.ar*
  - **Asunto:** *OC :: PR2 :: COM XX :: Apellido1 - Apellido2*
  - **Cuerpo del e-mail:**
    Se adjunta Proyecto Nº 2, de la comisión XX:
    Apellido, Nombre 1 - LU 1
    Apellido, Nombre 2 - LU 2
- El e-mail debe ser enviado con anterioridad al día **Jueves 23 de Noviembre de 2017, 22:00 hs**. Se considerará como hora de ingreso, la registrada en el servidor de e-mail del DCIC.

Sobre la corrección

- La cátedra evaluará tanto el **diseño** e **implementación** como la **documentación** y **presentación** del proyecto, y el cumpliemiento de **todas** las condiciones de entrega.
- Tanto para compilar el proyecto, como para verificar su funcionamiento, se utilizará la máquina virtual "OCUNS" publicada en el sitio web de la cátedra.