Organización de Computadoras
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Segundo Cuatrimestre de 2015

Proyecto Nº 2
Programación en Lenguaje Ensamblador

Enunciado

Realizar un programa que reciba como argumento un archivo de enteros positivos y un criterio de selección, y escriba en un nuevo archivo, o muestre por pantalla, los enteros del archivo que cumplen con dicho criterio. El programa debe ser implementado en lenguaje ensamblador para la arquitectura i386 haciendo uso de las llamadas al sistema provistas por el sistema operativo GNU/Linux.

Sintaxis

El programa implementado (al que llamaremos `seleccionar.asm`) debe conformar la siguiente especificación al ser invocado desde la línea de comandos:

$ seleccionar [-h] <criterio><archivo entrada>[<archivo salida>]

Los parámetros entre corchetes denotan parámetros opciones. El significado de las diversas opciones es el siguiente:

- En caso de no especificar parámetros o especificar el parámetro -h como primer argumento en la línea de comandos, el programa debe mostrar una pequeña ayuda por pantalla. Esta ayuda consiste en un breve resumen del propósito del programa junto con una reseña de las diversas opciones disponibles.
- Si se especifica un archivo de entrada y no uno de salida, el programa deberá encontrar todos los número del archivo de entrada que satisfacen el *criterio* ingresado y mostrarlos por pantalla.
- Si se especifica un archivo de entrada y uno de salida, el programa deberá encontrar todos los número del archivo de entrada que satisfacen el *criterio* ingresado y escribirlos en el archivo de salida.
- El criterio especificado debe ser uno de los siguientes:
  - `>N` Selecciona los números mayores a `N`.
  - `<N` Selecciona los números menores a `N`.
  - `%N` Selecciona los números que son múltiplo de `N`.
- El archivo de entrada debe ser una secuencia de enteros, uno por cada línea del archivo.

Notificación de errores

Toda vez que el programa termine su ejecución, se debe informar sobre la situación de terminación a quien haya invocado al programa. Para esto se debe hacer uso de 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.

Ejemplo de invocación

Si el archivo *números.txt* contiene los números:

1↲
5↲
6↲
3↲
25↲
63↲
12↲
11↲
21

en la siguiente tabla se muestra qué debería mostrar por pantalla cada una de la siguientes invocaciones:

Invocación	Resultado por pantalla
$ seleccionar %3 numeros.txt	6 3 63 12 21
$ seleccionar ">20" numeros.txt	25 63 21
$ seleccionar "<20" numeros.txt	1 5 6 3 12 11

Sobre la implementación

- Se debe implementar en lenguaje ensamblador para la arquitectura i386, haciendo uso de las llamadas al sistema provistas por el sistema operativo GNU/Linux.
- El programa debe funcionar en la distribución GNU/Linux provista por la cátedra.
- El archivo fuente principal se debe denominar `seleccionar.asm`.
- Se puede asumir que la entrada contiene solamente números positivos.

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 el informe que acompaña al programa implementado.
- Es una buena oportunidad para seguir fomentando el hábito de indentar adecuadamente el código fuente, adoptando un único estilo a lo largo de todo el programa.
- Se recomienda documentar el código implementado, indicando el propósito de cada sección del programa, así como el rol que desempeñan los distintos registros en cada bloque de código.

Sobre la documentación

- La documentación tiene que estar dirigida principalmente a un usuario con conocimientos de programación. Es importante que explique brevemente los programas realizados, así como las decisiones de diseño tomadas, y toda otra observación que se considere pertinente.
- El informe debe incluir una sección en la cual se explique claramente la estructura general del programa, así como la estrategia adoptada para resolver el problema planteado por el enunciado del proyecto.

- Las subrutinas implementadas deben ser explicadas enunciando el nombre de las mismas y los parámetros de entrada/salida utilizados (ya sean registros o direcciones de memoria), para cada uno de los cuales deberá darse una descripción breve de su propósito. *Las funciones más complejas deben incluir una descripción verbal de la estrategia utilizada para su resolución.* De ser necesario, se puede agregar algún bloque de pseudocódigo.
- Se puede incorporar en forma libre todos los ejemplos y gráficos que se consideren adecuados para ayudar al entendimiento de la implementación del proyecto.

Sobre la entrega

- Las comisiones deben estar conformadas de la forma previamente registrada con la cátedra para el proyecto anterior. Solamente se aceptarán cambios en aquellas comisiones que hayan perdido algún integrante.
- El código fuente del proyecto deberá ser enviado en un archivo *zip* por mail a la asistente de la cátedra hasta las **8:00hs** del día **Lunes 30 de noviembre de 2015**. *Toda comisión que no cumpla este punto estará automáticamente desaprobada.*

  Tanto el asunto del mail como el nombre del archivo comprimido debe ser el siguiente:

  [OC2015] Proyecto 2 - Comisión *número de comisión - apellidos de los integrantes de la comisión*

- El informe del proyecto deberá ser entregado el día **Lunes 30 de noviembre de 2015**, de **10 a 12hs** en el aula de clase. *Toda comisión que no cumpla este punto estará automáticamente desaprobada.*
- Deberá presentarse un folio plástico (no se aceptarán carpetas), cerrado con cinta adhesiva, conteniendo los siguientes elementos:
  - Una carátula que identifique claramente a los integrantes de la comisión e incluya el número de comisión.
  - El código fuente impreso en doble faz.
  - La documentación del proyecto impresa en doble faz.

  No se aceptarán discrepancias entre el código fuente impreso y el enviado por mail.