CC4301 Arquitectura de Computadores – Tarea 4 – Otoño 2020 – Profesor: Luis Mateu

La función *sort_x86* está programada en assembler x86 en el archivo *sort-x86.s*. Esta función ordena alfabéticamente un arreglo de strings usando un algoritmo ridículamente ineficiente. El código equivalente en C está comentado, mostrando la ubicación de las variables en los registros. La función recorre el arreglo revisando que los elementos consecutivos estén ordenados. Si encuentra 2 elementos consecutivos desordenados, los intercambia y reinicia el recorrido del arreglo desde el comienzo. El arreglo está ordenado cuando se recorre completamente sin encontrar elementos consecutivos desordenados.

El encabezado de la función es: *void sort_x86(char *noms[ ], int n);*

En el arreglo de strings *noms* vienen *n* nombres de personas como *"juan perez"*. Siempre viene un solo nombre y luego un solo apellido, separados por un solo espacio en blanco. El string no contiene otros espacios en blanco.

El archivo *sort-x86-**apnom.s*** es una copia de *sort-x86.s*. Modifique la función *sort_x86* en ***sort-x86-apnom.s*** de modo que se ordene el arreglo primero por apellido, y si los apellidos son iguales, entonces por nombre. El programa de prueba invoca *sort_x86* y muestra en pantalla el resultado del ordenamiento. La salida de su solución debe ser:

```
maria fernandez
monica fernandez
vero fernandez
ana gonzalez
diego gonzalez
pedro gonzalez
tatiana jerez
alberto perez
jose perez
juan perez
```

Instrucciones

Baje *t4.zip* de U-cursos y descomprímalo. Contiene el *Makefile* y los programas que necesita para hacer esta tarea. Compile y ejecute con:

```
$ make test-sort-x86
$ ./test-sort-x86
```

Se mostrará el contenido del arreglo ordenado ***incorrectamente*** primero por nombre y luego por apellido. Reprograme en assembler la función *sort_x86* en el archivo ***sort-x86-apnom.s***, de manera que ordene correctamente primero por apellido y luego por nombre. Compile y ejecute con:

```
$ make test-sort-x86-apnom
$ ./test-sort-x86-apnom
```

Restricciones

- Ud. solo puede modificar en *sort-x86-apnom.s* el código que compara los elementos consecutivos. Este código está delimitado por un par de comentarios en el archivo. No modifique nada más. Sin esta restricción la tarea sería trivial.
- Para comparar haga un ciclo buscando el espacio en blanco (en ascii es el entero 32) en cado uno de los nombres. Invoque *strcmp* para comparar los apellidos. El valor retornado está en *%eax*. Si es distinto de 0 salte a la etiqueta *.decision*. Si no, vuelva a comparar los nombres completos con *strcmp*, como está en el código original.

Ayuda

- El archivo *sort-c.c* es la versión en C de *sort_x86*. Compile y ejecute esta versión con:

```
$ make test-sort-c
$ ./test-sort-c
```

- Programe primero una versión en C de lo pedido en el archivo *test-sort-c.c*. Revise que funcione correctamente.
- Puede obtener ideas de las instrucciones en assembler x86 que debe usar generando el archivo *sort-c.s* con:

```
$ make sort-c.s
```

- Use *ddd* para *entender* y depurar su tarea. Seleccione el menú *View → Machine code window* para ver el assembler. Coloque breakpoints en lugares estratégicos con: *break .while_begin*. Lea la guía rápida para usar gdb en:

  http://www.dcc.uchile.cl/~lmateu/CC4301

- Debe compilar la tarea en 32 bits. Para ello debe instalar las bibliotecas de 32 bits con *sudo apt-get install gcc-multilib* en distribuciones derivadas de *Debian*, como por ejemplo *Ubuntu*.

Entrega

Entregue por medio de U-cursos el archivo ***sort-x86-apnom.s*** con su solución. Su tarea debe funcionar correctamente, si no su nota será 1.0. Se descontará medio punto por día de atraso (excluyendo sábados, domingos, festivos o vacaciones).