

CC4301 Arquitectura de Computadores. Semestre Primavera 2018. Minitarea 6

Profesor: Pablo Guerrero

Entrega: Lunes, 26 de noviembre de 2018, 23:59 hrs.

En esta minitarea desarrollaremos un compilador para una versión muy simplificada de la máquina abstracta SECD¹. Dicha máquina utiliza el stack para encolar instrucciones y el resultado parcial de su ejecución. **Su programa debe recibir una lista de instrucciones SECD y generar un código en Assembler ARM** que ejecute las instrucciones secuencialmente (en una máquina ARM). Las instrucciones SECD disponibles en la máquina que implementaremos son:

- (INT_CONST n): Apila el entero n.
- (ADD): Toma los dos elementos en el tope de la pila y los suma, reemplazando en la pila dichos elementos por el resultado de la suma.
- (SUB): Toma los dos elementos en el tope de la pila y los resta, reemplazando en la pila dichos elementos por el resultado de la resta.

Un ejemplo de una lista de instrucciones es el siguiente:

((INT_CONST 3) (INT_CONST 8) (INT_CONST 6) (ADD) (SUB))

La ejecución de dicha lista debería producir en el stack los siguientes estados, paso a paso:

Paso 1: (INT_CONST 3)	Paso 2: (INT_CONST 8)	Paso 3: (INT_CONST 6)	Paso 4: (ADD)	Paso 5: (SUB)
3	8 3	6 8 3	14 3	11

Para generar el stack necesario para la ejecución de la lista de instrucciones SECD, el programa compilado debe utilizar el manejo de stack provisto por ARM y respetar la convención del llamador con stack descendiente.

¹Más información en: La página de SECD en Wikipedia

Al finalizar de ejecutar el código de las instrucciones compiladas, debería quedar en la pila un único entero con el resultado de la ejecución. El compilador debe agregar al final del código Assembler un trozo que desapile este resultado y lo entregue, ya sea en un archivo o en alguna salida visible.

P1.- Implemente el compilador previamente descrito. Para ello, puede usar Scheme, C ó C++. Su programa debe leer la lista desde un archivo, recorrerla, y por cada instrucción SECD, agregar un trozo de código Assembler ARM en el programa de salida, el cual debe ser guardado en un archivo.

Si utiliza Scheme, puede utilizar la función `(display-to-file)` para guardar en un archivo. El siguiente es un ejemplo de su uso:

```
(display-to-file "Hola Mundo!"  
                "file.txt"  
                #:mode 'text  
                #:exists 'replace)
```

Si utiliza C ó C++, use la librería `sexpr2` para leer la lista de instrucciones a una estructura fácil de recorrer. Se adjunta un programa de ejemplo en C++ que usa dicha librería para recorrer una lista e imprimirla en pantalla como árbol (usando tabulaciones). Para compilar dicho programa puede usar la siguiente instrucción en linux:

```
g++ main.cpp sexprPath/libsexp.a -o readList
```

Utilice la misma máquina ARM que en la minitarea 5 (sea ésta real o emulada) para probar el código obtenido desde su compilador.

P2.- Elabore algunas secuencias de prueba de instrucciones SECD de manera de testear el funcionamiento de su compilador. Ejecute dichas secuencias y cheque que el resultado sea el esperado.

Entregue en u-cursos:

- Un documento en pdf que contenga:
 - Una explicación de la solución implementada.
 - Una descripción de las pruebas diseñadas, lo que pretende probar cada una de ellas, y un análisis de los resultados obtenidos.
- Los archivos de código fuente, en el lenguaje utilizado, con comentarios para poder entender clara y detalladamente su funcionamiento.

Descuentos: Una décima por hora, un punto por día de atraso.

²Descargar en [sourceforge](https://sourceforge.net/projects/sexpr/). Más información en: La página web del proyecto `sexpr`