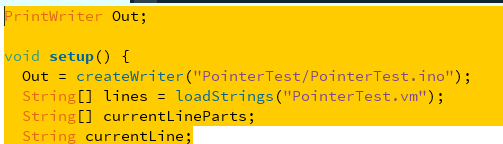
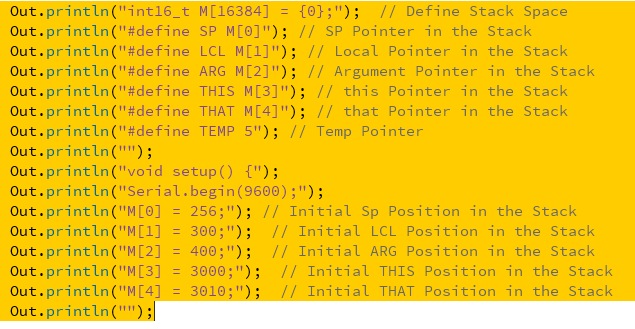
Documento De Funcionamiento Para PointerTest

Este trabajo consiste en leer un archivo .VM y hacer una traducción por medio de processing para que el archivo pueda ser leído en la plataforma de Arduino.

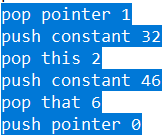
Para esto empezamos a utilizar algunas funciones específicas del lenguaje de processing( Java) para facilitar el proceso de lectura y traducción del archivo



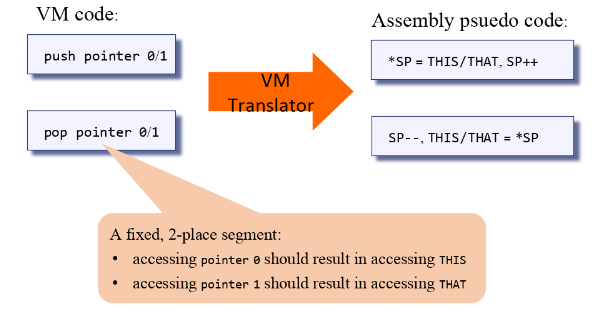
Primero creamos una variable PrintWriter que nos permitirá escribir dentro de un archivo de salida, Luego en el Setup del programa definimos el nombre del archivo de salida y la dirección, en este caso el archivo de salida llevará extensión. ino y se almacenará en dentro de la carpeta “PointerTest” con el nombre de PointerTest.ino



Antes de iniciar la traducción nos aseguramos de declarar las variables que necesitaremos en Arduino para el correcto funcionamiento del programa, en este caso necesitamos definir el tamaño del Stack, el Puntero de Sp y de cada uno de los diferentes segmentos de la memoria en el Stack, además de esto debemos también definir la posición inicial de cada uno de estos segmentos a excepción del Temp, también inicializaremos el serial de Arduino para poder visualizar los resultados del vector de prueba y compararlos con los del archivo.vm.

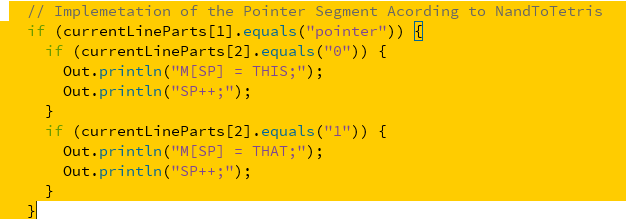


Para este ejemplo debemos implementar el pointer, haciéndole push y pop, de modo tal que debemos de repetir los procesos del archivo anterior añadiendo las siguientes líneas de código.



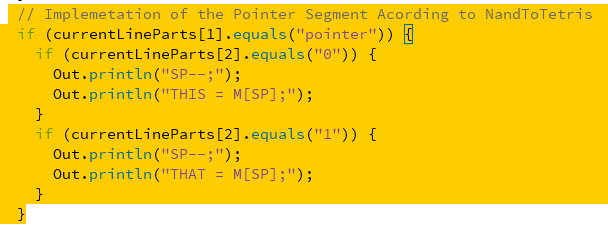
Las siguientes implementaciones están basadas en la implementación que se nos da en la documentación de NandToTetris

**Para el Push Pointer:**

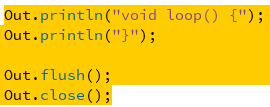
****

Lo que hacemos es verificar que en la segunda posición de la línea del archivo .vm exista un pointer, y si lo hay pasamos a buscar el valor de la tercera línea. Como podemos ver si es un 0 haremos un Push en el segmento This y si hay un 1 haremos un Push en el segmento That.

**Para el Pop Pointer:**

****

Lo que hacemos es verificar que en la segunda posición de la línea del archivo .vm exista un pointer, y si lo hay pasamos a buscar el valor de la tercera línea. Como podemos ver si es un 0 haremos un Pop en el segmento This y si hay un 1 haremos un Pop en el segmento That.



Finalmente abriremos y cerraremos la función loop para un correcto funcionamiento dentro del IDE de Arduino, y utilizaremos la función Flush para asegurarnos que no quede ningún dato en el buffer y la función close para cerrar el archivo una vez todo el proceso halla finalizado.