Ejercicios sesión de laboratorio PLG sobre WebAssebly

Albert Rubio

Ejercicio 1: *Primer ejemplo con WebAssembly.* Descargad todos estos archivos de la sesión del Campus Virtual. Entre los archivos están los ejecutables para pasar del formato textual wat al binario wasm (wat2wasm) del *WebAssembly Binary Toolkit* (wabt) para linux y windows.

Podéis obtener todos los binarios de la wabt para linux (Ubuntu), MacOS o windows en: `https://github.com/webassembly/wabt/releases`

También podéis conseguir los fuentes e instalar la última versión en: `https://github.com/webassembly/wabt`

Para hacer la primera prueba seguid los siguientes pasos:

1. Examinad el código wat que tenéis en main.wat
2. Mirad como se carga código wasm en JavaScript en main.js
3. Pasad el wat a wasm con:

   ```
   $ wat2wasm main.wat
   ```

   desde linux, o bien

   ```
   $ wat2wasm.exe main.wat
   ```

   desde windows.
4. Ahora podéis ejecutar el código JavaScript con `node`:

   ```
   $ node main.js
   ```

   En linux podéis redirigir la entrada.
5. Haced lo mismo con `main-ext.wat`

Ejercicio 2: *Memoria.* Haced las siguientes funciones wat:

1. Una función que dado un número n y una posición p, lee n enteros y los coloca en las n posiciones siguientes p.
2. Una función que dado un número n y una posición p, escribe los n enteros de las posiciones siguientes p.

3. Una función que dado un número n y una posición p, suma el contenido de las n posiciones desde p.

4. Una función que dado un número n y una posición p, incrementa en uno el contenido de las n posiciones desde p.

5. Una función que dado un número n (con $n \geq 1$ y una posición p, obtiene el entero más grande y el más pequeño de las n posiciones desde p.

6. Haced funciones de inicio para combinar la primera función con las otras y poder ver su funcionamiento.

Adaptad `main.js` para ejecutar el nuevo código wat.