

=====

## PRÁCTICA 5

### Generación de Código Intermedio

Los ficheros a los que se hace referencia en este enunciado los encontrarás en `merlin:`  
`/users2/COMPII/salidas/p5`

=====

## I. OBJETIVOS

Implementar los esquemas de generación de código para evaluación completa de expresiones y sentencias básicas de control y acciones de un lenguaje tipo PASCAL a una máquina abstracta de pila.

## II. CONTENIDO

En esta práctica deberás implementar un generador de código para la máquina P, que traduzca de Pascual expresiones, asignaciones, lectura y escritura, selección y mientras que, e invocación de acciones con parámetros por valor y referencia, así como cualquier extra que hayáis implementado en la Práctica 2-3.

## III. SE PROVEE

- Los ficheros `genvec.c` y `genast.c`, con funciones para la generación de código (a vector y por atributos en el árbol de sintaxis, respectivamente) para la máquina P.
- También está disponible una versión actualizada de la librería de manejo de la tabla de símbolos: `tabla.h` y `tabla.c`

## IV. SE PIDE

Como resultado de esta práctica debes completar la gramática de la práctica 2/3 implementado una de las dos formas de generación de código. Se debe someter como resultado un fichero `practica5.tar` que al descomprimirse cree el directorio `practica5/` y que incluya TODOS los ficheros necesarios para generar el compilador (incluyendo '`pascual.y`') además de la pertinente batería de pruebas (dejamos el número de programas a vuestra elección) que demuestre el correcto funcionamiento del programa. El formato de compilación debe ser:

```
$ make pascual
```

El formato de ejecución del compilador debe ser:

```
$ pascual fuente
```

donde `fuentes` es el nombre del fichero SIN EXTENSION (el compilador debe buscar un fichero con la extensión '`.pc`') que contiene el programa fuente Pascual a compilar. El compilador generará el fichero con el mismo nombre del fuente, pero con extensión '`.cp`', que debe contener el código ensamblador correspondiente.

Deberá someterse el fichero (`someter COMPII practica5.tar`) **no más tarde del 3 de Junio de 2009.**

## V. EXTRAS

Genera un fichero con extensión '.xml' que contenga la descripción XML de la implementación del programa fuente (en **genast.c** tienes disponibles funciones que escriben el código p en formato XML a un fichero de texto). Para el ejemplo 1 (transparencias de clase), el código XML sería:

<pre>&lt;max&gt; &lt;programa /&gt; &lt;p&gt;ENP L0&lt;/p&gt; &lt;i&gt; &lt;variable /&gt; &lt;entero /&gt; &lt;dir&gt;3&lt;/dir&gt; &lt;/i&gt; &lt;j&gt; &lt;variable /&gt; &lt;entero /&gt; &lt;dir&gt;4&lt;/dir&gt; &lt;/j&gt; &lt;cuerpo&gt; &lt;leer&gt; &lt;p&gt;SRF 0 3&lt;/p&gt; &lt;p&gt;RD 1&lt;/p&gt; &lt;/leer&gt; &lt;leer&gt; &lt;p&gt;SRF 0 4&lt;/p&gt; &lt;p&gt;RD 1&lt;/p&gt; &lt;/leer&gt; &lt;seleccion&gt; &lt;condicion&gt; &lt;p&gt;SRF 0 3&lt;/p&gt; &lt;p&gt;DRF&lt;/p&gt; &lt;p&gt;SRF 0 4&lt;/p&gt;</pre>	<pre>&lt;p&gt;DRF&lt;/p&gt; &lt;p&gt;GT&lt;/p&gt; &lt;/condicion&gt; &lt;p&gt;JMF L1&lt;/p&gt; &lt;entonces&gt; &lt;escribir&gt; &lt;p&gt;SRF 0 3&lt;/p&gt; &lt;p&gt;DRF&lt;/p&gt; &lt;p&gt;WRT 1&lt;/p&gt; &lt;/escribir&gt; &lt;/entonces&gt; &lt;p&gt;JMP L2&lt;/p&gt; &lt;l&gt;L1:&lt;/l&gt; &lt;sino&gt; &lt;escribir&gt; &lt;p&gt;SRF 0 4&lt;/p&gt; &lt;p&gt;DRF&lt;/p&gt; &lt;p&gt;WRT 1&lt;/p&gt; &lt;/escribir&gt; &lt;/sino&gt; &lt;l&gt;L2:&lt;/l&gt; &lt;/seleccion&gt; &lt;/cuerpo&gt; &lt;p&gt;LVP&lt;/p&gt; &lt;/max&gt;</pre>
---	---

## VI. NOTAS

- La batería de pruebas de esta práctica puede estar compuesta de los programas diseñados para la práctica 2-3 y unos nuevos programas que complementen los ya implementados.
- Se valorará positivamente el que se escoja la opción de generación no secuencial (AST) y se apliquen técnicas de optimización (ver lección 5) en la generación de código.
- Considerad la posibilidad de realizar una pequeña aplicación con un menú interactivo (modo texto) que permita, especificando un fichero con código Pascual de entrada, encadenar todas las prácticas realizadas: compilación, generación de código e interpretación.