

EL CÓDIGO OBJETO (Código de tres direcciones):

El código objeto es a su vez una extensión del código intermedio utilizado en la práctica principal de la asignatura. Se añaden algunas instrucciones necesarias para generar el código requerido por el lenguaje PLX. Todas las variables del código intermedio se considera que están previamente definidas y que su valor inicial es 0.

El conjunto de instrucciones del código ensamblador, y su semántica son las siguientes:

Instrucción	Acción
<code>x = a ;</code>	Asigna el valor de a en la variable x
<code>x = a + b ;</code>	Suma los valores de a y b, y el resultado lo asigna a la variable x
<code>x = a - b ;</code>	Resta los valores de a y b, y el resultado lo asigna a la variable x
<code>x = a * b ;</code>	Multiplica los valores de a y b, y el resultado lo asigna a la variable x
<code>x = a / b ;</code>	Divide (div. entera) los valores de a y b, y el resultado lo asigna a la variable x
<code>x = a +r b ;</code>	Suma de dos valores reales
<code>x = a -r b ;</code>	Resta de dos valores reales
<code>x = a *r b ;</code>	Multiplicación de dos valores reales
<code>x = a /r b ;</code>	División de dos valores reales
<code>x = (int) a ;</code>	Convierte un valor real a, en un valor entero, asignándose a la variable x
<code>x = (float) a ;</code>	Convierte un valor entero a, en un valor real, asignándose a la variable x
<code>x = y[a] ;</code>	Obtiene el a-esimo valor del array y, asignando el contenido en x
<code>x[a] = b ;</code>	Coloca el valor b en la a-esima posición del array x
<code>x = *y ;</code>	Asigna a x el valor contenido en la memoria referenciada por y.
<code>*x = y ;</code>	Asigna el valor y en la posición de memoria referenciada por x.
<code>x = &y ;</code>	Asigna a x la dirección de memoria en donde está situado el objeto y.
<code>goto l ;</code>	Salto incondicional a la posición marcada con la sentencia "label l"
<code>if (a == b) goto l ;</code>	Salta a la posición marcada con la sentencia "label l", si y solo si el valor de a es igual que el valor de b
<code>if (a != b) goto l ;</code>	Salta a la posición marcada con la sentencia "label l", si y solo si el valor de a es distinto que el valor de b
<code>if (a < b) goto l ;</code>	Salta a la posición marcada con la sentencia "label l", si y solo si el valor de a es estrictamente menor que el valor de b.
<code>l:</code>	Indica una posición de salto.
<code>label l ;</code>	Indica una posición de salto. Es otra forma sintáctica equivalente a la anterior.
<code>function f :</code>	Indica una posición de salto para comienzo de una función.
<code>end f ;</code>	Indica el final del código de una función.
<code>param n = x;</code>	Indica que x debe usarse como parámetro n-simo en la llamada a la próxima función.
<code>x = param n ;</code>	Asigna a la variable x el valor del parámetro n-simo definido antes de la llamada a la función.
<code>call f ;</code>	Salto incondicional al comienzo de la función f. Al alcanzar la sentencia return el control vuelve a la instrucción inmediatamente siguiente a esta
<code>gosub l ;</code>	Salto incondicional a la etiqueta f. Al alcanzar la sentencia return el control vuelve a la instrucción inmediatamente siguiente a esta
<code>return ;</code>	Salta a la posición inmediatamente siguiente a la de la instrucción que hizo la llamada (call f) o (gosub l)
<code>write a ;</code>	Imprime el valor de a (ya sea entero o real)
<code>writec a ;</code>	Imprime el carácter Unicode correspondiente al número a
<code>print a ;</code>	Imprime el valor de a, y un salto de línea
<code>printc a ;</code>	Imprime el carácter Unicode correspondiente al número a, y un salto de línea
<code>error ;</code>	Indica una situación de error, pero no detiene la ejecución.
<code>halt ;</code>	Detiene la ejecución. Si no aparece esta instrucción la ejecución se detiene cuando se alcanza la última instrucción de la lista.
<code># ...</code>	Cualquier línea que comience con un # se considera un comentario.

En donde a,b representan tanto variables como constantes enteras, x,y representan siempre una variable, n representa un número entero, l representa una etiqueta de salto y f un nombre de función.

NOTA: Se han resaltado en negrita las nuevas instrucciones para el manejo de funciones.