Universidad de Oviedo

Diseño de Lenguajes de Programación

Grado en Ingeniería Informática del Software

Escuela de Informática de Oviedo

Lenguaje Fuente: Características

Nivel Léxico

- Comentarios
  - De una línea, comenzarán por el carácter almohadilla '#'
  - Multilínea, tipo C/C++ /* …. */
- Constantes
  - Enteras sin signo
  - Reales con punto decimal o mantisa y exponente.
  - De carácter (entre comillas simples).
- Identificadores
  - Formados por letras y dígitos (admitirá caracteres españoles) comenzando por una letra.

Lenguaje Fuente. Características.
Nivel Léxico

- Operadores
  - Aritméticos
    - +,-,*,/,% (módulo)
  - De comparación
    - eq, lt, le, gt, ge , ne
  - Lógicos
    - and, or , not
  - Indexación y de acceso a campos
    - [,], .
  - Asignación
    - =

Lenguaje Fuente. Características.
Nivel Léxico

- Otras elementos léxicos
  - (,),;,’,’
- Palabras reservadas
  - **types**
  - **globals**
  - **procedures**
  - **function**
  - **endfunction**
  - **as**
  - **main**
  - **endmain**

Lenguaje Fuente. Características.
Nivel Léxico

- Palabras reservadas (continuación)
  - **while**
  - **endwhile**
  - **if**
  - **else**
  - **endif**
  - **return**
  - **read**
  - **write**
  - **void**
  - **int**
  - **char**
  - **float**
  - **struct**
  - **endstruct**

Lenguaje Fuente. Características.

Nivel Sintáctico

- Declaración de variables de tipos simples **entero**, **real** y **carácter**.
- Declaración de variables de tipos compuestos: **arrays** y **estructuras**.
- Las variables podrán ser **globales** o **locales** a funciones.
- Se permite declaración múltiple de variables de un mismo tipo.

Lenguaje Fuente. Características.

Nivel Sintáctico

- Las funciones pueden recibir y devolver valores de tipo simple (entero, real o carácter).
- Sentencias:
  - Asignación.
  - Lectura y escritura múltiple de tipo simples por consola.
  - Invocación a funciones y procedimientos (paso por valor).
  - Bucles tipo while.
  - Condicionales tipo if-then-else.

Lenguaje Fuente. Características.

Nivel Sintáctico

- Expresiones:
  - Invocaciones a función (paso por valor).
  - Conversiones de tipo (cast).
  - Aritméticas (suma, resta, multiplicación, división, módulo).
  - Comparación (mayor, mayor o igual, menor, menor o igual, igual, distinto).
  - Lógicas (conjunciones, disyunciones o negación lógica).
  - Indexación.
  - Acceso a campos de un registro.

Lenguaje Fuente. Características.

Nivel Semántico

- Todas las comprobaciones semánticas de un lenguaje C/Pascal.
- Se verán más adelante.

Lenguaje Fuente. Características.

Generación de Código.

- Segmentos de memoria.
  - Datos: ocupa 1024 bytes (1KB).
  - Código: separado del de datos. Cada instrucción ocupa una dirección.
- Distribución del segmento de datos.
  - La memoria estática comienza en la dirección 0.
  - La pila comienza en la última dirección del segmento de datos (1023) y crece hacia abajo (direcciones de memoria más bajas).
- Registros:
  - IP: Puntero a instrucción que se está ejecutando actualmente.
  - SP: Apunta a la cima de la pila.
  - BP: Dirección (dirección de retorno y antiguo BP) de la función actual. **stack frame**

Lenguaje Fuente. Características.

Generación de Código.

- Tamaño de los tipos primitivos.
  - Carácter: 1 byte.
  - Entero: 2 bytes.
  - Real: 4 bytes.
  - Puntero (dirección de memoria): 2 bytes.

Lenguaje Fuente. Características.
Generación de Código.

- Instrucciones (introducción).
  - **PUSHA**: Colocar una dirección en la pila.
  - **PUSHI/F/B**: Coloca un valor entero/real/byte en la pila.
  - **INI/F/B**: Lee un valor entero/real/byte y lo mete en la pila.
  - **OUTI/F/B**: Muestra el valor entero/real/byte del tope de la pila y lo muestra en la consola.
  - **STOREI/F/B**: Quita de la pila
    - Un valor entero/real/byte.
    - Una dirección.
    - Almacena el valor en esa dirección.
  - **LOADI/F/B**:
    - Quita de la pila una dirección.
    - Pone en la pila el valor almacenado en la dirección anterior.

Lenguaje Fuente. Características.
Generación de Código.

- Instrucciones (introducción…).
  - **ADDI, SUBI, MULI, DIVI, MOD**, etc.: Quita dos operandos de la pila y apilan el resultado.
    - Existen instrucciones aritméticas para valores reales (**ADDF, SUBF, MULF, DIVF**).
  - Instrucciones de salto:
    - **JZ** "etiqueta": Salto condicional.
      - Salta a "etiqueta" si el valor almacenado en el tope de la pila es 0.
      - Quita ese valor de la pila.
    - **JMP** "etiqueta": Salto incondicional.
      - Salta a la etiqueta indicada.
      - No modifica la pila.

Lenguaje Fuente. Características.

Generación de Código.

- Instrucciones (introducción…).
  - Instrucciones de comparación:
    - **LTI**: Compara (**L**ess **T**han) los dos elementos superiores de la pila (a y b) suponiendo que ambos son enteros.
      - Retira ambos elementos de la pila una vez realizada la comparación.
      - Coloca un 0 en la pila si la comparación resultó falsa (**a<b es falso**).
    - **GTI, GTE, LEI, GEI, NEI**
    - Existen sus correspondientes para comparar valores reales (**LTF, GTF**, etc.).

Lenguaje Fuente. Características.
Generación de Código.

- Instrucciones (introducción…).
  - Instrucciones lógicas: **AND ,OR, NOT**
  - Conversión: **B2I, I2B, I2F, F2I**.
  - Manipulación de la pila: **POPI/F/B**
  - Funciones: **CALL, RET, ENTER**
  - Para la referencia completa leer el documento **Juego de Instrucciones.pdf** del capítulo **3. MAPL** del Tutorial.

Lenguaje Fuente. Características.
Generación de Código.

- Ejemplos
  - Ejemplo 1 código fuente

```
types
globals
  integer a;  #valdría tambien integer a,b;
  integer b;
procedures

main ()
  a = 2;
  b = 2*a;
  write(b);
endmain
```

Lenguaje Fuente. Características.
Generación de Código.

- Ejemplos
  - Ejemplo 1 código ensamblador MAPL

```
call main
halt

main:
 pusha 0
 pushi 2
 storei
 pusha 2
 pusha 0
 loadi
 pushi 2
 mul
 storei
 pusha 2
 loadi
 outi
 ret
```

Lenguaje Fuente. Características.
Generación de Código.

- Ejemplos
  - Ejemplo 2 código fuente

```
types
globals
  integer a;  #valdría tambien integer a,b;
  integer b;
procedures

main ()
  a = 0;
  b = 10;
  while (a lt b)
    a=a+1;
  endwhile
endmain
```

Lenguaje Fuente. Características.
Generación de Código.

- Ejemplos
  - Ejemplo 2 código ensamblador MAPL

```
call main
halt

main:
   pusha 0
   pushi 0
   storei
   pusha 2
   pushi 10
   storei
bucle:
   pusha 0
   loadi
   pusha 2
   loadi
   lti
   jz fin_bucle
   #continuación
   pusha 0
   pusha 0
   loadi
   pushi 1
   addi
   storei
   jmp bucle
fin_bucle:
   ret
```

Lenguaje Fuente. Características.
Generación de Código.

- Ejemplos
  - Ejemplo 3 código fuente

```
types
globals
  integer a,b;
procedures

main ()
  a = 2;
  b = 3;
  if (a lt b)
  then
    write(a);
  else
    write(b);
  endif
endmain
```

Lenguaje Fuente. Características.
Generación de Código.

- Ejemplos
  - Ejemplo 3 código ensamblador MAPL

```
call main
halt

main:
 pusha 0
 pushi 2
 storei
 pusha 2
 pushi 3
 storei
 pusha 0
 loadi
 pusha 2
 loadi
 lti
 #continuación
 jz parte_else
   pusha 0
   loadi
   outi
   jmp fin
   parte_else:
   pusha 2
   loadi
   outi
fin:
   ret
```