

Compiladores

Laboratorio 17

Objetivo

Trabajar con herramientas de flujo de control en el lenguaje Assembly para la arquitectura x86-64.

Laboratorio

E ha implementado la generación de código para la gramática:

- Program ::= Body
- Body ::= VarDecList StmtList
- VarDecList ::= (VarDec)*
- VarDec ::= "var" Type VarList ;
- Type ::= id
- VarList ::= id ("," id)*
- StmtList ::= Stmt (; Stmt)*
- Stmt ::= **id** = CExp |
 print (CExp)
 if CExp **then** Body [**else** Body]
- CExp ::= Exp [(< | <= | ==) Exp]
- Exp ::= Term ((+ | -) Term)*
- Term ::= Factor ((* | /) Factor)*
- Factor ::= **id** | **Num** | (Exp)

Se debe extender al uso de bucle:

- while i<10 do print(i); i=i+1 endwhile
- for(i=0, i <= 10, i=i+1) print(i) end for
- Sentencias anidadadas
- Se aceptan variables dentro de las sentencias de control, aunque no hay un problema con la cantidad de memoria a reservar. Asimismo, no asuman problemas con variables locales y globales.

Implementación WHILE:**while CExp do Body endwhile**

Instrucción	Descripción
while:	Etiqueta de inicio del bucle
Generar código de la condición	
cmpq \$0, %rax	
je endwhile	Salta al final si la condición es falsa
Genera código para el Body	
jmp while	Salta de vuelta al inicio del bucle
endwhile:	Etiqueta de fin del bucle

Implementación for:**for (AssignStatement1 , Cexp , AssignStatement2) Body**

Instrucción	Descripción
Genera código para AssignStatement1	
for:	Etiqueta que marca el inicio del bucle
Genera código para la condición	
cmpq \$0, %rax	Compara la condición con 0
je endfor	Si la condición es falsa, salta al final
Genera código para el Body	
Genera código para AssignStatement2	
jmp for	Salta al inicio para reevaluar la condición
endfor:	Etiqueta que marca el final del bucle