Facultad de Ingeniería
Semestre 2020-2
Compiladores
Proyecto Final

Alejandro Hernández Rodríguez

Manual

Para poder utilizar el compilador es necesario tener instalado python 3.7 o mayor. Y asignar permisos al programa compilador.py. Abajo se muestra la sintaxis para ejecutarlo desde la línea de comandos. Nótese que los picos paréntesis significan variables, es decir, pueden tomar otro valor.

./compilador.py <bandera(s)> <archivo a compilar>

Para el despliegue de información extra sobre la compilación se pueden utilizar las siguientes banderas.

-h : Bandera para mostrar la ayuda

-t : Con esta bandera se despliega el árbol de sintaxis abstracta, desarrollado por el compilador.

-a : Despliega el código ensamblador, generado por el compilador.

-l : Despliega la lista de tokens detectados,

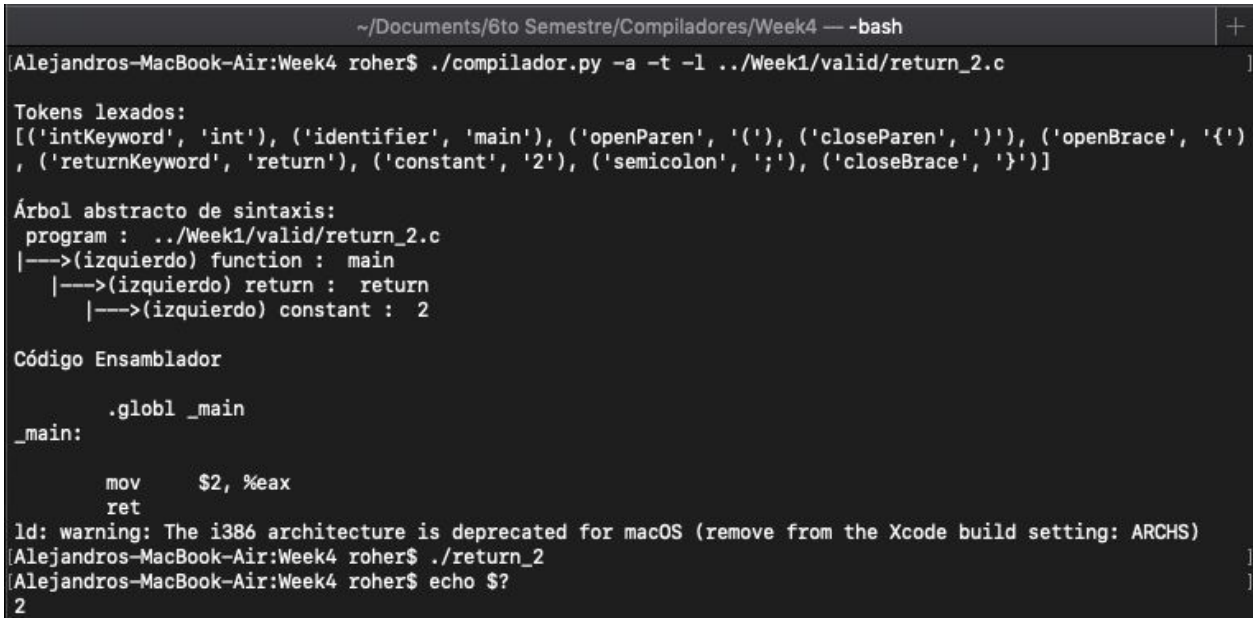
-n : Definir nombre del ejecutable

./compilador.py <bandera(s)> -n <nombre ejecutable> <archivo a compilar>

En las siguientes páginas se adjuntará una captura de pantalla con diferentes códigos de las distintas pruebas del blog de nora.

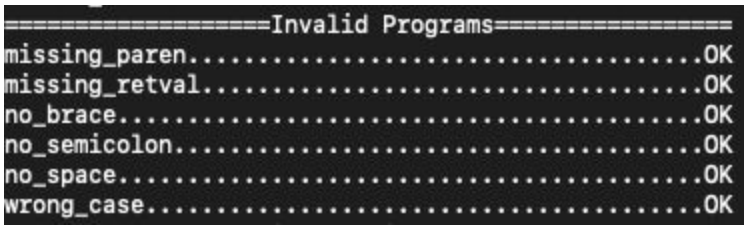
Ejemplo Semana 1

En el ejemplo siguiente se compilara el archivo multi_digit.c . Para poder ejecutarlo se utilizaría la sentencia



Test

Nora provee de un script para la prueba automática del compilador.



Ejemplo Semana 2

```
Alejandros-MacBook-Air:Week4 roher$ ./compilador.py -a -t -l ../Week2/valid/bitwise.c

Tokens lexados:
[('intKeyword', 'int'), ('identifier', 'main'), ('openParen', '('), ('closeParen', ')'), ('openBrace', '{')
, ('returnKeyword', 'return'), ('logicalNegation', '!'), ('constant', '12'), ('semicolon', ';'), ('closeBra
ce', '}')]

Árbol abstracto de sintaxis:
 program :  ../Week2/valid/bitwise.c
|--->(izquierdo) function :  main
   |--->(izquierdo) return :  return
      |--->(izquierdo) unaryop :  !
         |--->(izquierdo) constant :  12

Código Ensamblador

        .globl _main
_main:

        cmpl    $0, %eax
        movl    $0, %eax
        sete    %al
        mov     $12, %eax
        ret
ld: warning: The i386 architecture is deprecated for macOS (remove from the Xcode build setting: ARCHS)
```

Test

```
====================Invalid Programs==================
missing_const.....................................OK
missing_semicolon.................................OK
nested_missing_const..............................OK
wrong_order.......................................OK
```

Ejemplo Semana 3

```
[Alejandros-MacBook-Air:Week4 roher$ ./compilador.py -a -t -l ../Week3/valid/add.c

Tokens lexados:
[('intKeyword', 'int'), ('identifier', 'main'), ('openParen', '('), ('closeParen', ')'), ('openBrace', '{')
, ('returnKeyword', 'return'), ('constant', '1'), ('addition', '+'), ('constant', '2'), ('semicolon', ';'),
 ('closeBrace', '}')]

Árbol abstracto de sintaxis:
 program :  ../Week3/valid/add.c
|--->(izquierdo) function :  main
   |--->(izquierdo) return :  return
      |--->(izquierdo) binaryop :  +
         |--->(izquierdo) constant :  1
         |--->(derecho) constant :  2

Código Ensamblador

        .globl _main
_main:

        mov     $1, %eax
        push    %eax
        mov     $2, %eax
        pop     %ecx
        addl    %ecx, %eax
        ret
ld: warning: The i386 architecture is deprecated for macOS (remove from the Xcode build setting: ARCHS)
[Alejandros-MacBook-Air:Week4 roher$ ./add
[Alejandros-MacBook-Air:Week4 roher$ echo $?
3
```

Test

```
====================Invalid Programs==================
malformed_paren.....................................OK
missing_first_op....................................OK
missing_second_op...................................OK
no_semicolon........................................OK
====================Stage 3 Summary===================
```

Ejemplo Semana 4

```
Alejandros-MacBook-Air:Week 4 roher$ ./compilador.py -l -t -a valid/and_true.c

Tokens lexados:
[('intKeyword', 'int'), ('identifier', 'main'), ('openParen', '('), ('closeParen', ')'), ('openBrace', '{')
, ('returnKeyword', 'return'), ('constant', '1'), ('AND', '&&'), ('negation', '-'), ('constant', '1'), ('se
micolon', ';'), ('closeBrace', '}')]

Árbol abstracto de sintaxis:
 program :  valid/and_true.c
|--->(izquierdo) function :  main
   |--->(izquierdo) return :  return
      |--->(izquierdo) binaryop :  &&
         |--->(izquierdo) constant :  1
         |--->(derecho) unaryop :  -
            |--->(izquierdo) constant :  1

Código Ensamblador

        .globl _main
_main:

        mov     $1, %eax
        cmpl    $0, %eax
        je      _clause2
        jmp     _end
_clause2:
        neg     %eax
        cmpl    $0, %eax
        movl    $0, %eax
        setne   %al
_end:
        ret
ld: warning: The i386 architecture is deprecated for macOS (remove from the Xcode build setting: ARCHS)
Alejandros-MacBook-Air:Week 4 roher$ ./and_true
Alejandros-MacBook-Air:Week 4 roher$ echo $?
1
```

Test

```
==================Invalid Programs================
missing_first_op.................................OK
missing_mid_op...................................OK
missing_second_op................................OK
missing_semicolon................................OK
```