

Core 4.0 Implementation of the if Construct

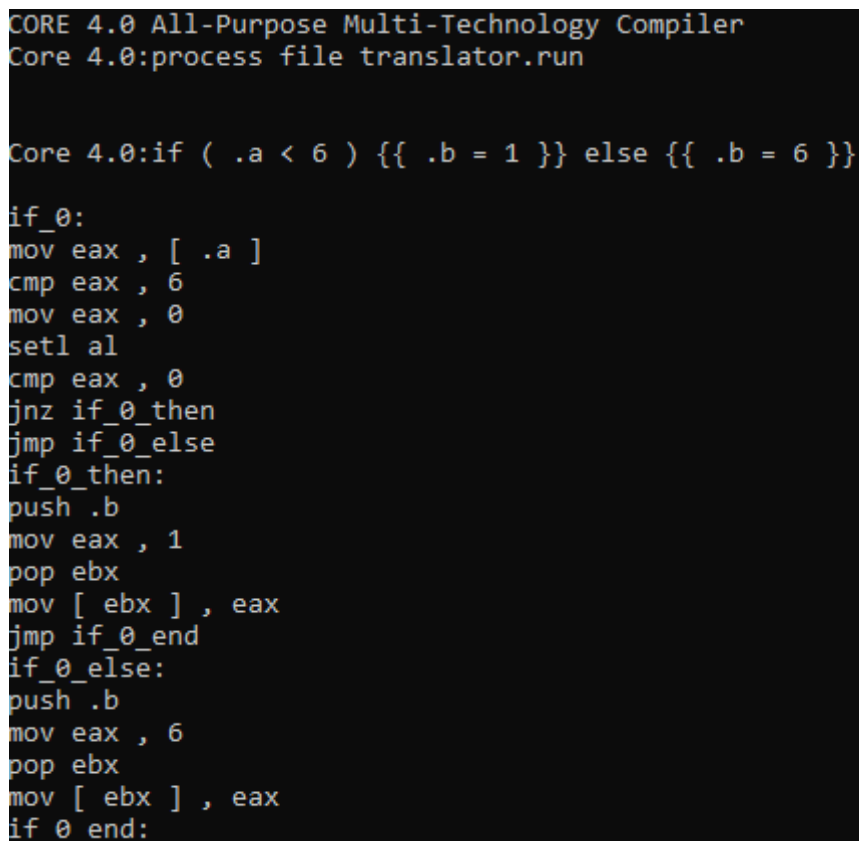
Today we deal with the if which has the following syntax:

```
if&: if|(str) (|(.CONDITION)(**) )|(str) {{|.STATEMENT|(**) ;|(**)|(opt) }}|(str) else|(str)|(trc) {{|.ELSE|(**) ;|(**)|(opt) }}
```

As you can read and understand, we have a condition, a set of statements to execute if the condition is true, and another set of statements if it is false, if the block is defined. If it is not defined you will jump directly to the end of the instructions for this construct. The implementation follows that of for, but with some modifications. Here it is:

```
public function if.autoexec ( null ) {{  
    set ::id = "if_" + exec unique_id ( null ) ;  
    set ::code = ::id + ":" + ::CONDITION.code + "\ncmp eax , 0\njnz " + ::id + "_then\njmp " + ::id + "_else\n" + ::id + "_then:" ;  
    for ( set .i = 0 ; .i < size ::STATEMENT ; .i = .i + 1 ) {{ ::code = ::code + ::STATEMENT[ .i ].code }} ;  
    ::code = ::code + "\njmp " + ::id + "_end" + "\n" + ::id + "_else:" ;  
    if ( ::ELSE ) {{ for ( set .i = 0 ; .i < size ::ELSE ; .i = .i + 1 ) {{ ::code = ::code + ::ELSE[ .i ].code }} }} ;  
    ::code = ::code + "\n" + ::id + "_end:" ;  
    case ( ::_CALLER type SYSstart ) {{ print ::code }}  
}}
```

And this is the product code:



```
CORE 4.0 All-Purpose Multi-Technology Compiler  
Core 4.0:process file translator.run  
  
Core 4.0:if ( .a < 6 ) {{ .b = 1 }} else {{ .b = 6 }}  
  
if_0:  
mov eax , [ .a ]  
cmp eax , 6  
mov eax , 0  
setl al  
cmp eax , 0  
jnz if_0_then  
jmp if_0_else  
if_0_then:  
push .b  
mov eax , 1  
pop ebx  
mov [ ebx ] , eax  
jmp if_0_end  
if_0_else:  
push .b  
mov eax , 6  
pop ebx  
mov [ ebx ] , eax  
if_0_end:
```

If the ELSE is not present, we have:

```
Core 4.0:if ( .a < 6 ) {{ .a = 0 }}  
  
if_1:  
mov eax , [ .a ]  
cmp eax , 6  
mov eax , 0  
setl al  
cmp eax , 0  
jnz if_1_then  
jmp if_1_else  
if_1_then:  
push .a  
mov eax , 0  
pop ebx  
mov [ ebx ] , eax  
jmp if_1_end  
if_1_else:  
if_1_end:
```

To close the discussion of examples, functions, definitions of variables and vectors in the stack and optimization are missing. Having done this, I would close the experience of this demo, which is really leading to the implementation of a simple small but complete compiler.

Thank you