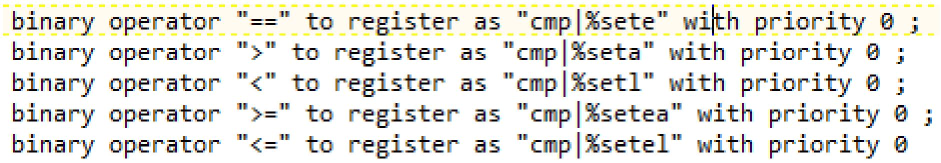
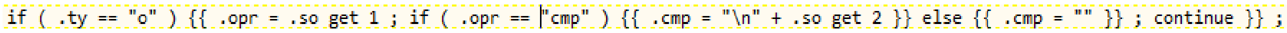
# Core 4.0 : Example of Conditional Expressions

Again as an exercise and example we will deal with the conditional operators "==" ">" "<" "<=" ">=". To do this we first insert these operators:



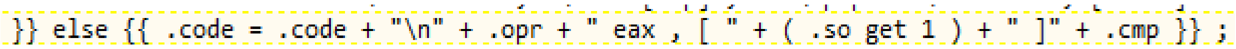
```
binary operator "==" to register as "cmp|%sete" with priority 0 ;
binary operator ">" to register as "cmp|%seta" with priority 0 ;
binary operator "<" to register as "cmp|%setl" with priority 0 ;
binary operator ">=" to register as "cmp|%setea" with priority 0 ;
binary operator "<=" to register as "cmp|%setel" with priority 0
```

And we modify the translate function as follows.



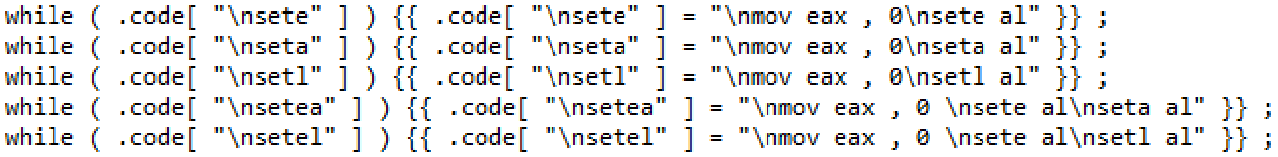
```
if ( .ty == "o" ) {{ .opr = .so get 1 ; if ( .opr == "cmp" ) {{ .cmp = "\n" + .so get 2 }} else {{ .cmp = "" }} ; continue }} ;
```

Now in various parts of the translation code we will add this new .cmp variable (for example)



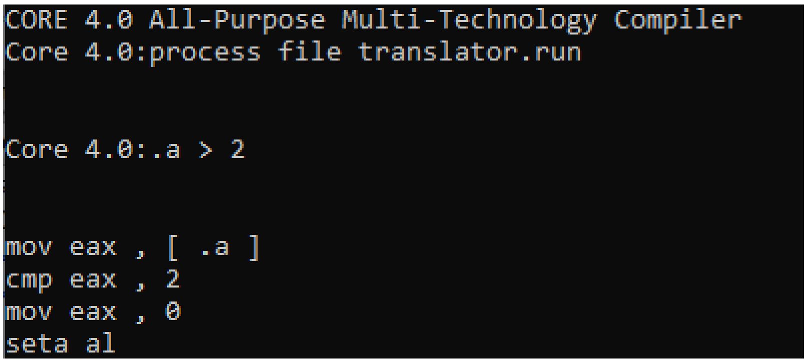
```
}} else {{ .code = .code + "\n" + .opr + " eax , [ " + ( .so get 1 ) + " ]" + .cmp }} ;
```

Before running the code, we will perform these substitutions:



```
while ( .code[ "\nsete" ] ) {{ .code[ "\nsete" ] = "\nmov eax , 0\nsete al" }} ;
while ( .code[ "\nseta" ] ) {{ .code[ "\nseta" ] = "\nmov eax , 0\nseta al" }} ;
while ( .code[ "\nsetl" ] ) {{ .code[ "\nsetl" ] = "\nmov eax , 0\nsetl al" }} ;
while ( .code[ "\nsetea" ] ) {{ .code[ "\nsetea" ] = "\nmov eax , 0 \nsete al\nseta al" }} ;
while ( .code[ "\nsetel" ] ) {{ .code[ "\nsetel" ] = "\nmov eax , 0 \nsete al\nsetl al" }} ;
```

And we're done. Let's launch the application:

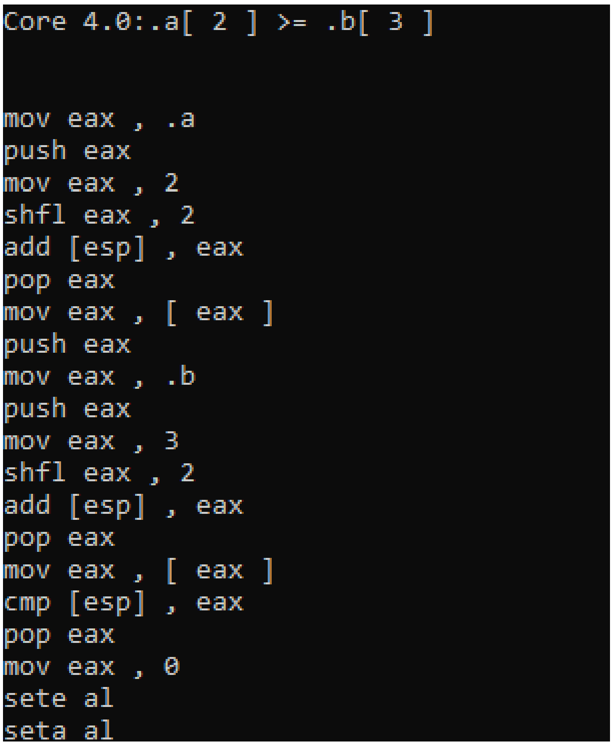


```
CORE 4.0 All-Purpose Multi-Technology Compiler
Core 4.0:process file translator.run


Core 4.0:.a > 2


mov eax , [ .a ]
cmp eax , 2
mov eax , 0
seta al
```

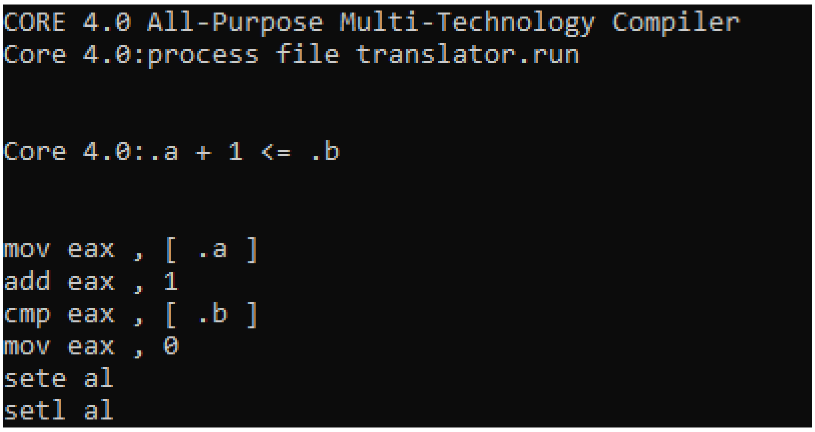
And we can complicate it as much as we want:



```
Core 4.0:.a[ 2 ] >= .b[ 3 ]


mov eax , .a
push eax
mov eax , 2
shfl eax , 2
add [esp] , eax
pop eax
mov eax , [ eax ]
push eax
mov eax , .b
push eax
mov eax , 3
shfl eax , 2
add [esp] , eax
pop eax
mov eax , [ eax ]
cmp [esp] , eax
pop eax
mov eax , 0
sete al
seta al
```

But I prefer to keep it simple and understandable:



```
CORE 4.0 All-Purpose Multi-Technology Compiler
Core 4.0:process file translator.run


Core 4.0:.a + 1 <= .b


mov eax , [ .a ]
add eax , 1
cmp eax , [ .b ]
mov eax , 0
sete al
setl al
```

We have also implemented conditional operators. I think we are ready now and will do so in the next few days to implement some constructs like “if”.

Good evening