While, Do While, For loops in Assembly Language (emu8086)

Asked 7 years, 7 months ago Modified 7 months ago Viewed 192k times

33

12

I want to convert simple loops in high-level languages into assembly language (for emu8086) say, I have this code:

```
for(int x = 0; x<=3; x++)
{
 //Do something!
}
```

or

```
int x=1;
do{
//Do something!
}
while(x==1)
```

or

```
while(x==1){
//Do something
}
```

edited Aug 20, 2019 at 3:16
Peter Cordes
297k 42 541 766

asked Feb 23, 2015 at 1:07
Glynn Bacanto
399 1 5 12

ns and conditional jumps in assembly already, rewrite the code using `if` and `goto` first and/or create a flowchart.

3, 2015 at 1:19

Siegler Feb 23, 2015 at 1:27

1 Just lose the e from the register name. e stands for extended (I believe) - it indicates 32 bit-wide registers instead of 16 bits. – 500 - Internal Server Error Feb 23, 2015 at 2:44

1 Yes, apart from whatever else you need in the body. For example a do-while is a single conditional jump `start: { body }; if (condition) goto start;` – Jester Jul 10, 2021 at 11:48

2 Answers

Sorted by: Highest score (default)

66

For-loops:

For-loop in C:

```
for(int x = 0; x<=3; x++)
{
    //Do something!
}
```

The same loop in 8086 assembler:

```
        xor cx,cx   ; cx-register is the counter, set to 0
loop1   nop         ; Whatever you wanna do goes here, should not change cx
        inc cx      ; Increment
        cmp cx,3    ; Compare cx to the limit
        jle loop1   ; Loop while less or equal
```

That is the loop if you need to access your index (cx). If you just wanna to something 0-3=4 times but you do not need the index, this would be easier:

```
oes here, should not change cx
ents cx and jumps to label if not 0
```

ction a constant amount of times, you could also use an assembler-directive which will just hardcore that

```
int x=1;
```

```
do{
    //Do something!

}
while(x==1)
```

The same loop in assembler:

```
        mov ax,1
loop1   nop         ; Whatever you wanna do goes here
        cmp ax,1    ; Check wether cx is 1
        je loop1    ; And loop if equal
```

While-loops

While-loop in C:

```
while(x==1){
    //Do something
}
```

The same loop in assembler:

```
        jmp loop1   ; Jump to condition first
cloop1  nop         ; Execute the content of the loop
loop1   cmp ax,1    ; Check the condition
                                      loop if met
```

er because it is pretty much standard. For the other loop conditions you can take a register of your liking. with all the instructions you wanna perform in the loop.

answered Mar 21, 2015 at 17:47

Maximilian Schier
1,489 14 18

rn CPUs.. – Peter Cordes Nov 8, 2017 at 1:37

whenever possible, for the same reason compilers do: code runs faster with fewer instructions inside the loop. (Usually nping to the bottom of the loop like you're doing here in your `while` loop.) – Peter Cordes Jan 14, 2018 at 4:41

to the loop condition check before the loop's first iteration. – ecm Aug 28, 2019 at 19:33

mization that `x<=3` is known to be true on the first iteration, allowing that check to be skipped. That's 100% standard you know they will run at least once, because of fixed loop bounds. Other kinds of for loops may need to sometimes ops that follow the idiomatic pattern for looping x from 0 .. n. – Peter Cordes Apr 9, 2021 at 1:23

-3

```
Do{
    AX = 0
    AX = AX + 5
    BX = 0
    BX= BX+AX
} While( AX != BX)
```

Do while loop always checks the loop the condition at the end of each iteration.

Share Improve this answer Follow

edited Apr 15, 2021 at 7:21
Dilan
2,453 7 21 31

answered Apr 9, 2021 at 0:06
DHS2020
1

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.