Heimadæmi 7

Arnar Sigurðsson

1. a) Minnsta staðlaða er þar sem exp hlutinn er 0 og frac hlutinn er 1 svo 0 00000 11111.

b) 1 00101.01000 svo 1.0101000 \* 2^2. Bias = (2^4)-1 = 15.

E = 15+2 = 17 = 10001 = exp og mantissan er 01010 svo á réttu sniði :

1 10001 01010 eða -1 \* (1 + 0.25 + 1/16) \* 2^2 = 5.25

c)1.0 = 1.0\*2^0 og E=15 = 01111 svo 0 01111 00000 og

2.0 = 10.0 = 1.0\*2^1 og e=16 = 10000 svo 0 10000 00000 svo sá fjöldi sem hægt er að tákna er það sem mantissan getur táknað sem er 2^1 fyrir einn bitann, 2^2 fyrir tvo bita svo 5 bitar = 2^5 = 32 mismunandi gildi á milli 1 og 2 og svo bætist við 1 þar sem 1 er tekinn með í [1-2) svo alls eru þetta 33 gildi.

d) NaN er þegar exp = 11111 og frac ekki jafnt og 00000 svo það eru 11111 = 31 og svo getur formerkjabitinn verið 1 svo 31\*2 = 62 möguleg NaN.

1. a)% esi er 4 bæti svo það er tag long og sama með %edi svo tag x og y er long eða unsigned long

(líklega unsigned þar sem notað er unsigned aðgerð sienna í kóðanum). ja þýðir að ef %esi í

cmp %esi, %edi er stærra en %edi er hoppað í L2 og gildinu í %eax skilað sem er breytan y. Annnars er breytan x í %edi færð í %eax og henni skilað.

b) cmpl ber long/unsigned long x=%edi við y=%esi eða x-y. je athugar hvort x==y og ef svo er hoppar það í L3 og skilar gildinu í x. Annars skilar það gildinu í y.

1. a) L7 er for-loopan.

#5 leggur saman %rax og %rdx og setur í %rcx. Þetta er 0+0=0 í byrjun út af upphafsstillingunni.

b) #6 og-ar fyrstu bæti í %dl í %rdx við 7 eða 0111.

#7 ef %dl = 7 þá er %rcx sett í skilagildi %rax

#8 bætir 1 við teljarann %rdx.

#9 ber saman viðfangið n og teljarann %rdx.

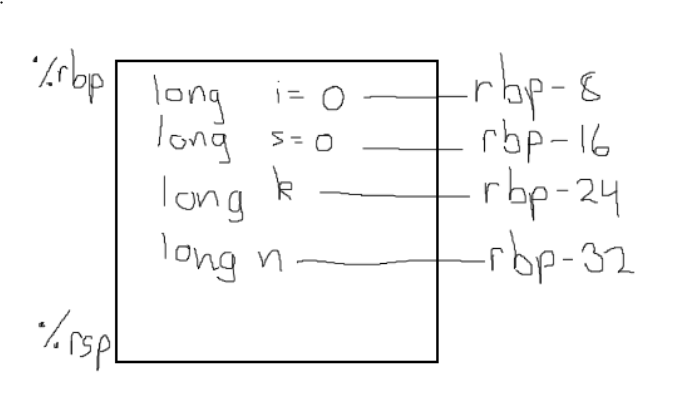
# Ef viðfangið og teljarinn eru ekki eins fer loopan annan hring. Ef eins skilast gildið í %rax sem er summa talna minna en eða jafnt og n þar sem síðustu 3 bitarnir eru 111. Fyrir n=26 væru það

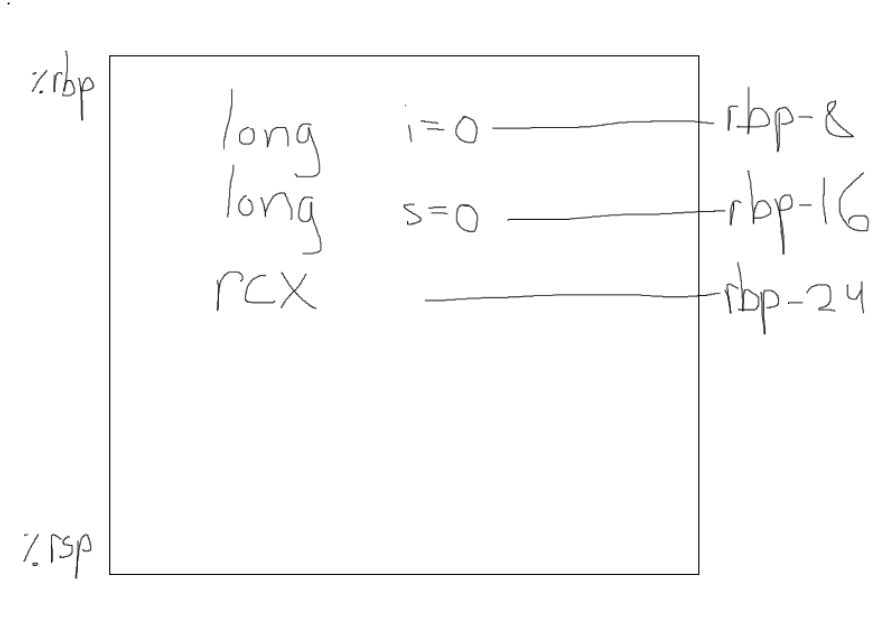
7(111) + 15 (1111) + 23(10111).

1. a) Það breytist úr stökktöflu ef fjarlægt er eitt tilfelli eða case.

b) Ef breytt gildi á 5 í 41 er hætt að nota töflu.

c) Þá bætast við 5 cases eða tilfelli.

1. a)



b)

c) Í -Og þarf 3 skipanir til að fara í gegnum for-loopuna eftir að búið er að upphafsstilla gildi og athuga hvort n sé 0 en í því tilfelli er hægt að sleppa loopunni. Auk þess eru 2 reikniaðgerðir sem þarf að gera inni í loopunni.

Í -O1 þarf 3 skipanir til að fara í gegnum for-loopuna eftir að búið er að upphafsstilla gildi og athuga hvort n sé 0 eins og í fyrra. Reikningarnir eru þó aðeins öðruvísi en k + 2\*n er geymt í gistinu rcx.