



**HÁSKÓLI ÍSLANDS**

## Forritunarverkefni II

Sævar Öfjörð Magnússon

16. febrúar 2006

08.31.18 Tölvutækni  
Kennari: Karl Guðmundsson

# 1 Verkefni

Skrifið forrit fyrir Easy68k herminn til að lesa innslegin gildi og setja þau á stafla. Þegar staflinn er fullur á forritið að tæma staflann sjálfkrafa og prenta gildin á skjáinn. Þið þurfið að skrifa push og pop rútnur sem notast við post-increment mode fyrir pop og pre-decrement mode fyrir push. Staflinn á að vera fyrir 10 gildi og push og pop eiga að framkvæma error checking. Lesið kafla 2.8 og notið dæmin þar til hliðsjónar.

## 1.1 Skilyrði

1. Notið subrútnur fyrir kóðann ykkar.
2. Notið post-increment og pre-decrement við aðgerðir.
3. Framkvæmið villu próf við pop og push.
4. Sækið innslegin gildi eitt í einu frá lyklaborðinu.
5. Sýnið innihald staflans á einni línu með bil á milli gilda þegar þið tæmið staflann.

# 2 Lausn verkefnis

## 2.1 Sauðakóði

Byrjað var á að útbúa sauðakóða sem sýna átti helstu virkni forrits:

---

```
1 MAIN forrit :
2     Láta stack pointer benda á ákveðna addressu utan stacksins
3     Endurtaka endalaust :
4         Biðja um gildi frá notanda
5         pusha gildinu í stack
6 FLUSHSTACK rúttína :
7     Endurtaka endalaust :
8         poppa stackinn
9         birta gildið sem poppað var
10 PUSH rúttína :
11     Athuga hvort stack sé fullur
12         ef fullur, fara í FLUSHSTACK rúttínu
13         annars pusha gildinu á stackinn
14     hætta
15 POP rúttína :
16     Athuga hvort stack sé tómur
17         ef tómur, fara í MAIN forrit
18         annars poppa efsta gildinu af stacknum
19     hætta
```

---

Mér fannst eðlilegast að láta forritið ganga í endalausri lykkju. Flæði forritsins stýrist svo í rauninni af villuprófunum í POP og PUSH aðferðunum og auðvitað stærð staflans sem ég skilgreini sem 10 stök.

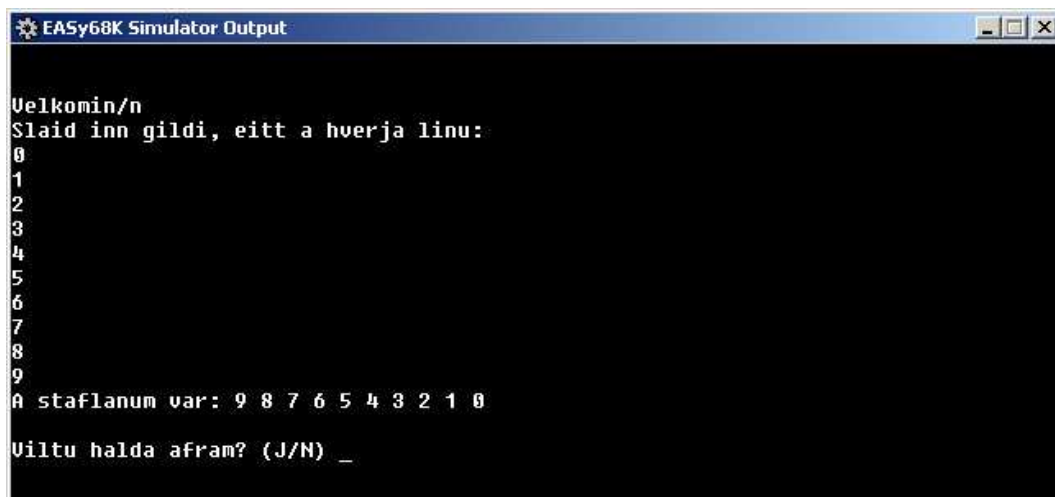
Ég útfærði svo þennan sauðakóða í smalamáli fyrir Motorola 68020 í Easy68K herminum.

## 2.2 Niðurstöður

Í raun var lítið sem fór úrskeiðis hjá mér, það virkaði flest allt í fyrstu tilraun hjá mér. Það var reyndar eitt atriði sem ég sá ekki fyrir sem ég man eftir. Það var að *Postincrement* og *Predecrement* vistfangahættirnir hækkuðu og lækkuðu bendinn um 4 í hvert sinn, sem er eðlilegt þegar verið er að nota *MOVE.L*, ég breytti því frekar skipuninni í *MOVE.B* í staðinn fyrir að nota 32 bita minnishólf, þar sem það sparar „mikið“ minni.

Ég held að forritið mitt virki bara fint. Eftir að ég var búinn að útfæra það samkvæmt sauðakóðanum hér að ofan fór ég að bæta við aukafíðusum eins og að spyrja notandann hvort hann vilji fylla staflann aftur og hreinsa þá skjáinn eftir hverja törn. Það kom vel út að nota endalausar lykkjur og láta villutékkinn í *POP* og *PUSH* sjá um aðal flæði forritsins.

Í heildina séð var gaman að útfæra þessar gagnagrindur sem maður þekkir úr tölvunarfræði 2 og sjá sína eigin útfærslu virka eins og maður hafði lært.



```
EASy68K Simulator Output
Velkomin/n
Slaid inn gildi, eitt a hverja linu:
0
1
2
3
4
5
6
7
8
9
A staflanum var: 9 8 7 6 5 4 3 2 1 0
Viltu halda afram? (J/N) _
```

Mynd 1: Dæmi um úttak úr forritinu

# A Smalamálskóði

## A.1 forritunarverkefni2.X68

```
1 *-----
2 * Program Number: Forritunarverkefni 2
3 * Written by      : Sævar Öfjörð Magnússon
4 * Date Created   : 2005-02-17
5 * Description    :
6 *
7 * Þetta forrit er útfærsla á stack gagnagrindinni.
8 * Notandinn er beðinn um 10 gildi sem sett eru á stack.
9 * Þegar stackurinn er orðinn fullur eru gildin poppuð af
10 * samkvæmt reglum gagnagrindarinnar með pop aðferðinni og
11 * birt á skjánum.
12 *
13 *-----
14
15      org      $800
16 CR     equ     $0d          ; carriage return
17 LF     equ     $0a          ; line feed
18 welcome dc.b   CR,LF,CR,LF, 'Velkomin/n',CR,LF
19      dc.b     'Slaid_inn_gildi ,_eitthvað_hverja_línu: ',CR,LF,0
20 display dc.b   'A_staflanum_var: ',0
21 newline dc.b   CR,LF,0
22 cont   dc.b   CR,LF,CR,LF, 'Viltu_halda_afra?_(J/N) ',0
23
24 start  org     $1000
25      bra     MAIN          ; förum beint í main forrit
26
27 FLUSHSTACK:
28      lea     display ,A1
29      move.b #14,D0          ; trap kóði 14
30      trap   #15
31 LOOP   bsr    POP          ; poppum stackinn
32      move.b #6,D0           ; trap kóði 6
33      trap   #15            ; birtum það sem er í D1
34      move.b #$20,D1        ; setjum stafabil í D1
35      trap   #15            ; birtum það sem er í D1
36      bra    LOOP
37
38 POP:
39      cmp.l  #$200A,A0      ; Athugum hvort sp sé stærri en $2028
40      bge   AGAIN
41      move.b (A0)+,D1       ; Tökum efsta stakið af og setjum í D1
42      rts
43
44 PUSH:
45      move.b D1,-(A0)       ; Setjum það sem er í D1 í stackinn
46      lea   newline ,A1    ; addressan á newline í A1
```

```

47     move.b  #14,D0           ; trap kóði 13
48     trap   #15
49     cmp.l  #$2000,A0        ; Athugum hvort sp sé orðinn $2000
50     ble   FLUSHSTACK       ; Sturtum úr stackinum ef fullur
51     rts
52
53 MAIN  move.l  #$FF00,D1
54     move.b  #11,D0
55     trap   #15              ; clear display
56     lea   welcome,A1       ; addressan á welcome streng í A1
57     move.b  #14,D0          ; trap kóði 14
58     trap   #15
59     move.l  #$200A,A0       ; Byrjum á að skilgreina stackinn sem tóman.
60 FOREVER move.b  #5,D0
61     trap   #15              ; biðjum um eitt gildi
62     bsr   PUSH              ; Pushum innslegnu gildi
63     bra   FOREVER          ; Keyrum lykkju endalaust
64
65 AGAIN lea    cont,A1
66     move.b  #14,D0
67     trap   #15              ; prentum "viltu_halda_áfram?"
68     move.b  #5,D0
69     trap   #15              ; biðjum um já eða nei
70     and   #$DF,D1          ; breytum í hástaf
71     cmp.b  #$4a,D1         ; athugum hvort það sé J
72     beq   MAIN
73     cmp.b  #$4e,D1         ; athugum hvort það sé N
74     beq   END
75     bra   AGAIN
76
77 END   move.b  #9,D0
78     trap   #15              Halt Simulator
79
80     end    start
81
82
83
84 *~Font name~Courier New~
85 *~Font size~10~
86 *~Tab type~1~
87 *~Tab size~8~

```