

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Υπολογιστών

Παράδειγμα εκτέλεσης στην ΚΜΕ

(εύρεση μεγίστου ακεραίου σε πίνακα)

<http://mixstef.github.io/courses/csintro/>

Μ.Στεφανιδάκης



Το παράδειγμα

- Εύρεση μεγίστου σε λίστα 10 ακεραίων
 - C ints, 32 bits ο καθένας
 - Κάθε στοιχείο απέχει 4 bytes από το προηγούμενο
- Η γλώσσα assembly ανήκει στην αρχιτεκτονική x86 (των PC)
 - `eax`, `ecx` και `edi` είναι ονόματα καταχωρητών 32-bit
 - Οι αγκύλες `[..]` υποδηλώνουν προσπέλαση μνήμης
- Το πρόγραμμα μπορεί να γραφεί και με πιο αποδοτικό τρόπο

Το πρόγραμμα σε assembly

```
<_start>
8048080: a1 ac 90 04 08      mov     eax,[0x80490ac]      ; max = list[0]
8048085: bf 01 00 00 00      mov     edi,0x1             ; i = 1
<again>
804808a: 8b 0c bd ac 90 04 08 mov     ecx,[edi*4+0x80490ac] ; new = list[i]
8048091: 39 c8               cmp     eax,ecx             ; if max >= new...
8048093: 7d 02               jge     8048097             ; ...then goto <skip>
8048095: 89 c8               mov     eax,ecx             ; max = new
<skip>
8048097: 83 c7 01           add     edi,0x1             ; i = i + 1
804809a: 83 ff 0a           cmp     edi,0xa             ; if i!=10...
804809d: 75 eb             jne     804808a             ; ...then goto <again>
<_end>
804809f:
```

- Το λειτουργικό σύστημα θα ξεκινήσει την εκτέλεση στη θέση `_start`
- Μετά το `_end` ακολουθούν εντολές τερματισμού του προγράμματος

Τα δεδομένα (η λίστα 10 ακεραίων)

<list>

0x80490ac:	0x0000000b	; list[0] = 11
0x80490b0:	0xffffffffd	; list[1] = -3
0x80490b4:	0x0000001b	; list[2] = 27
0x80490b8:	0x00000023	; list[3] = 35
0x80490bc:	0xffffffffbd	; list[4] = -67
0x80490c0:	0xffffffffe4	; list[5] = -28
0x80490c4:	0x00000007	; list[6] = 7
0x80490c8:	0xffffffffd	; list[7] = -3
0x80490cc:	0x0000002b	; list[8] = 43
0x80490d0:	0xfffffffff	; list[9] = -1

- 10 C int (4 bytes ο κάθε αριθμός)
- Σε πίνακα (συνεχόμενες θέσεις μνήμης)

Η εκτέλεση βρίσκεται στο `_start`

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048080

program
counter

C S Z V

flags

00000000

eax

00000000

ecx

00000000

edi

max = list[0]

<_start>

8048080: **mov** **eax**, [80490ac]

8048085: **mov** **edi**, 0x1

<again>

804808a: **mov** **ecx**, [edi*4+80490ac]

8048091: **cmp** **eax**, **ecx**

8048093: **jge** 8048097

8048095: **mov** **eax**, **ecx**

<skip>

8048097: **add** **edi**, 0x1

804809a: **cmp** **edi**, 0xa

804809d: **jne** 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffff fffffffe 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048085

program
counter

C S Z V

flags

0000000b

eax

00000000

ecx

00000000

edi

$i = 1$

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804808a

program
counter

C S Z V

flags

0000000b

eax

00000000

ecx

00000001

edi

new = list[i]

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd ffffffff4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048091

program
counter

C S Z V

flags

0000000b

eax

ffffffffd

ecx

00000001

edi

1*4+80490ac =
80490b0

max - new (compare)

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: **cmp eax,ecx**

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048093

program
counter

C S Z V

flags

0000000b>fffffffd
(11>-3)

0000000b

eax

fffffffd

ecx

00000001

edi

branch if max \geq new

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: **jge 8048097**

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048097

program
counter

C S Z V

flags

branch if
S flag == V flag

0000000b

eax

fffffffbd

ecx

00000001

edi

$$i = i + 1$$

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804809a

program
counter

C S Z V

flags

0000000b

eax

fffffffffd

ecx

00000002

edi

i – 10 (compare)

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804809d

program
counter

C S Z V

flags

2<10

0000000b

eax

00000000

ecx

00000002

edi

branch if i == 10

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804808a

program
counter

C S Z V

flags

branch if
Z flag == 0

0000000b

eax

fffffffffd

ecx

00000002

edi

new = list[i]

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048091

program
counter

C S Z V

flags

0000000b

eax

0000001b

ecx

00000002

edi

2*4+80490ac =
80490b4

max - new (compare)

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: **cmp eax,ecx**

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffffbd

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048093

program
counter

C S Z V

flags

0000000b<0000001b
(11<27)

0000000b

eax

0000001b

ecx

00000002

edi

branch if max \geq new

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: **jge 8048097**

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048095

program
counter

C S Z V

flags

branch if
S flag == V flag

0000000b

eax

0000001b

ecx

00000002

edi

max = new

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048097

program
counter

C S Z V

flags

0000001b

eax

0000001b

ecx

00000002

edi

$$i = i + 1$$

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffffbd

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804809a

program
counter

C S Z V

flags

0000001b

eax

0000001b

ecx

00000003

edi

i – 10 (compare)

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804809d

program
counter

C S Z V

flags

3<10

0000001b

eax

0000001b

ecx

00000003

edi

branch if i == 10

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804808a

program
counter

C S Z V

flags

branch if
Z flag == 0

0000001b

eax

0000001b

ecx

00000003

edi

new = list[i]

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048091

program
counter

C S Z V

flags

0000001b

eax

00000023

ecx

00000003

edi

3*4+80490ac =
80490b8

max - new (compare)

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: **cmp eax,ecx**

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048093

program
counter

C S Z V

flags

0000001b<00000023
(27<35)

0000001b

eax

00000023

ecx

00000003

edi

branch if max \geq new

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: **jge 8048097**

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048095

program
counter

C S Z V

flags

branch if
S flag == V flag

0000001b

eax

00000023

ecx

00000003

edi

max = new

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

08048097

program
counter

C S Z V

flags

00000023

eax

00000023

ecx

00000003

edi

$$i = i + 1$$

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804809a

program
counter

C S Z V

flags

00000023

eax

00000023

ecx

00000004

edi

i – 10 (compare)

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd ffffffffe4 00000007 ffffffffdd

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804809d

program
counter

C S Z V

flags

4<10

00000023

eax

00000023

ecx

00000004

edi

branch if i == 10

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd ffffffffe4 00000007 ffffffffdd

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804808a

program
counter

C S Z V

flags

branch if
Z flag == 0

00000023

eax

00000023

ecx

00000004

edi

Στη συνέχεια...

- Η διαδικασία επαναλαμβάνεται μέχρι να ολοκληρωθούν οι συγκρίσεις έως και το τελευταίο στοιχείο του πίνακα ακεραίων (`list[9]`)
- Τα βήματα αυτά παραλείπονται, στην επόμενη διαφάνεια έχει τελειώσει και η σύγκριση με το `list[9]`

$$i = i + 1$$

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffffbd

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804809a

program
counter

C S Z V

flags

0000002b

eax

ffffffff

ecx

0000000a

edi

i – 10 (compare)

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804809d

program
counter

C S Z V

flags

10==10

0000002b

eax

00000023

ecx

0000000a

edi

branch if i == 10

<_start>

8048080: mov eax,[80490ac]

8048085: mov edi,0x1

<again>

804808a: mov ecx,[edi*4+80490ac]

8048091: cmp eax,ecx

8048093: jge 8048097

8048095: mov eax,ecx

<skip>

8048097: add edi,0x1

804809a: cmp edi,0xa

804809d: jne 804808a

<_end>

804809f:

<list>

80490ac: 0000000b ffffffff 0000001b 00000023

80490bc: ffffffffbd fffffffe4 00000007 ffffffff

80490cc: 0000002b ffffffff

επόμενη
εντολή

0804809f

program
counter

C S Z V

flags

0000002b

eax

ffffffff

ecx

0000000a

edi

branch if
Z flag == 0

Ολοκλήρωση επιλογής μεγίστου

- Η διαδικασία έχει ολοκληρωθεί
 - Ο μέγιστος ακέραιος (43 ή 0000002b δεκαεξαδικά) βρίσκεται στον καταχωρητή eax
- Ένα «κανονικό πρόγραμμα» στο σημείο αυτό θα συνέχιζε τη λειτουργία του
 - Π.χ. θα τύπωνε το αποτέλεσμα με την printf()
 - Θα έπρεπε όμως να γραφεί με διαφορετικό τρόπο για να μπορεί να καλέσει συναρτήσεις της βιβλιοθήκης stdlib