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

Η γλώσσα assembly

- Μνημονική αναπαράσταση των εντολών μηχανής
  - Αντί να γράφουμε σειρές από 0 και 1...
  - Κάθε εντολή assembly αντιστοιχεί σε μία συγκεκριμένη εντολή μηχανής
- Συγγραφή προγραμμάτων σε χαμηλό επίπεδο
  - Π.χ. κώδικας αρχικοποίησης του υπολογιστή
- Ο κώδικας assembly είναι διαφορετικός ανά επεξεργαστή
  - Άλλο σετ εντολών, άλλα ονόματα καταχωρητών...
  - Δεν μεταφέρεται το ίδιο πρόγραμμα assembly σε διαφορετικό επεξεργαστή (non-portable)
- Στα παραδείγματά μας: assembly x86 (στα PCs)

Σημαίες κατάστασης (flags)

- Μια ομάδα bits που αναφέρουν την κατάσταση της ΚΜΕ μετά την εκτέλεση μιας εντολής
  - Κάθε εντολή επηρεάζει ορισμένα μόνο flags
- Τα πιο κοινά flags:
  - (Z)ero flag = μηδενικό αποτέλεσμα (της προηγούμενης πράξης)
  - (S)ign flag = αρνητικό αποτέλεσμα
  - (C)arry flag = ύπαρξη τελικού κρατουμένου
  - o(V)erflow flag = ένδειξη υπερχείλισης
- Άλλα flags τίθενται από το πρόγραμμα για να ειδοποιήσουν την ΚΜΕ για μια επιλογή
  - Π.χ. το (I)nterrupt flag δηλώνει αν επιτρέπουμε διακοπές ή όχι

Διακλάδωση υπό συνθήκη και flags

- Κάθε εντολή διακλάδωσης υπό συνθήκη εξετάζει ορισμένα flags για να αποφασίσει αν θα εκτελεστεί η διακλάδωση ή όχι
- Τα flags έχουν τεθεί από την αμέσως προηγούμενη εντολή
- Παράδειγμα:

```
cmp edi,10          ; σύγκριση του περιεχομένου του
                    ; καταχωρητή edi με το 10
; η εντολή cmp θέτει ανάλογα τα Z, C, S και V flags
jne again           ;  διακλάδωση εάν Z flag = 0
```

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

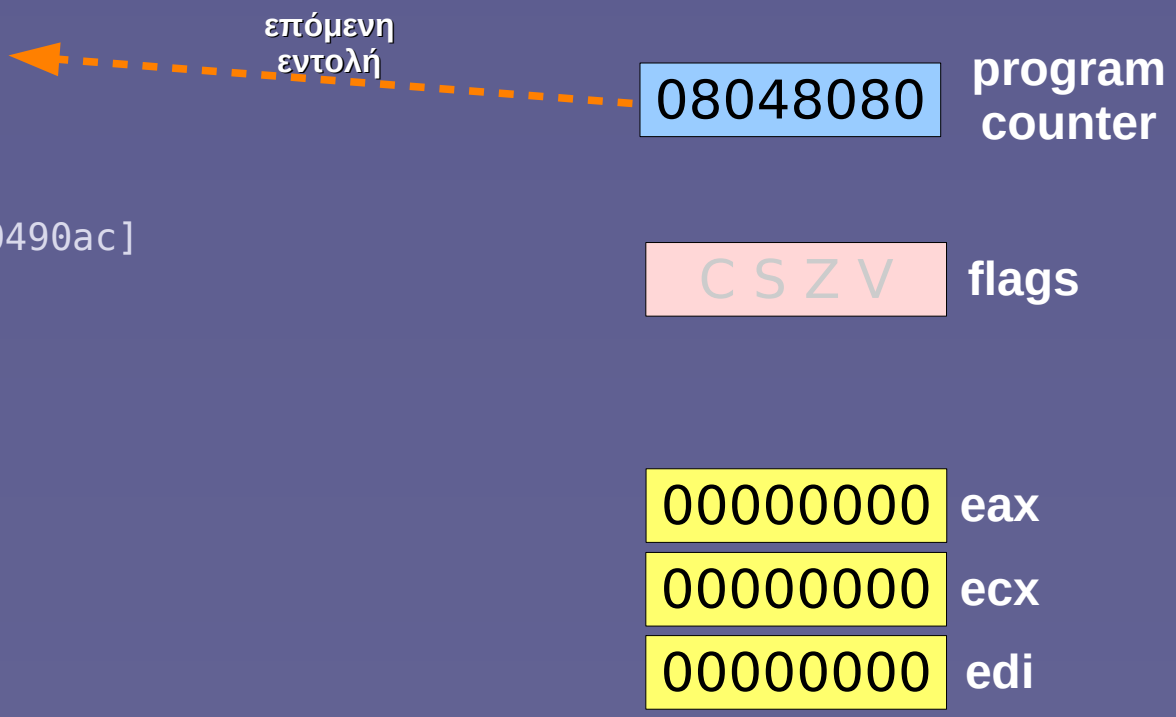
Τα δεδομένα (η λίστα 10 ακεραίων)

```
<list>
 0x80490ac:      0x0000000b      ; list[0] =  11
 0x80490b0:      0xfffffffd      ; list[1] =  -3
 0x80490b4:      0x0000001b      ; list[2] =  27
 0x80490b8:      0x00000023      ; list[3] =  35
 0x80490bc:      0xffffffbd      ; list[4] = -67
 0x80490c0:      0xffffffe4      ; list[5] = -28
 0x80490c4:      0x00000007      ; list[6] =   7
 0x80490c8:      0xfffffffd      ; list[7] =  -3
 0x80490cc:      0x0000002b      ; list[8] =  43
 0x80490d0:      0xffffffff      ; list[9] =  -1
```

- Δέκα C integers (4 bytes ο κάθε αριθμός)
- Σε πίνακα (συνεχόμενες θέσεις μνήμης)

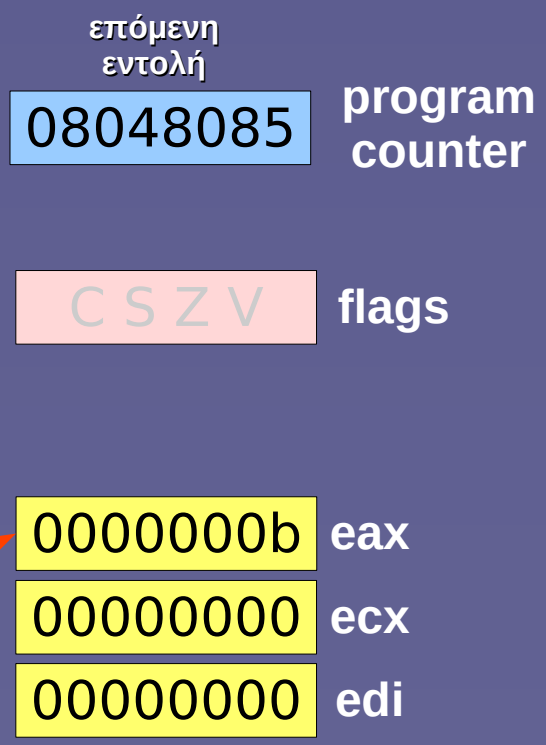
Η εκτέλεση βρίσκεται στο _start

```
<_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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```



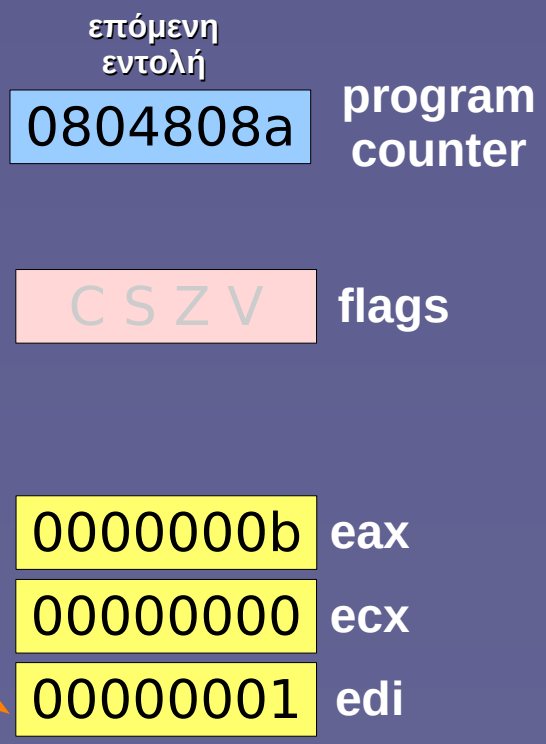
max = list[0]

```
<_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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  ffffffd
 80490cc: 0000002b  ffffffff
```



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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```



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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

1*4+80490ac = 80490b0

επόμενη εντολή

08048091 **program counter**

C S Z V **flags**

0000000b **eax**

fffffffd **ecx**

00000001 **edi**

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

0000000b>fffffffd
(11>-3)

επόμενη εντολή

08048093 **program counter**

C S Z V **flags**

0000000b **eax**

fffffffd **ecx**

00000001 **edi**

branch if 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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

branch if
S flag == V flag

επόμενη εντολή

08048097 **program counter**

C S Z V **flags**

0000000b **eax**

fffffffd **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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

2<10

επόμενη εντολή

0804809d **program counter**

C S Z V **flags**

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

branch if
Z flag == 0

επόμενη εντολή

0804808a — **program counter**

C S Z V — **flags**

0000000b — **eax**

fffffffd — **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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

2*4+80490ac = 80490b4

επόμενη εντολή

08048091 **program counter**

C S Z V **flags**

0000000b **eax**

0000001b **ecx**

00000002 **edi**

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

0000000b<0000001b
(11<27)

επόμενη εντολή

08048093 **program counter**

C S Z V **flags**

0000000b **eax**

0000001b **ecx**

00000002 **edi**

branch if 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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

branch if
S flag == V flag

επόμενη εντολή

08048095 **program counter**

C S Z V **flags**

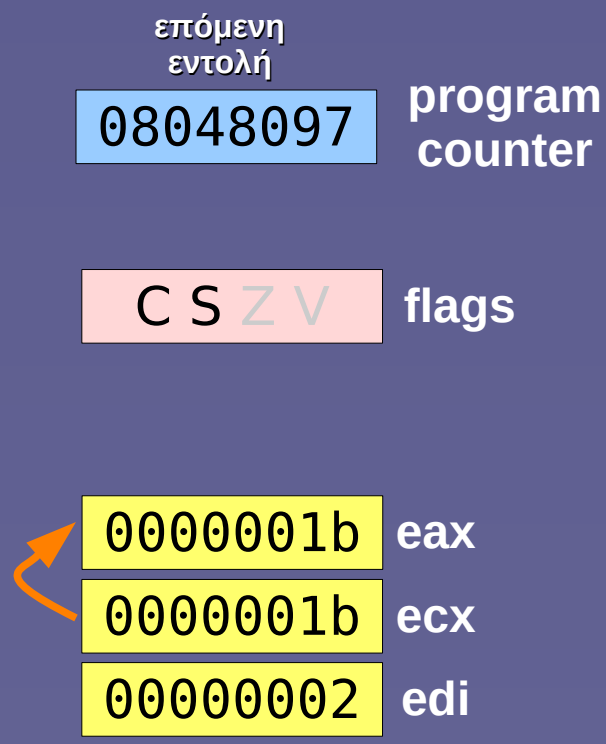
0000000b **eax**

0000001b **ecx**

00000002 **edi**

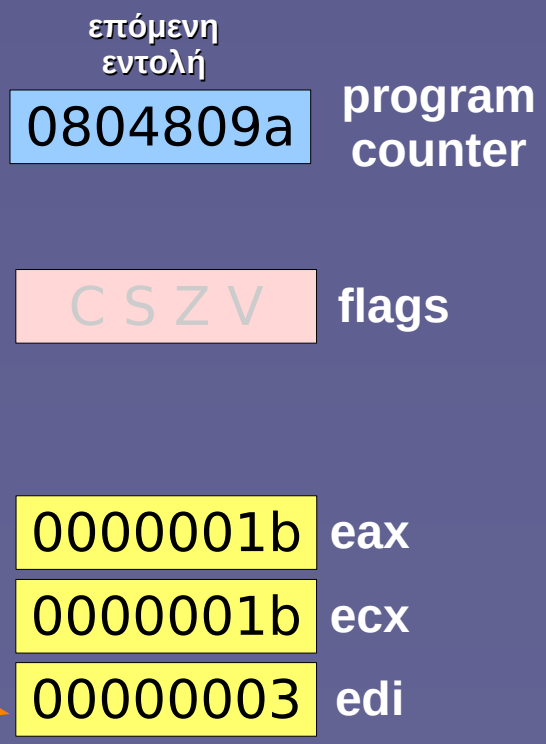
max = new

```
<_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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```



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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```



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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

3<10

επόμενη εντολή

0804809d **program counter**

C S Z V **flags**

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

branch if
Z flag == 0

επόμενη εντολή

0804808a **program counter**

C S Z V **flags**

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

3*4+80490ac = 80490b8

επόμενη εντολή

08048091 **program counter**

C S Z V **flags**

0000001b **eax**

00000023 **ecx**

00000003 **edi**

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

0000001b<00000023
(27<35)

επόμενη εντολή

08048093 **program counter**

C S Z V **flags**

0000001b **eax**

00000023 **ecx**

00000003 **edi**

branch if 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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

branch if
S flag == V flag

επόμενη εντολή

08048095 **program counter**

C S Z V **flags**

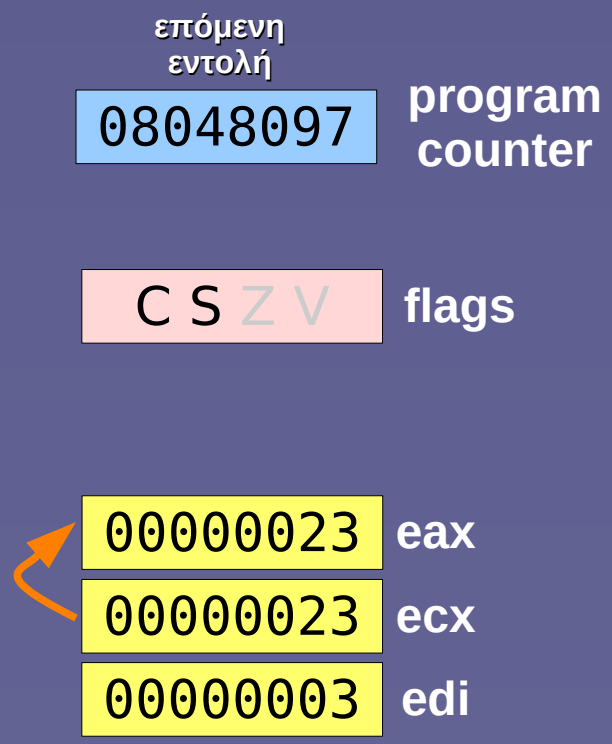
0000001b **eax**

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```



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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

4<10

επόμενη εντολή

0804809d **program counter**

C S Z V **flags**

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

branch if
Z flag == 0

επόμενη εντολή

0804808a **program counter**

C S Z V **flags**

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, έστω ότι έχει τελειώσει και η σύγκριση με το list[9] στην επόμενη διαφάνεια

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

10==10

επόμενη εντολή

0804809d **program counter**

C S Z V **flags**

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  fffffffd  0000001b  00000023
 80490bc: ffffffbd  ffffffe4  00000007  fffffffd
 80490cc: 0000002b  ffffffff
```

branch if
Z flag == 0

επόμενη εντολή

0804809f — program counter

C S Z V — flags

0000002b — eax

ffffffff — ecx

0000000a — edi

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