Αρχιτεκτονική Υπολογιστών

Ενότητα 10: Πέρασμα Παραμέτρων σε Διαδικασίες.

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org
Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών
http://arch.ece.uowm.gr/mdasyg

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Είδη συναρτήσεων

- Υπάρχουν 2 είδη συναρτήσεων ως προς το πέρασμα παραμέτρων από το καλών πρόγραμμα προς το υποπρόγραμμα:
  - Συναρτήσεις που δε δέχονται παραμέτρους.
  - Συναρτήσεις που δέχονται παραμέτρους.

Χωρίς παραμέτρους

- Οι συναρτήσεις αυτές κάνουν πάντα μια συγκεκριμένη λειτουργία.
- Δε μπορεί να τροποποιηθεί η λειτουργία τους.
- Δε χρειάζονται παραμέτρους.

Παράδειγμα

Η παρακάτω συνάρτηση πάντα εκτυπώνει ένα μήνυμα:

```
printmsg proc
push dx
mov ah,09
lea dx,printprompt
int 21h
pop dx
ret
printmsg endp
```

Με παραμέτρους

- Τις περισσότερες φορές υπάρχει ανάγκη για παραμετροποίηση της λειτουργίας μιας συνάρτησης.
- Θα πρέπει να γίνει η κλήση της συνάρτησης με συγκεκριμένες παραμέτρους.
- Θα πρέπει λοιπόν ΠΡΙΝ την κλήση της συνάρτησης να τοποθετήσουμε τους παραμέτρους εκεί που τους περιμένει να τους βρει.
- Η συνάρτηση αναλόγως της λειτουργίας της μπορεί να επιστρέφει κάποια ή κάποιες τιμές.
- Υπάρχουν συναρτήσεις που δέχονται παραμέτρους, αλλά δεν επιστρέφουν τίποτα.

Κατηγοριοποίηση ως προς την επικοινωνία

Η επικοινωνία με το καλών πρόγραμμα γίνεται με 3 τρόπους

Η επικοινωνία της συνάρτησης με το καλών πρόγραμμα μπορεί να γίνει με τους παρακάτω 3 τρόπους:

- Μέσω συγκεκριμένων καταχωρητών.
- Μέσω συγκεκριμένων διευθύνσεων μνήμης (δηλωμένες θέσεις μνήμης στο τμήμα δεδομένων).
- Μέσω του σωρού.

Ποια είναι η καλύτερη επικοινωνία;

- Ο καλύτερος τρόπος επικοινωνίας (και πιο δύσκολος) είναι μέσω του σωρού. Δηλαδή, το καλών πρόγραμμα βάζει τις παραμέτρους στο σωρό, η συνάρτηση της διαβάζει και τις χρησιμοποιεί, και τοποθετεί την τιμή επιστροφής πάλι στο σωρό.
- Αν χρησιμοποιήσουμε τους άλλους τρόπους, τότε έχουμε πρόβλημα στο αν η συνάρτηση αυτή γίνει κλήση ενώ ήδη εκτελείται (αναδρομική κλήση). Αυτό έχει ως συνέπεια να σβηστούν οι αρχικές τιμές.
- Με τη χρήση με το σωρό λύνονται όλα τα προβλήματα αναδρομικότητας (άλλα είναι δύσκολη η χρήση του).

Επικοινωνία μέσω συγκεκριμένων καταχωρητών

Επικοινωνία μέσω συγκεκριμένων καταχωρητών

- Οι παράμετροι της διαδικασίας τοποθετούνται σε συγκεκριμένους καταχωρητές που εμείς επιλέγουμε.
- Η διαδικασία επιστρέφει μια ή περισσότερες τιμές σε καταχωρητές που εμείς επιλέγουμε.
- Εύκολη υλοποίηση, αλλά αν γίνει 2η κλήση ενώ ήδη εκτελείται τότε χάνονται οι τιμές της 1ης κλήσης.
- Προσοχή στα PUSH/POP. Δε θα είναι 100% διαφανής συνάρτηση γιατί η τιμή επιστρέφεται μέσω καταχωρητή (δηλαδή, τουλάχιστον ένας καταχωρητής θα πρέπει να μην επανέλθει στην αρχική του τιμή).
- Ακολουθεί παράδειγμα....

Επικοινωνία μέσω συγκεκριμένων καταχωρητών: Παράδειγμα (1/3)

Υλοποιήστε τη συνάρτηση max(a,b) (μέγιστο δύο τιμών) με την τεχνική επικοινωνίας μέσω καταχωρητών.

- Επιλέγουμε *(αυθαίρετα)* ότι η 1η παράμετρος της συνάρτησης θα τοποθετείται στον καταχωρητή CH.
- Επιλέγουμε *(αυθαίρετα)* ότι η 2η παράμετρος της συνάρτησης θα τοποθετείται στον καταχωρητή CL.
- Επιλέγουμε *(αυθαίρετα)* ότι η τιμή επιστροφής συνάρτησης θα τοποθετείται στον καταχωρητή DH.
- Θα μπορούσαμε να επιλέξουμε οποιονδήποτε καταχωρητή.

Επικοινωνία μέσω συγκεκριμένων καταχωρητών: Παράδειγμα (2/3)

```
max_registers proc
;1st   parameter CH
;2nd parameter CL
;return: DH
cmp CH,CL
JB ch_lower
;edw to ch einai bigger
mov dh,ch
jmp _max_registers_end
      ch_lower:
      ;edw to cl einai bigger
      mov dh,cl
_max_registers_end:
RET
max_registers endp
```

Επικοινωνία μέσω συγκεκριμένων καταχωρητών: Παράδειγμα (3/3)

Κλήση της συνάρτησης.

- Στο κυρίως πρόγραμμα πριν την κλήση, θα πρέπει να τοποθετήσουμε τις παραμέτρους τις συνάρτησης στους καταχωρητές CH και CL.
- Παράδειγμα:

```
mov CH, πρώτη_παράμετρος
mov CL, δεύτερη παράμετρος
call max_registers
;το αποτέλεσμα βρίσκεται στον  DH
```

Επικοινωνία μέσω συγκεκριμένων διευθύνσεων μνήμης

Επικοινωνία μέσω συγκεκριμένων δ/νσεων μνήμης

- Οι παράμετροι της διαδικασίας τοποθετούνται σε συγκεκριμένες διευθύνσεις μνήμης.
- Έχουμε ορίσει στο τμήμα δεδομένων τις θέσεις μνήμης.
- Η διαδικασία επιστρέφει μια ή περισσότερες τιμές σε διευθύνσεις μνήμης που εμείς επιλέγουμε.
- Εύκολη υλοποίηση, άλλα δύσκολη μεταφέρσιμη (portable), αλλά αν γίνει 2η κλήση ενώ ήδη εκτελείται τότε χάνονται οι τιμές της 1ης κλήσης.
- Μπορεί να είναι 100% διαφανής. Πρέπει να χρησιμοποιήσουμε push/pop για κάθε καταχωρητή που χρησιμοποιείται.
- Ακολουθεί παράδειγμα....

Παραδείγματα (1/4)

Υλοποιήστε τη συνάρτηση max(a,b) *(μέγιστο δύο τιμών)* με την τεχνική επικοινωνίας μέσω διευθύνσεων μνήμης.

- Επιλέγουμε *(αυθαίρετα)* ότι η 1η παράμετρος της συνάρτησης θα τοποθετείται στη θέση μνήμης: first_parameter
- Επιλέγουμε (*αυθαίρετα*) ότι η 2η παράμετρος της συνάρτησης θα τοποθετείται στη θέση μνήμης: second_parameter
- Επιλέγουμε (*αυθαίρετα*) ότι η τιμή επιστροφής συνάρτησης θα τοποθετείται στη θέση μνήμης:

  max_result.

Παραδείγματα (2/4)

- Θα πρέπει να δηλώσουμε στο τμήμα δεδομένων τις παρακάτω διευθύνσεις μνήμης.

```
DATA SEGMENT
    first_parameter db    0
    second_parameter db   0
    max_result db   0
DATA ENDS
```

Παραδείγματα (3/4)

```
max_memory proc
;1st first_parameter
;2nd second_parameter
;return: max_result
PUSH CX ;για να είναι διαφανής
MOV CH,first_parameter
MOV CL,second_parameter
cmp CH,CL
JB ch_lower
;edw to ch einai bigger
mov max_result,ch
jmp _max_memory_end
        ch_lower:
        ;edw to cl einai bigger
        mov max_result,cl
_max_memory_end:
RET
POP CX
max_memory endp
```

Παραδείγματα (4/4)

Κλήση της συνάρτησης

- Για να γίνει η κλήση της συνάρτησης θα πρέπει πριν την κλήση να τοποθετήσουμε στις συγκεκριμένες διευθύνσεις μνήμης τις παραμέτρους.
- Παράδειγμα:

```
mov first_parameter, πρώτη_παράμετρος
mov second_parameter, δεύτερη παράμετρος
call max_registers
;το αποτέλεσμα βρίσκεται στο max_result
```

Επικοινωνία μέσω του σωρού

Επικοινωνία μέσω σωρού

- Η πιο σωστή τεχνική.
- Οι συναρτήσεις μπορούν να είναι 100% διαφανείς.
- Η επικοινωνία γίνεται μέσω της μνήμης του σωρού.
- Τοποθετούμε τις παραμέτρους στο σωρό με τις εντολές PUSH.
- Για να διαβάσουμε τις παραμέτρους ΔΕ χρησιμοποιούμε τις εντολές POP γιατί μέσα στο σωρό υπάρχει και η διεύθυνση επιστροφής της συνάρτησης. Χρησιμοποιούμε μια τεχνικής άμεσης πρόσβασης στο σωρό.

Ο σωρός: μια επανάληψη (1/4)

- Ο σωρός στη x86 ISA είναι μια δομή που διευθυνσιοδοτείται από το δείκτη SP.
- Ο δείκτης αυτός κάθε φορά δείχνει στο τελευταίο στοιχείο που έχει τοποθετηθεί στο σωρό.
- Ο σωρός ορίζεται στο τμήμα σωρού:

```
soros segment stack
    db   256  dup(0)
soros ends
```

Ο σωρός: μια επανάληψη (2/4)

- Η παραπάνω δήλωση, δημιουργεί ένα τμήμα μνήμης μεγέθους 256Byte ή 0100h Byte.
- Ο δείκτης SP παίρνει τη μεγαλύτερη τιμή (=μεγαλύτερη μετατόπιση στο σωρό). Στο παραπάνω παράδειγμα ο δείκτης SP θα έχει την τιμή 0100h.
- Όταν γίνεται μια εντολή PUSH τότε τοποθετείται ένα στοιχείο 2 Byte στη θέση μνήμης που δείχνει το SP και το SP μειώνεται κατά 2 (αφού τοποθετήθηκαν 2 Byte). Για παράδειγμα αν SP=0100h και γίνει PUSH AX, το SP=00FEh.

Ο σωρός: μια επανάληψη (3/4)

- Όταν γίνεται μια εντολή POP τότε αφαιρείται ένα στοιχείο 2 Byte στη θέση μνήμης που δείχνει το SP και το SP αυξάνεται κατά 2 (αφού αφαιρέθηκαν 2 Byte). Για παράδειγμα αν SP=0010h και γίνει POP AX, το SP=0012h.
- Όταν γίνει μια κλήση συνάρτησης CALL μέσα στο ίδιο τμήμα (το πιο συνηθισμένο και αυτό που κάνουμε σε όλες τις ασκήσεις) τότε τοποθετείται στο σωρό η δ/νση επιστροφής και είναι ακριβώς όμοιο με το PUSH.

Ο σωρός: μια επανάληψη (4/4)

- Όταν γίνει μια επιστροφή συνάρτησης RET μέσα στο ίδιο τμήμα (το πιο συνηθισμένο και αυτό που κάνουμε σε όλες τις ασκήσεις) τότε απομακρύνεται από το σωρό η δ/νση επιστροφής και είναι ακριβώς όμοιο με το POP.
- Αν θέλουμε να διαβάσουμε κάποιο στοιχείο από το σωρό θα πρέπει να χρησιμοποιήσουμε απευθείας διευθυνσιοδότηση μνήμης με το δείκτη SP.

Πρόσβαση στο σωρό χωρίς εντολή POP

- Επειδή το SP δε μπορεί να χρησιμοποιηθεί στην απευθείας διευθυνσιοδότηση, θα πρέπει να τοποθετηθεί στο BP.
- Στη συνέχεια μπορούμε να χρησιμοποιήσουμε απευθείας διευθυνσιοδότηση με το BP για να διαβάσουμε στοιχεία από το σωρό.
- Υπενθυμίζουμε ότι αν προστίθεται κάτι στο BP τότε διαβάζουμε ένα στοιχείο πιο κοντά στην αρχή (δηλαδή την κορυφή) του σωρού.
- Υπενθυμίζουμε ότι αν αφαιρείται κάτι από το BP τότε διαβάζουμε ένα στοιχείο πιο κοντά στο τέλος του σωρού (δηλαδή μακρυά από την κορυφή) του σωρού.
- Προσοχή: Χρησιμοποιούμε πάντα μετακίνηση σε καταχωρητή 2Byte, γιατί στο σωρό πάντα τοποθετείται κάτι σε μεγέθη των 2Byte

Πρόσβαση στο σωρό χωρίς POP: Παραδείγματα

- `mov bp,sp` **;save SP pointer for direct addressing**

- Μετακίνηση της τελευταίας τιμής του σωρού:

  `mov ax,[bp]`

- Μετακίνηση της τιμής που βρίσκεται πριν τη δ/νση επιστροφής:

  `mov ax,[bp+2]`

- Μετακίνηση της τιμής που βρίσκεται μετά την τρέχουσα θέση:

  `mov ax,[bp-2]`

Επικοινωνία μέσω σωρού: Παράδειγμα (1/5)

Υλοποιήστε τη συνάρτηση max(a,b) (μέγιστο δύο τιμών) με την τεχνική επικοινωνίας μέσω σωρού.

- Αυτή η τεχνική σημαίνει ότι κάνουμε PUSH όλες τις παραμέτρους πριν την κλήση της συνάρτησης.
- Χρησιμοποιούμε την άμεση διευθυνσιοδότηση χρησιμοποιώντας το SP/BP.
- Θα αναλύσουμε ένα παράδειγμα για να δούμε πως λειτουργεί.
- Έστω αρχικά το SP έχει τιμή 0100h.

Επικοινωνία μέσω σωρού: Παράδειγμα (2/5)

- Κάνουμε PUSH τις τιμές μας στο σωρό. Έστω, οι τιμές μας βρίσκονται στους καταχωρητές AX και BX.

```
PUSH AX              ;το SP γίνεται 0100h-2 = 00FEh
PUSH BX              ;το SP γίνεται 00FEh-2 = 00FCh
CALL max_stack       ;το SP γίνεται 00FCh-2 = 00FAh
```

Επικοινωνία μέσω σωρού: Παράδειγμα (3/5)

```
max_stack       proc
;function to compute MAX
;1st 2nd parameter at stack
;Αρχικά εντολές PUSH για κάθε καταχωρητή που
;τροποποιούμε μέσα στη συνάρτηση, για να τη
;κάνουμε διαφανή
PUSH BP         ; to be used for stack direct addressing
                ; SP γίνεται  00FAh – 2 = 00F8h
PUSH CX         ; to be used for computation
                ; SP γίνεται  00F8h – 2 = 00F6h
PUSH DX         ; to be used for computation
                ; SP γίνεται  00F6h – 2 = 00F4h
```

Επικοινωνία μέσω σωρού: Παράδειγμα (4/5)

```
mov bp,sp              ;save SP pointer for direct addressing
mov cx,[bp+8] ;η 2 παράμετρος είναι 8 byte πίσω
mov dx,[bp+10] ;η 1 παράμετρος είναι 10 byte πίσω
cmp CX,DX
JB CX_lower
;εδώ το CX είναι μεγαλύτερο
mov DX,CX
CX_lower:
;εδώ η μεγαλύτερη τιμή είναι στο DX
mov [bp-2],DX ;αποθήκευσε τη max τιμή στο σωρό
POP DX         ;επαναφορά των τιμών των καταχωρητών
POP CX
POP BP
RET
max_stack endp
```

Επικοινωνία μέσω σωρού: Παράδειγμα (5/5)

- Η τιμή επιστροφής αποθηκεύεται στο σωρό.
- Προκειμένου να τη διαβάσουμε θα κάνουμε τα εξής στο κυρίως πρόγραμμα.

```
call max_stack
mov bp,sp
mov ax,[bp-10]
```

- Βρίσκεται 10 Byte μείον από την τρέχουσα τιμή του SP. Γιατί;
  - Μέσα στη συνάρτηση χρησιμοποιούνται 6 Byte από τα 3 PUSH.
  - Η συνάρτηση χρησιμοποιεί 2 Byte για την IP.
  - Η συνάρτηση τοποθετεί την τιμή 2 Byte παρακάτω από την τρέχουσα τιμή (εντολή: `mov [bp-2],dx`).

Κατηγοριοποίηση ως προς το είδος των παραμέτρων

Κατηγοριοποίηση: το είδος των παραμέτρων (1/2)

Υπάρχουν 2 κατηγορίες ως προς το είδος των παραμέτρων.

- Πέρασμα των πραγματικών τιμών *(τεχνική pass-by-value).*
- Στέλνουμε τις πραγματικές τιμές στις διαδικασίες. Πιο εύκολο, αλλά δε μπορούμε να στείλουμε πολλαπλές παραμέτρους.
- Πέρασμα των διευθύνσεων μνήμης των τιμών *(τεχνική pass-by-reference).*
- Στέλνουμε τις διευθύνσεις μνήμης των παραμέτρων και όχι τις ίδιες τις παραμέτρους. Πιο δύσκολο στη χρήση, αλλά μπορούμε να στείλουμε πολλαπλές παραμέτρους.

Κατηγοριοποίηση: το είδος των παραμέτρων (2/2)

- Οι 2 τεχνικές είναι ορθογώνιες ως προς την επικοινωνία με το καλών πρόγραμμα.
- Δηλαδή, επιτρέπονται όλοι οι συνδυασμοί των 2 κατηγοριών ως προς το είδος των παραμέτρων και των 3 κατηγοριών ως προς την επικοινωνία με το καλών πρόγραμμα *(συνολικά 6 συνδυασμοί).*
- Για παράδειγμα, μπορούμε να έχουμε:
  - "επικοινωνία μέσω συγκεκριμένων καταχωρητών με πέρασμα των πραγματικών τιμών".
    (δηλαδή στους καταχωρητές τοποθετούμε τις τιμές).
  - "επικοινωνία μέσω συγκεκριμένων καταχωρητών με πέρασμα των διευθύνσεων των τιμών".
    (δηλαδή στους καταχωρητές τοποθετούμε τις διευθύνσεις μνήμης).

Πέρασμα των πραγματικών τιμών *(τεχνική pass-by-value):* Παράδειγμα

- Όλα τα παραδείγματα που παρουσίαζαν τις διάφορες τεχνικές επικοινωνίας χρησιμοποιούσαν την τεχνική αυτή.

Πέρασμα των διευθύνσεων μνήμης των τιμών: Παράδειγμα (1/4)

- Θα παρουσιάσουμε την τεχνική σε συνδυασμό με την επιλογή της επικοινωνίας μέσω καταχωρητή.
- Για να βρούμε τη διεύθυνση μιας μεταβλητής θα χρησιμοποιήσουμε την εντολή lea *(load effective address).*
- Παράδειγμα:

  Να υλοποιηθεί συνάρτηση που θα υπολογίζει το μέγιστο όρο μιας σειράς 10 αριθμών, η οποία θα υλοποιηθεί με το πέρασμα διευθύνσεων μνήμης των τιμών μέσω καταχωρητή.

Πέρασμα των διευθύνσεων μνήμης των τιμών: Παράδειγμα (2/4)

- Υποθέτουμε ότι έχουμε μια μεταβλητή στο τμήμα δεδομένων:

```
array1      db
10,123,43,1,23,4,5,65,10,1
```

- Επιλέγουμε *(αυθαίρετα)* τον καταχωρητή AX στον οποίο τοποθετούμε τη διεύθυνση του πρώτου στοιχείου, πριν την κλήση του προγράμματος ως εξής:

```
lea AX,array1
call compute_max
```

Πέρασμα των διευθύνσεων μνήμης των τιμών: Παράδειγμα (3/4)

```
compute_maxproc
PUSH BP
PUSH CX
mov bp,ax                ;για χρήση άμεσης διευθυνσιοδότησης
mov cx,9                 ;για χρήση μετρητή
mov dl,ds:[bp]   ;για χρήση του μέγιστου. Πρώτο στοιχείο

compute_loop:
inc bp           ;κάθε φορά αυξάνουμε το δείκτη μνήμης κατά 1
cmp dl,ds:[bp]
JA next_iteration
mov dl,ds:[bp]   ;νέος μέγιστος αριθμός
next_iteration:
Loop compute_loop

POP CX
POP BP
RET              ;η μέγιστη τιμή επιστρέφεται στον DX
compute_max endp
```

Πέρασμα των διευθύνσεων μνήμης των τιμών: Παράδειγμα (4/4)

Παρατηρήστε ότι:

- Χρησιμοποιήσαμε ως δείκτη μνήμης το **BP** το οποίο επειδή ως προεπιλογή συνδέεται με το σωρό, τοποθετούμε το **ds:[..]** για να δείξουμε ότι θέλουμε πρόσβαση στο τμήμα δεδομένων. Εναλλακτικά, θα μπορούσαμε να χρησιμοποιήσουμε τους δείκτες καταχωρητές di, si.
- Στην εντολή **mov dl,[bp]** μεταφέρουμε 1 Byte επειδή στο τμήμα δεδομένων έχουμε ορίσει τον πίνακα ως array1 db. Αν είχαμε γράψει την εντολή **mov dx,[bp]** τότε θα μεταφέραμε κάθε φορά 2 byte, που είναι λάθος στη συγκεκριμένη περίπτωση.
- Δεν έχουμε push/pop για το **DX** επειδή αποφασίσαμε να τοποθετήσουμε τη μέγιστη τιμή σε αυτόν τον καταχωρητή.

Εντολές PUSHA/POPA για επίτευξη διαφάνειας

- Προκειμένου να κάνουμε διαφανείς τις συναρτήσεις, πρέπει να διατηρούμε τις τιμές των καταχωρητών με εντολές PUSH και να τις επαναφέρουμε με εντολές POP.
- Αν χρησιμοποιούμε πολλαπλούς καταχωρητές τότε τοποθετούμε πολλαπλές εντολές PUSH στην αρχή της συνάρτησης και ισάριθμες εντολές POP με αντίστροφη σειρά πριν την εντολή RET.
- Η όλη διαδικασία έχει κάποια δυσκολία και απαιτεί ιδιαίτερη προσοχή.
- Για να διευκολύνει τους προγραμματιστές η Intel υλοποίησε τις εντολές PUSHA/POPA.

Εντολές PUSHA/POPA: Χρήση (1/2)

- Οι εντολές PUSHA/POPA χρησιμοποιήθηκαν για πρώτη φορά στο 80186, αλλά μπορούν να χρησιμοποιηθούν για ευκολία και στο emu8086 *(ο οποίος προσομοιώνει παλαιότερο υπολογιστή).*
- Η εντολή PUSHA ωθεί όλους τους 8 βασικούς καταχωρητές στο σωρό, δηλαδή τους: ax, cx, dx, bx, sp, bp, si, di.
- Η εντολή POPA εξάγει όλους τους 8 βασικούς καταχωρητές από το σωρό με την αντίστροφη σειρά.
- Έτσι, ο προγραμματιστής τοποθετεί μια εντολή PUSHA αμέσως μετά το PROC και μια εντολή POPA πριν το RET και η συνάρτηση έχει γίνει διαφανής.

Εντολές PUSHA/POPA: Χρήση (2/2)

- Οι εντολές PUSHA/POPA έχουν ως συνέπεια να διαβαστούν 8 καταχωρητές και να τοποθετηθούν στο σωρό, δηλαδή στην εξωτερική μνήμη.
- Αυτό σημαίνει ότι απαιτούνται 8 προσβάσεις στην εξωτερική μνήμη, κάτι που καθυστερεί το σύστημα.
- Οι εντολές PUSHA/POPA πρέπει να αποφεύγονται αν τροποποιείται μέσα στη συνάρτηση ελάχιστος αριθμός καταχωρητών, γιατί το ίδιο αποτέλεσμα γίνεται βελτιστοποιημένο με την εντολή PUSH.
- Για παράδειγμα, αν τροποποιείται μόνο το AX τότε δε θα κάνουμε PUSHA *(8 προσβάσεις στην εξωτερική μνήμη)*, αλλά PUSH AX *(1 πρόσβαση μόνο στην εξωτερική μνήμη).*

Τέλος Ενότητας

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης