

Ακολουθιακή εκτέλεση εντολών και διακλαδώσεις

Μια Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) εκτελεί εντολές **ακολουθιακά**:

- Οι εντολές μηχανής εκτελούνται **η μία μετά την άλλη**, δηλαδή μετά την **τρέχουσα** εντολή εκτελείται η **επόμενη** (στην επόμενη θέση μνήμης).
- Η ΚΜΕ διαθέτει έναν **καταχωρητή ειδικού σκοπού**, τον μετρητή προγράμματος (**program counter** ή **instruction pointer**), ο οποίος **δείχνει στη θέση μνήμης** όπου βρίσκεται η **επόμενη προς εκτέλεση εντολή μηχανής**.
- Η ΚΜΕ διαθέτει λογικά κυκλώματα (πύλες) που, αμέσως μετά τη λήψη της νέας εντολής, **αυξάνουν αυτόματα** τον program counter (PC) έτσι ώστε να δείχνει στην επόμενη εντολή.

Τι σημαίνει «ο program counter δείχνει στη θέση μνήμης»;

Υποθέστε ότι η επόμενη προς εκτέλεση εντολή μηχανής βρίσκεται αποθηκευμένη στη θέση μνήμης με διεύθυνση 1000. Ο program counter θα περιέχει την τιμή 1000, θα είναι δηλαδή ένας «δείκτης» προς τη θέση μνήμης αυτή.

Παράδειγμα:

Έστω ότι έχουμε μια σειρά εντολών μηχανής (*εντολή1*, *εντολή2*, κλπ) αποθηκευμένες στις θέσεις μνήμης 1000 και κάτω. Υποθέστε ότι **κάθε εντολή καταλαμβάνει 4 bytes** και ότι ο program counter (PC) έχει την τιμή 1000.

1000	<i>εντολή1</i>	<i>εντολή1</i>	<i>εντολή1</i>	<i>εντολή1</i>
1004	<i>εντολή2</i>	<i>εντολή2</i>	<i>εντολή2</i>	<i>εντολή2</i>
1008	<i>εντολή3</i>	<i>εντολή3</i>	<i>εντολή3</i>	<i>εντολή3</i>
1012	<i>εντολή4</i>	<i>εντολή4</i>	<i>εντολή4</i>	<i>εντολή4</i>

Βήμα 1: PC=1000 και στην ΚΜΕ εισάγεται η *εντολή1* που βρίσκεται στη μνήμη στις διευθύνσεις 1000, 1001, 1002 και 1003. Ο καταχωρητής PC αυξάνεται κατά 4 και τώρα περιέχει το 1004.

Βήμα 2: PC=1004 και στην ΚΜΕ εισάγεται η *εντολή2* που βρίσκεται στη μνήμη στις διευθύνσεις 1004, 1005, 1006 και 1007. Ο καταχωρητής PC αυξάνεται κατά 4 και τώρα περιέχει το 1008.

Βήμα 3: Η διαδικασία επαναλαμβάνεται με τον ίδιο τρόπο.

Βήμα 4: κ.ο.κ

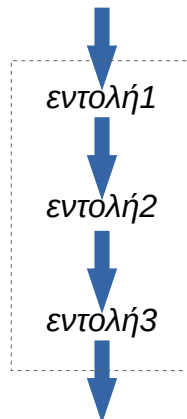
...

Κάθε εντολή μηχανής καταλαμβάνει πάντα 4 bytes;

Αυτό εξαρτάται από την αρχιτεκτονική σχεδίαση του κάθε επεξεργαστή. Υπάρχουν κάποιες αρχιτεκτονικές που οι εντολές τους έχουν σταθερό μήκος (όπως στο παράδειγμα), καθώς και άλλες όπου οι εντολές έχουν μεταβλητό μήκος, διαφορετικό ανά είδος εντολής. **Σε κάθε περίπτωση, τα κυκλώματα της ΚΜΕ αυξάνουν αυτόματα τον PC έτσι ώστε πάντα να δείχνει στην αρχή της επόμενης εντολής μηχανής.**

Διακλαδώσεις

Σύμφωνα με τα προηγούμενα, δεν υπάρχει τρόπος να αλλάξουμε τη ροή του προγράμματος: οι εντολές μηχανής θα εκτελεστούν η μία μετά την άλλη. Εάν έχουμε μια ομάδα εντολών και αρχίσει να εκτελείται η πρώτη εντολή, ξέρουμε ότι σίγουρα η εκτέλεση θα φτάσει στην τελευταία της ομάδας αφού εκτελεστούν και όλες οι ενδιάμεσες.

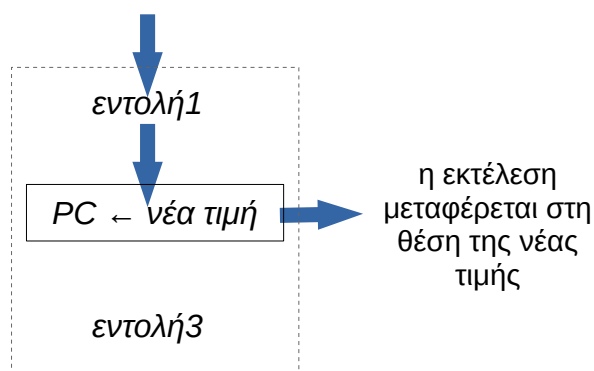


Φυσικά, μόνο με την ακολουθιακή εκτέλεση εντολών δεν μπορούμε να κατασκευάσουμε πολύ χρήσιμα προγράμματα: δεν υπάρχει η δυνατότητα υλοποίησης δομών if...else. Για τον λόγο αυτόν διατίθεται μια ειδική κατηγορία εντολών μηχανής, οι **διακλαδώσεις** (branches ή jumps).

- Οι εντολές διακλάδωσης **επιτρέπουν την αλλαγή του PC σε μια τιμή της επιλογής μας**, διαφορετική από τη διεύθυνση της επόμενης εντολής. Με μια νέα τιμή στον PC, η εκτέλεση του προγράμματος θα συνεχιστεί από τη νέα διεύθυνση μνήμης, όπου κι αν είναι αυτή.

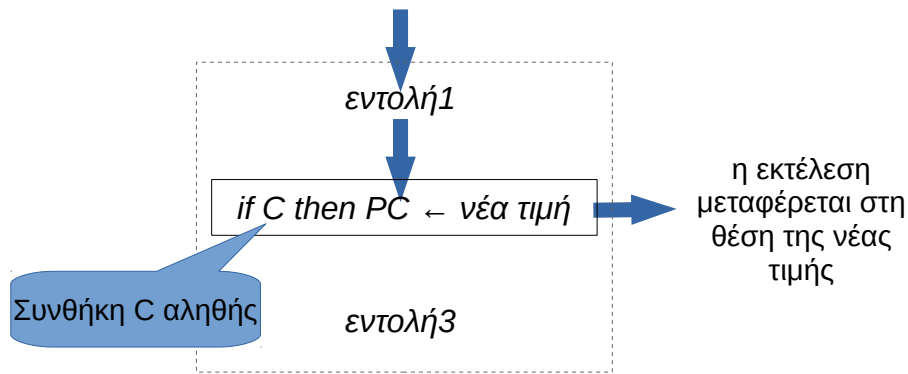
Διακλάδωση χωρίς συνθήκη

Μια κατηγορία εντολών διακλάδωσης αλλάζει **πάντα** την τιμή του PC. Εάν η ροή του προγράμματος φτάσει σε μια τέτοια εντολή, η εκτέλεση στη συνέχεια θα μεταφερθεί στη νέα θέση μνήμης που θα δείχνει πλέον ο PC.

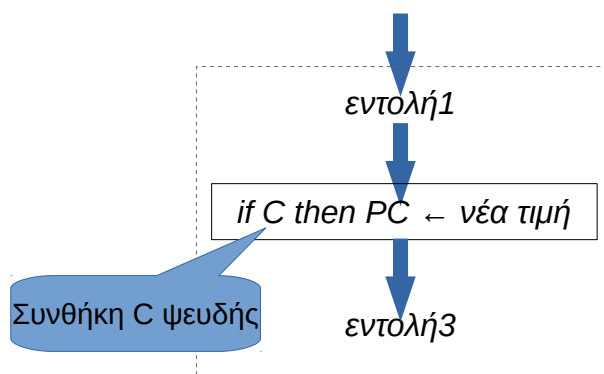


Διακλάδωση υπό συνθήκη

Η δεύτερη κατηγορία εντολών διακλάδωσης αλλάζει την τιμή του PC **μόνο όταν ισχύει κάποια συνθήκη**:



Εάν η συνθήκη **δεν είναι αληθής**, η εκτέλεση συνεχίζεται κανονικά **στην εντολή που ακολουθεί** την εντολή διακλάδωσης:



Τι ακριβώς είναι η συνθήκη;

Βρισκόμαστε σε επίπεδο εντολών μηχανής. Ανάλογα με την αρχιτεκτονική του επεξεργαστή, η συνθήκη είναι μια τιμή που έχει υπολογιστεί από τις προηγούμενες εντολές μηχανής, συνήθως μια σύγκριση των τιμών δύο καταχωρητών.

Δομές επιλογής γλωσσών υψηλότερου επιπέδου

Ακολουθούν παραδείγματα υλοποίησης δομών if..else γλωσσών υψηλού επιπέδου με τη βοήθεια των διακλαδώσεων. Τα παραδείγματα αυτά δεν είναι ο μόνος τρόπος υλοποίησης· συνήθως οι μεταγλωττιστές επιλέγουν τον πιο αποδοτικό τρόπο μετάφρασης (με τις λιγότερες διακλαδώσεις, ή με βελτιστοποίηση της θέσης του πιο συχνά χρησιμοποιούμενου μέρους του προγράμματος).

Δομή if

Στο παράδειγμα που ακολουθεί τα S1, S2 και S3 είναι εντολές της γλώσσας υψηλού επιπέδου, και το E μια έκφραση που όταν υπολογιστεί δίνει αληθή ή ψευδή τιμή.

```
S1;
if (E) {
    S2;
}
S3;
```

Σύμφωνα με τη συνήθη σημασιολογία, τα S1 και S3 εκτελούνται πάντα, ενώ το S2 θα εκτελεστεί μόνο όταν το E είναι αληθές. Μια πιθανή υλοποίηση με εντολές μηχανής είναι η ακόλουθη:

```

; εντολές μηχανής για το S1
εντολή
...
εντολή
; εντολές μηχανής για τον υπολογισμό του E' (αντίστροφο του E)
εντολή
...
εντολή
if E' PC ← εδώ
; εντολές μηχανής για το S2
εντολή
...
εντολή
εδώ:
; εντολές μηχανής για το S3
εντολή
...
εντολή

```

Δομή if..else

Στο παράδειγμα που ακολουθεί τα S1, S2, S3 και S4 είναι εντολές της γλώσσας υψηλού επιπέδου, και το E μια έκφραση που όταν υπολογιστεί δίνει αληθή ή ψευδή τιμή.

```

S1;
if (E) {
    S2;
}
else {
    S3;
}
S4;

```

Μια πιθανή υλοποίηση με εντολές μηχανής είναι η ακόλουθη:

```

; εντολές μηχανής για το S1
εντολή
...
εντολή
; εντολές μηχανής για τον υπολογισμό του E' (αντίστροφο του E)
εντολή
...
εντολή
if E' PC ← εδώ
; εντολές μηχανής για το S2
εντολή
...
εντολή
PC ← εκεί
εδώ:
; εντολές μηχανής για το S3

```

εντολή

...

εντολή

εκεί:

; εντολές μηχανής για το S4

εντολή

...

εντολή

Τι είναι στην πραγματικότητα τα «εδώ» και «εκεί»;

Όταν ολοκληρωθεί η μεταγλώττιση ενός προγράμματος και παραχθεί ο εκτελέσιμος κώδικας, τα «εδώ» και «εκεί» θα έχουν υπολογιστεί ως συγκεκριμένες διευθύνσεις μνήμης, εκεί που θέλουμε πραγματικά να μεταφερθεί η ροή εκτέλεσης.

Ερωτήσεις

Έχοντας διαβάσει τα προηγούμενα θα πρέπει να μπορείτε να απαντήσετε στις παρακάτω ερωτήσεις:

1. Τι είναι και πώς λειτουργεί ο program counter (PC);
2. Σε μια ΚΜΕ ο program counter έχει τιμή 2000. Πού βρίσκεται στη μνήμη η εντολή που θα εκτελεστεί στη συνέχεια;
3. Η ροή εκτέλεσης έχει φτάσει στην εντολή1 του επόμενου κώδικα. Καμία από τις εντολές αυτές δεν είναι εντολή διακλάδωσης. Ποιες εντολές μηχανής θα εκτελεστούν στη συνέχεια;

εντολή1

εντολή2

εντολή3

εντολή4

4. Η ροή εκτέλεσης έχει φτάσει στην εντολή1 του επόμενου κώδικα. Ποιες εντολές μηχανής θα εκτελεστούν όταν η συνθήκη C είναι αληθής και ποιες όταν η C είναι ψευδής;

εντολή1

εντολή2

if C PC ← **εδώ**

εντολή3

εντολή4

PC ← **εκεί**

εδώ:

εντολή5

εντολή6

εκεί:

εντολή7

εντολή8

5. Δοκιμάστε να δώσετε μια παρόμοια υλοποίηση των δομών while και for.

Οι επόμενες ερωτήσεις είναι ευκαιρία για σκέψη: δεν έχετε διδαχτεί ακόμα την ύλη για να απαντήσετε ακριβώς, προσπαθήστε όμως να σκεφτείτε μια λύση χωρίς τεχνικές λεπτομέρειες.

1. Η κλήση συναρτήσεων στις γλώσσες υψηλού επιπέδου είναι κι αυτή μια αλλαγή της ροής του προγράμματος, άρα θα πρέπει να γίνει με κάποιο είδος εντολών μηχανής διακλάδωσης. Τι χρειάζεται όμως επιπλέον για την επιτυχή **επιστροφή** από τη συνάρτηση;
2. Κάποιες ΚΜΕ μπορούν να εκτελούν ταυτόχρονα δύο (ή 4, ή 8) διαφορετικά προγράμματα (threads). Τι αλλαγές πρέπει να γίνουν ως προς τον PC για να υποστηρίξουμε κάτι τέτοιο;
3. Αν το λειτουργικό σύστημα ενός υπολογιστή αποφασίσει να διακόψει ένα πρόγραμμα, τι θα πρέπει να αποθηκεύσει έτσι ώστε να μπορεί αργότερα να συνεχίσει την εκτέλεσή του;