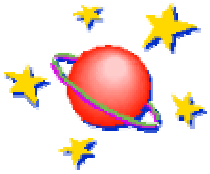

Ενότητα 2η: *Είδη εξαρτήσεων, Εξαρτήσεις Δεδομένων και ονομάτων*

Σκοπός **Ο σκοπός της ενότητας αυτής είναι να παρουσιάσει την έννοια της εξάρτησης μεταξύ εντολών και τα είδη της εξάρτησης. Συγκεκριμένα θα αναπτύξουμε την εξάρτηση δεδομένων, την επιρροή αυτής στην παραλληλία σε επίπεδο εντολής, καθώς και τρόπους αντιμετώπισής της. Τέλος, θα εξετάσουμε την εξάρτηση ονομάτων και θα αναπτύξουμε την τεχνική της μετονομασίας.**

Προσδοκώμενα Αποτελέσματα **Όταν θα έχετε μελετήσει την ενότητα, θα είστε σε θέση:**

<u>να καθορίζετε τον τρόπο εξάρτησης μιας εντολής από μια άλλη εντολή,</u>

<u>να αναγνωρίζετε τα είδη των εξαρτήσεων,</u>

<u>να αναγνωρίζετε τις εξαρτήσεις δεδομένων σε ένα πρόγραμμα και να προτείνετε τρόπους αντιμετώπισής τους,</u>

<u>να προσδιορίζετε πότε προκύπτουν εξαρτήσεις ονομάτων,</u>

<u>να εφαρμόζετε την τεχνική της μετονομασίας για την εξάλειψη των εξαρτήσεων ονομάτων.</u>

εξάρτηση δεδομένων, εξάρτηση ονομάτων, εξαρτώμενη εντολή, αντι-εξάρτηση, εξάρτηση εξόδου, μετονομασία καταχωρητή

Γενικά για τις Εξαρτήσεις

Ο καθορισμός του τρόπου εξάρτησης μιας εντολής από μία άλλη, παίζει αποφασιστικό ρόλο **:**

- στον καθορισμό του βαθμού της παραλληλίας στο πρόγραμμα,
- στον τρόπο εκμετάλλευσης αυτής της παραλληλίας,
- στην διαδικασία προγραμματισμού της σωλήνωσης .

Η εκμετάλλευση των δυνατοτήτων της παραλληλίας σε επίπεδο εντολής, προϋποθέτει να βρούμε ποιες εντολές μπορούν να εκτελεστούν παράλληλα. Δύο εντολές που δεν έχουν εξαρτήσεις, εκτελούνται ταυτόχρονα χωρίς καθυστερήσεις, υποθέτοντας όμως ότι η σωλήνωση έχει επαρκείς πόρους. Αντιθέτως, δύο εντολές που είναι εξαρτώμενες δεν μπορούν να εκτελεστούν παράλληλα και να αναδιαταχθούν.

Το κλειδί και στις δύο περιπτώσεις είναι να καθορίσουμε κατά πόσο μία εντολή εξαρτάται από μία άλλη.

Γενικά υπάρχουν τρία είδη εξαρτήσεων :

- Εξαρτήσεις δεδομένων (data dependencies).
- Εξαρτήσεις ονομάτων (name dependencies).
- Εξαρτήσεις ελέγχου (control dependencies).

Οι εξαρτήσεις είναι ιδιότητα των προγραμμάτων. Όταν μία εξάρτηση καταλήγει σε έναν πραγματικό κίνδυνο, ο οποίος έχει εντοπιστεί, και ο κίνδυνος προκαλεί πραγματική καθυστέρηση, τότε οι εξαρτήσεις γίνονται ιδιότητα της οργάνωσης της σωλήνωσης.

Εξαρτήσεις δεδομένων

Μια εντολή j είναι **εξαρτώμενη**, όσον αφορά στα δεδομένα, από μία εντολή i αν ισχύει μία από τις ακόλουθες συνθήκες **:**

- Η εντολή i παράγει ένα τουλάχιστον αποτέλεσμα που χρησιμοποιείται από την εντολή j.
- Η εντολή j είναι εξαρτώμενη, όσον αφορά στα δεδομένα, από την εντολή k, και η εντολή k είναι αντίστοιχα εξαρτώμενη από την εντολή i.

Η δεύτερη συνθήκη δηλώνει ότι μία εντολή εξαρτάται από μία άλλη, αν υπάρχει μεταξύ τους μία 'αλυσίδα' από εξαρτήσεις δεδομένων. Πιθανόν η αλυσίδα δεδομένων να είναι τόσο μεγάλη, όσο και ολόκληρο το πρόγραμμα.

Παράδειγμα:

Δίνεται ο παρακάτω βρόχος :

```
loop:  LD      F0 , 0 (R1)    ;  F0=στοιχείο πίνακα

       ADDD    F4 , F0 , F2   ;  πρόσθεση με το βαθμωτό
                                 μέγεθος του F2

       SD      0 (R1) , F4    ;  αποθήκευση του αποτελέσματος

       SUBI    R1 , R1 , #8   ;  μείωση δείκτη
                              ;  8 bytes  (ανά DW)

       BNEZ    R1 , loop      ;  διακλάδωση R1 !=μηδέν
```

Απάντηση:

Στο βρόχο υπάρχει εξάρτηση δεδομένων, όπως φαίνεται και από τα τόξα, τα οποία καταλήγουν στις εντολές που εξαρτώνται από τις προηγούμενες τους. Στην αρχή του προγράμματος υπάρχει εξάρτηση της εντολής ADD από την εντολή LD, επειδή η ADD χρειάζεται την τιμή F0 πριν ακόμα υπολογιστεί από την LD. Η εξάρτηση αυτή οδηγεί σε κίνδυνο και καθυστέρηση. Στη συνέχεια υπάρχει εξάρτηση της εντολής SD από την εντολή ADD, λόγω του F4. Τέλος υπάρχει μία εξάρτηση δεδομένων μεταξύ των εντολών SUBI και BNEZ. Η εξάρτηση αυτή προκαλεί καθυστέρηση επειδή μεταφέραμε τον έλεγχο της διακλάδωσης στο στάδιο ID. Όσο ο έλεγχος γινόταν στο στάδιο EX, η εξάρτηση αυτή δεν προκαλούσε καθυστέρηση. Τα τόξα υποδεικνύουν ότι η διάταξη των εντολών πρέπει να διατηρείται για την επίτευξη της σωστής εκτέλεσης του προγράμματος.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

- **Να εκτελέσετε των κώδικα του παραδείγματος χρησιμοποιώντας το λογισμικό προσομοίωσης του DLX, Windlx, το οποίο παρατίθεται στις ιστοσελίδες:**

  http://www.gup.uni-linz.ac.at/downloads/ti3/windlx

  http://www.mkp.com/pub/dlx
- **Να συγκρίνετε την απάντηση του παραδείγματος με τα αποτελέσματα από τη χρήση του λογισμικού προσομοίωσης Windlx.**
- **Για περισσότερες λεπτομέρειες καλό θα ήταν να συμβουλευόσασταν το παράρτημα Α: «Εργαλεία Προσομοίωσης του DLX».**

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

- **Μπορείτε να προσδιορίσετε τι συμβαίνει με την εκτέλεση δύο εντολών οι οποίες έχουν εξάρτηση δεδομένων;**
- **Θεωρείτε πως υπάρχει περίπτωση κινδύνου σε αυτή την περίπτωση; Να αιτιολογήσετε την απάντησή σας.**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2

Εάν δύο εντολές έχουν εξάρτηση δεδομένων, δεν μπορούν να εκτελεστούν ταυτόχρονα ή να έχουν πλήρη επικάλυψη. Η εξάρτηση υποδηλώνει την πιθανή ύπαρξη ενός ή περισσοτέρων RAW κινδύνων, αφού υπάρχει περίπτωση μια εντολή (έστω j) να εκτελέσει ανάγνωση ενός πηγαίου τελεστέου, πριν την ενημέρωση αυτού από την εντολή που προηγείται (έστω i). Σε αυτή την περίπτωση η εντολή παίρνει λάθος την παλιά τιμή και προκύπτει κίνδυνος RAW.

Εκτελώντας αυτές τις εντολές ταυτόχρονα σε έναν επεξεργαστή με κλείδωμα σωλήνωσης, αυτός εντοπίζει κινδύνους και καθυστερήσεις, μειώνοντας ή εξαλείφοντας έτσι, την επικάλυψη των εντολών. Ένας επεξεργαστής χωρίς κλείδωμα σωλήνωσης βασίζεται στο προγραμματισμό ενός μεταγλωττιστή. Όμως, ο μεταγλωττιστής δεν μπορεί να προγραμματίσει εξαρτώμενες εντολές ώστε να έχουμε πλήρη επικάλυψη.

Η σπουδαιότητα της εξάρτησης δεδομένων συνίσταται στο γεγονός ότι η εξάρτηση:

- **Υποδηλώνει την πιθανότητα κινδύνου.**
- **Καθορίζει την σειρά σύμφωνα με την οποία τα αποτελέσματα πρέπει να υπολογιστούν.**
- **Θέτει ένα ανώτατο όριο όσον αφορά στο βαθμό πιθανής εκμετάλλευσης της παραλληλίας.**

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Η εξάρτηση δεδομένων μπορεί να περιορίσει το βαθμό εκμετάλλευσης της παραλληλίας σε επίπεδο εκτέλεσης εντολής. Μπορείτε να υποδείξετε τρόπους με τους οποίους αυτό μπορεί να αποφευχθεί;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3

Για να αποφευχθεί ο περιορισμός στο βαθμό εκμετάλλευσης της παραλληλίας σε επίπεδο εντολής, μπορούμε να εφαρμόσουμε μία από τις δύο τεχνικές:

- **Διατήρηση της εξάρτησης των δεδομένων αλλά αποφεύγοντας τον κίνδυνο (προώθηση δεδομένων και πρόσβαση σε σειρά).**
- **Μείωση της εξάρτησης των δεδομένων, τροποποίηση του κώδικα (προγραμματισμός σωλήνωσης και ανάπτυξη βρόχου).**

Παράδειγμα:

Χρήση της ανάπτυξης του βρόχου για μείωση των εξαρτήσεων δεδομένων.
Σε αυτό το παράδειγμα καθώς και στα επόμενα θα χρησιμοποιήσουμε διακλαδώσεις χωρίς καθυστέρηση. Τα τόξα δείχνουν τις εξαρτήσεις δεδομένων που είναι μέσα στο ανεπτυγμένο σώμα και αφορούν τις SUBI εντολές.

```
loop:  LD      F0 , 0 (R1)
       ADDD    F4  , F0 , F2
       SD      0 (R1) , F4
       SUBI    R1 , R1 , #8  ; παράλειψη της BNEZ

       LD      F6 , 0 (R1)
       ADDD    F8 , F6 , F2
       SD      0 (R1) , F8
       SUBI    R1 , R1 , #8 ; παράλειψη της BNEZ

       LD      F10 , 0 (R1)
       ADDD    F12 , F10 , F2
       SD      0 (R1) , F12
       SUBI    R1 ,  R1 , #8 ; παράλειψη της BNEZ

       LD      F14 , 0 (R1)
       ADDD    F16 , F14 , F2
       SD      0 (R1) , F16
       SUBI    R1 , R1 , #8
       BNEZ    R1 , loop
```

Απάντηση:

Όπως δείχνουν τα τόξα, οι εντολές SUBI σχηματίζουν μία εξάρτηση που περιλαμβάνει τις εντολές SUBI, LD και SD, αναγκάζοντας το σώμα να εκτελείται σε σειρά. Η εντολή SUBI εφαρμόζεται όπου κρίνεται απαραίτητο, γεγονός που αυξάνει τον αριθμό των εντολών. Για να βελτιώσουμε τον παραπάνω βρόχο ο μεταγλωττιστής υπολογίζει τις ενδιάμεσες τιμές του R1, αποθηκεύει τον υπολογισμό στο offset των εντολών LD και SD, μειώνει την τελική τιμή του R1 κατά 32 κάνοντας τις τρεις εντολές SUBI περιττές, και επομένως, ο μεταγλωττιστής μπορεί να τις παραλείψει.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

- **Να εκτελέσετε των κώδικα του παραδείγματος χρησιμοποιώντας το λογισμικό προσομοίωσης του DLX, Windlx, το οποίο παρατίθεται στις ιστοσελίδες:**

  http://www.gup.uni-linz.ac.at/downloads/ti3/windlx

  http://www.mkp.com/pub/dlx

- **Να συγκρίνετε την απάντηση του παραδείγματος με τα αποτελέσματα από τη χρήση του λογισμικού προσομοίωσης Windlx.**
- **Για περισσότερες λεπτομέρειες καλό θα ήταν να συμβουλευόσασταν το παράρτημα Α: «Εργαλεία Προσομοίωσης του DLX».**

Παράδειγμα:

Έστω οι δύο διευθύνσεις 100(R4) και 20(R6), οι οποίες φαίνονται διαφορετικές. Αν το περιεχόμενο του R4 είναι 20 και το περιεχόμενο του R6 είναι 100, τότε οι δύο διευθύνσεις είναι πανομοιότυπες και αντιστοιχούν στη θέση μνήμης 120.

Απάντηση:

Έστω η διεύθυνση 20(R4). Αν το περιεχόμενο του R4 είναι 1000, τότε σε μία εντολή φόρτωσης LW η 'ενεργή' διεύθυνση θα είναι 1020. Το περιεχόμενο του R4 μπορεί να μεταβληθεί λόγω της εκτέλεσης μιας άλλης εντολής. Αν στη συνέχεια ακολουθεί μια εντολή φόρτωσης SW, τότε η 'ενεργή' διεύθυνση 20(R4) θα είναι διαφορετική. Ένας απλός κώδικας που δείχνει τη δεύτερη περίπτωση είναι ο παρακάτω **:**

```
LW    R1 ,  20(R4)
ADD   R4 ,  R4 , R2
SW    20(R4) , R1
```

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Έστω ο παρακάτω κώδικας:

```
.data
        .word 2,0,100
        .text
main:   lw r1,$DATA(r0)
        lw r4,$DATA +4(r0)
        lw r1,$DATA +8(r0)
loop:   add r4,r4,r2
        sw      $DATA(r0),r4
        subi    r1,r1,#1
        bnez    r1,loop
        trap    #0
```

- **Να προσδιορίσετε σε ποια σημεία του προγράμματος εμφανίζονται εξαρτήσεις δεδομένων.**
- **Να εκτελέσετε των κώδικα του παραδείγματος χρησιμοποιώντας το λογισμικό προσομοίωσης του DLX, Windlx, το οποίο παρατίθεται στις ιστοσελίδες:**
  http://www.gup.uni-linz.ac.at/downloads/ti3/windlx

  http://www.mkp.com/pub/dlx
- **Να συγκρίνετε την απάντησή σας με τα αποτελέσματα από την εκτέλεση του προγράμματος από τον προσομοιωτή Wndlx.**
- **Για περισσότερες λεπτομέρειες καλό θα ήταν να συμβουλευόσασταν το παράρτημα Α: «Εργαλεία Προσομοίωσης του DLX».**

Εξαρτήσεις ονομάτων

Η **εξάρτηση ονομάτων** προκύπτει όταν δύο εντολές χρησιμοποιούν τον ίδιο καταχωρητή ή την ίδια θέση μνήμης, που καλείται όνομα, αλλά δεν υπάρχει καμιά ροή δεδομένων μεταξύ των εντολών, που να σχετίζεται με το όνομα.

Υπάρχουν δύο είδη εξαρτήσεων ονομάτων μεταξύ μιας εντολής i, η οποία προηγείται της εντολής j**:**

1. Η **αντι-εξάρτηση** μεταξύ της εντολής i και της εντολής j, προκύπτει όταν η εντολή j γράφει στον καταχωρητή ή στη θέση μνήμης, όπου η εντολή i διαβάζει και έχει εκτελεστεί πρώτη. Η αντι-εξάρτηση αντιστοιχεί σε κίνδυνο WAR και ο εντοπισμός των WAR κινδύνων, οδηγεί στη διάταξη ενός μη- εξαρτώμενου ζεύγους εντολών. Παράδειγμα **:**

```
MULTD    F3 , F0 , F5
ADDD     F0 , F2 , F4
```

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Μπορείτε να εντοπίσετε σε ποιο σημείο προκαλείται ο κίνδυνος WAR;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Ο κίνδυνος WAR προκαλείται όταν η εντολή ADDD προσπαθήσει να γράψει στον καταχωρητή F0 πριν η MULTD διαβάσει από τον F0.

2. Η **εξάρτηση εξόδου**, προκύπτει όταν η εντολή i και η εντολή j γράφουν στον ίδιο καταχωρητή ή στην ίδια περιοχή μνήμης. Η διάταξη μεταξύ των εντολών πρέπει να διατηρείται. Οι εξαρτήσεις εξόδου διασφαλίζονται εντοπίζοντας WAW κινδύνους. Παράδειγμα **:**

```
ADD      R4 , R0 , R2
LW       R4 , 100(R1)
```

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 6

Μπορείτε να εντοπίσετε σε ποιο σημείο προκαλείται ο κίνδυνος WAW;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 6

Ο κίνδυνος WAW προκαλείται όταν η εντολή LW προσπαθήσει να γράψει στον καταχωρητή R4 πριν γράψει η εντολή ADD.

Στις εξαρτήσεις ονομάτων, σε αντιπαράθεση με τις πραγματικές εξαρτήσεις δεδομένων, δεν υπάρχει τιμή που να μεταδίδεται μεταξύ των εντολών. Αυτό υποδηλώνει, ότι εντολές που περικλείονται στην ίδια εξάρτηση ονόματος, μπορούν να εκτελεστούν ταυτόχρονα ή να αναδιαταχθούν, αν το όνομα καταχωρητή ή θέσης μνήμης αλλάξει, έτσι ώστε οι εντολές να μην συγκρούονται.

Αυτή η μετονομασία μπορεί πιο εύκολα να εφαρμοσθεί σε τελεστέους που βρίσκονται σε καταχωρητές και καλείται **μετονομασία καταχωρητή**. Η μετονομασία καταχωρητή μπορεί να επιτευχθεί με δύο τρόπους **:**

- Στατικά, από έναν μεταγλωττιστή.
- Δυναμικά με χρήση του υλικού.

Παράδειγμα:

Αναπτύξτε τον αρχικό βρόχο που χρησιμοποιήσαμε στα παραδείγματά μας, εξαλείφοντας τις επιπλέον εντολές SUBI και χρησιμοποιώντας τους ίδιους καταχωρητές σε κάθε αντίγραφο του κύριου σώματος του βρόχου.

- **Να υποδείξετε τις εξαρτήσεις δεδομένων και ονομάτων μέσα στο σώμα του βρόχου.**
- **Να δείξετε πώς η μετονομασία εξαλείφει τις εξαρτήσεις ονομάτων που ελαττώνουν την παραλληλία.**

Απάντηση:

Παραθέτουμε τον ανεπτυγμένο βρόχο που χρησιμοποιεί τους ίδιους καταχωρητές σε κάθε αντίγραφο του κύριου σώματος του βρόχου. Οι εξαρτήσεις δεδομένων είναι με τα γκρι τόξα, ενώ τα μαύρα δείχνουν τις εξαρτήσεις ονομάτων.

```
loop: LD      F0 , 0 (R1)
      ADDD    F4 , F0 , F2
      SD      0 (R1) , F4 ; παράλειψη των SUBI και BNEZ
      LD      F0 , -8 (R1)
      ADDD    F4 , F0 , F2
      SD      -8 (R1) , F4 ; παράλειψη των SUBI και BNEZ
      LD      F0 , -16 (R1)
      ADDD    F4 , F0 , F2
      SD      -16 (R1) , F4
      LD      F0 , -24 (R1)
      ADDD    F4 , F0 , F2
      SD      -24 (R1) , F4
      SUBI    R1 , R1 , #32
      BNEZ    R1 , loop
```

Οι εξαρτήσεις ονομάτων αναγκάζουν τις εντολές στο βρόχο, να είναι σχεδόν στο σύνολό τους διατεταγμένες, επιτρέποντας μόνο την αλλαγή διάταξης μεταξύ των εντολών LD και SD. Όταν οι καταχωρητές που χρησιμοποιούνται σε κάθε αντίγραφο του κυρίου σώματος του βρόχου μετονομάζονται, μόνο οι εξαρτήσεις δεδομένων παραμένουν. Η εφαρμογή της μετονομασίας φαίνεται παρακάτω.

```
loop: LD      F0 , 0 (R1)
      ADDD    F4 , F0 , F2
      SD      0 (R1) , F4 ; παράλειψη των SUBI και
                            BNEZ
      LD      F6 , -8 (R1)
      ADDD    F8 , F6 , F2
      SD      -8 (R1) , F8 ; παράλειψη των SUBI και
                            BNEZ
      LD      F10 , -16 (R1)
      ADD     F12 , F10 , F2
      SD      -16 (R1) , F12
```

```
LD      F14 , -24 (R1)
ADDD    F16 , F14 , F2
SD      -24 (R1) , F16
SUBI    R1 , R1 , #32
BNEZ    R1 , loop
```

Με την μετονομασία, τα αντίγραφα γίνονται ανεξάρτητα και επομένως μπορούν να επικαλυφθούν ή να εκτελεστούν παράλληλα.

Ανακεφαλαιώνοντας λοιπόν…

Υπάρχουν τρία είδη εξαρτήσεων : εξάρτηση δεδομένων, εξάρτηση ονομάτων και εξάρτηση ελέγχου.

Δύο εντολές που είναι εξαρτώμενες ,δεν μπορούν να εκτελεστούν παράλληλα ή να αναδιαταχθούν.

Οι εξαρτήσεις είναι ιδιότητα των προγραμμάτων. Όταν μία εξάρτηση καταλήγει σε έναν πραγματικό κίνδυνο, ο οποίος έχει εντοπιστεί, και ο κίνδυνος προκαλεί πραγματική καθυστέρηση, τότε οι εξαρτήσεις γίνονται ιδιότητα της οργάνωσης της σωλήνωσης.

Οι εξαρτήσεις δεδομένων προκύπτουν όταν μία εντολή χρησιμοποιεί το αποτέλεσμα προηγούμενης εντολής.

Εάν δύο εντολές έχουν εξάρτηση δεδομένων, δεν μπορούν να εκτελεστούν ταυτόχρονα ή να επικαλυφθούν τελείως. Η εξάρτηση υποδηλώνει την πιθανή ύπαρξη ενός ή περισσοτέρων RAW κινδύνων.

Η εξάρτηση δεδομένων μπορεί να περιορίσει το βαθμό εκμετάλλευσης της παραλληλίας σε επίπεδο εκτέλεσης εντολής. Αυτό μπορεί να αποφευχθεί με δύο τρόπους:

- **Διατηρώντας την εξάρτηση των δεδομένων αλλά αποφεύγοντας τον κίνδυνο (προώθηση δεδομένων και πρόσβαση σε σειρά).**
- **Μειώνοντας την εξάρτηση των δεδομένων, τροποποιώντας τον κώδικα (προγραμματισμός σωλήνωσης και ανάπτυξη βρόχου).**

Η εξάρτηση ονομάτων προκύπτει όταν δύο εντολές χρησιμοποιούν τον ίδιο καταχωρητή ή την ίδια θέση μνήμης, που καλείται όνομα. Επειδή δεν υπάρχει καμιά ροή δεδομένων μεταξύ των εντολών που να σχετίζεται με το όνομα, μπορούν να εκτελεστούν παράλληλα.

Η εξάρτηση ονομάτων μπορεί να αντιμετωπιστεί με την αλλαγή του ονόματος (μετονομασία), ώστε να μην υπάρχει σύγκρουση δεδομένων. Η μετονομασία μπορεί να υλοποιηθεί είτε στατικά, είτε δυναμικά.