13 Η ΤΕΧΝΙΚΗ ΤΟΥ ΑΓΩΓΟΥ

ΣΚΟΠΟΣ ΤΟΥ ΚΕΦΑΛΑΙΟΥ

- Να γίνει κατανοητό ότι στόχος της τεχνολογίας υπολογιστών ήταν και είναι πάντα η βελτίωση της απόδοσης των συστημάτων
- Να παρουσιαστεί η τεχνική του αγωγού (pipeline) που στις σημερινές ΚΜΕ θεωρείται αυτονόητη δυνατότητα
- Να φανεί ότι διαμορφώνοντας λίγο διαφορετικά την αρχιτεκτονική της ΚΜΕ μπορούμε να επιτύχουμε μικρότερους χρόνους εκτέλεσης των εντολών

13.1 Εισαγωγή

Από την εμφάνιση των πρώτων κιόλας εμπορικών κεντρικών μονάδων επεξεργασίας (ΚΜΕ) εμφανίστηκε και η ανάγκη για βελτίωση της απόδοσής τους που ήταν συνυφασμένη και με την αντίστοιχη συχνότητα λειτουργίας. Τόσο η πολυπλοκότητα των κυκλωμάτων όσο και η συχνότητα λειτουργίας βασίζονταν πάντα στις δυνατότητες της τεχνολογίας που υπήρχαν σε κάθε εποχή. Βασικό χαρακτηριστικό της τεχνολογίας ήταν και είναι η κλίμακα ολοκλήρωσης η οποία με απλά λόγια καθορίζει το πλήθος και το είδος των ημιαγωγών που μπορούν να χρησιμοποιηθούν σε μια δεδομένη επιφάνεια (chip). Όσο μεγαλύτερη είναι η κλίμακα ολοκλήρωσης τόσο μεγαλύτερες δυνατότητες έχει ένα ολοκληρωμένο κύκλωμα αφού σε αυτό μπορούν να αναπτυχθούν περισσότερα κυκλώματα που να καλύπτουν πιο απαιτητικές εφαρμογές. Για παράδειγμα, με τη σημερινή τεχνολογία υπάρχει η δυνατότητα συνύπαρξης πολλαπλών πυρήνων στο ίδιο ολοκληρωμένο κύκλωμα με αποτέλεσμα να μπορούν να εκτελεστούν περισσότερες από μια εντολές στο ίδιο χρονικό διάστημα. Για να είναι αυτό δυνατό, κάθε πυρήνας περιλαμβάνει όλα τα απαραίτητα κυκλώματα που είναι αναγκαία για την ανεξάρτητη εκτέλεση εντολών (π.χ. αριθμητική και λογική μονάδα, καταχωρητές, κλπ).

Αν κάποιος από εσάς αναρωτιέται γιατί η υπολογιστική ισχύς (π.χ. ταχύτητα υπολογισμών και εκτέλεσης εντολών της ΚΜΕ) δεν είναι ποτέ αρκετή θα τον παραπέμψουμε να μελετήσει και άλλου είδους εφαρμογές από αυτές που στηρίζονται στην απόδοση του κλασικού υπολογιστικού συστήματος. Τέτοιες είναι

αυτές που ανήκουν στις εφαρμογές πραγματικού χρόνου όπου τα αποτελέσματα της επεξεργασίας έχουν νόημα και αξία μόνο αν παράγονται σε συγκεκριμένο χρονικό διάστημα.

Για παράδειγμα, το αυτόματο σύστημα πλοήγησης σε ένα αεροσκάφος θα πρέπει να διορθώνει την πορεία του και να διαμορφώνει τα στοιχεία πτήσης μέσα σε ελάχιστο χρόνο. Αυτό σημαίνει ότι θα πρέπει να γίνονται συγκεκριμένοι υπολογισμοί οι οποίοι αν ξεφύγουν από το χρονικό όριο τότε τα αποτελέσματα θα είναι καταστροφικά. Οι σύγχρονες εφαρμογές που επεξεργάζονται χιλιάδες ή εκατομμύρια δεδομένα τα οποία αλλάζουν σε κλάσματα του δευτερολέπτου πρέπει να βασίζονται σε υπολογιστικά συστήματα με ιδιαίτερα υψηλές αποδόσεις.

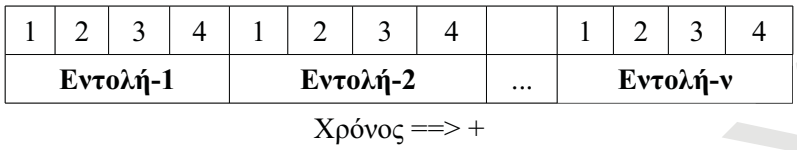
13.2 Η Τεχνική του αγωγού

Πριν από αρκετά χρόνια που δεν υπήρχε η δυνατότητα σχεδιασμού πολλαπλών πυρήνων λόγω της χαμηλής κλίμακας ολοκλήρωσης αλλά και άλλων τεχνολογικών περιορισμών, οι σχεδιαστές μηχανικοί προσπαθούσαν με ορισμένες παραλλαγές της αρχιτεκτονικής της ΚΜΕ να επιτύχουν μεγαλύτερες επιδόσεις. Μια τέτοια τεχνική (παραλλαγή) είναι αυτή του αγωγού (pipeline) η οποία βασίζεται σε μια απλή προσέγγιση και αποτελεί στάνταρτ πλέον κύκλωμα σε όλες τις σημερινές ΚΜΕ. Η συγκεκριμένη τεχνική βασίζεται στην αναδιαμόρφωση της διαδικασίας εκτέλεσης εντολής όπως τη γνωρίσαμε μέχρι τώρα αλλά σε επίπεδο υλοποίησης. Σε προηγούμενο κεφάλαιο παρουσιάστηκε η διαδικασία εκτέλεσης εντολής ως ακολούθως:

- Προσκόμιση κώδικα εντολής
- Αποκωδικοποίηση
- Προσκόμιση ορισμάτων
- Εκτέλεση εντολής

Στη συμβατική υλοποίηση της αρχιτεκτονικής (ακολουθιακή λογική) κάθε εντολή ολοκληρώνεται ως προς την εκτέλεσή της και στη συνέχεια το σύστημα ξεκινά τη διαδικασία για την εκτέλεση της επόμενης (σχήμα 13.1).

Σχήμα 13.1 Ακολουθιακή εκτέλεση εντολών



Η εκτέλεση των εντολών με αυτή την προσέγγιση είναι αποκλειστικά ακολουθιακή αφού οι βαθμίδες που είναι υπεύθυνες για την υλοποίηση των βημάτων 1, 2, 3 και 4 δεσμεύονται μέχρι την παραγωγή του αντίστοιχου αποτελέσματος.

Για να συγκρίνουμε όμως την απόδοση της συμβατικής προσέγγισης (ακολουθιακή εκτέλεση) με αυτή του αγωγού που θα παρουσιαστεί στη συνέχεια θα πρέπει να ορίσουμε ορισμένες έννοιες και μέτρα σύγκρισης. Το βασικό μέτρο σύγκρισης είναι ο συνολικός χρόνος που απαιτείται για την ολοκλήρωση της εκτέλεσης ν εντολών (ν επαναλήψεις της ίδιας εντολής). Το σύνολο ν των εντολών το ονομάζουμε διανυσματική εντολή μήκους ν. Αν όλες οι εντολές εκτελούσαν πρόσθεση τότε θα μπορούσαμε να πούμε ότι πρόκειται για διανυσματική εντολή πρόσθεσης μήκους ν. Κάθε εντολή απαιτεί την υλοποίηση τεσσάρων βημάτων (υπολειτουργίες) τα οποία απαιτούν κάποιο χρόνο. Για να είναι δυνατή η σύγκριση θεωρούμε εξισορροπημένους χρόνους για τα βήματα, δηλαδή κάθε βήμα απαιτεί για να υλοποιηθεί ίδιο σταθερό χρόνο t_0. Μια διανυσματική εντολή ε απαρτίζεται από $p(\varepsilon)$ υπολειτουργίες (εδώ αντιπροσωπεύουν τα τέσσερα βήματα της διαδικασίας εκτέλεσης). Έτσι, θα ισχύει ότι $p(\varepsilon)=4$. Εφόσον κάθε εντολή απαιτεί $p(\varepsilon)$ υπολειτουργίες με χρόνο t_0 για κάθε υπολειτουργία, τότε ο συνολικός χρόνος θα είναι $p(\varepsilon)*t_0$. Επεκτείνοντας αυτό τον τύπο για διανυσματική εντολή μήκους ν, ο συνολικός χρόνος που απαιτείται για την εκτέλεση ν εντολών θα είναι:

$$T_{\alpha\kappa}(\varepsilon,\nu) = \nu * p(\varepsilon) * t_0 \quad (13.1)$$

Ξεπερνώντας την προηγούμενη προσέγγιση, θεωρούμε τώρα ότι η αρχιτεκτονική είναι έτσι διαμορφωμένη ώστε να σχηματίζεται ένας αγωγός που αποτελείται από τέσσερις βαθμίδες για την υλοποίηση των τεσσάρων βημάτων (σχήμα 13.2):

Σχήμα 13.2 Βαθμίδες αγωγού

Η λειτουργία των βαθμίδων είναι:

- **B1**: Προσκόμιση κώδικα εντολής
- **B2**: Αποκωδικοποίηση
- **B3**: Προσκόμιση ορισμάτων
- **B4**: Εκτέλεση εντολής

Το αποτέλεσμα κάθε βαθμίδας αποτελεί είσοδο για την επόμενη. Η φιλοσοφία αυτής της τεχνικής βασίζεται στην ανεξαρτησία των αντίστοιχων βημάτων της διαδικασίας εκτέλεσης μεταξύ διαφορετικών εντολών (όπου αυτό είναι εφικτό). Για παράδειγμα η ακολουθία των εντολών

```
X=A+B
Y=X+1
```

δεν μπορεί να βελτιστοποιηθεί επειδή η δεύτερη εντολή έχει εξάρτηση δεδομένων από την πρώτη. Έτσι, η σειρά εκτέλεσης είναι προκαθορισμένη.

Ας φανταστούμε όμως τώρα ότι πρόκειται να εκτελεστούν δύο εντολές ανεξάρτητες μεταξύ τους. Αρχικά, η πρώτη εντολή καταλαμβάνει τη βαθμίδα B1 για προσκόμιση του κώδικα εντολής. Την επόμενη χρονική στιγμή και όταν ολοκληρωθεί η διαδικασία στη βαθμίδα B1, η πρώτη εντολή περνά στη φάση της βαθμίδας B2 ελευθερώνοντας ταυτόχρονα τη θέση στη βαθμίδα B1 η οποία καταλαμβάνεται από τη δεύτερη εντολή που τώρα μπαίνει στη φάση της προσκόμισης του κώδικα της εντολής. Στη συνέχεια, η πρώτη εντολή περνά στην επόμενη φάση της βαθμίδας B3, η πρώτη περνά στη φάση της βαθμίδας B2 ενώ μένει ελεύθερη η βαθμίδα B1 η οποία τώρα μπορεί να δεχτεί και τρίτη εντολή που μπαίνει στην αρχή της διαδικασίας. Έτσι, σε κάθε χρονική στιγμή, κάθε εντολή περνά στην επόμενη βαθμίδα ελευθερώνοντας ταυτόχρονα την προηγούμενή της. Καθώς ο αγωγός βρίσκεται σε λειτουργία, κάθε εντολή προχωρά και όταν φτάσει στη βαθμίδα B4 παράγεται το αντίστοιχο αποτέλεσμα.

Ας δούμε όμως τώρα σε βήματα τι συμβαίνει όταν πρόκειται να εκτελεστεί μια διανυσματική εντολή μήκους v=5 (5 το πλήθος εντολές).

Σχήμα 13.3 Βήμα 1. (Εντολή 1, B1)

Βαθμίδα								
B1	ΠΚΕ1							
B2								
B3								
B4								
Χρόνος	t_0	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
Αποτέλεσμα								

Στο πρώτο βήμα (χρόνος t_0), η εντολή 1 εισέρχεται στη βαθμίδα B1 για την έναρξη της διαδικασίας εκτέλεσης (Προσκόμιση Κώδικα Εντολής 1 – ΠΚΕ1).

Σχήμα 13.4 Βήμα 2. (Εντολή 2, B1), (Εντολή 1, B2)

Βαθμίδα								
B1	ΠΚΕ1	ΠΚΕ2						
B2		ΑΕ1						
B3								
B4								
Χρόνος	t_0	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
Αποτέλεσμα								

Στο δεύτερο βήμα (χρόνος $2t_0$), η εντολή 1 εισέρχεται στη βαθμίδα B2 για τη διαδικασία της αποκωδικοποίησης (Αποκωδικοποίηση Εντολής 1 – ΑΕ1) ενώ η B1 που απελευθερώνεται θα δεχτεί τη δεύτερη εντολή για την έναρξη της διαδικασίας εκτέλεσης (Προσκόμιση Κώδικα Εντολής 2 – ΠΚΕ2).

Σχήμα 13.5 Βήμα 3. (Εντολή 3, Β1), (Εντολή 2, Β2), (Εντολή 1, Β3)

Βαθμίδα								
Β1	ΠΚΕ1	ΠΚΕ2	ΠΚΕ3					
Β2		ΑΕ1	ΑΕ2					
Β3			ΠΟΕ1					
Β4								
Χρόνος	t_0	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
Αποτέλεσμα								

Στο τρίτο βήμα (χρόνος $3t_0$), η εντολή 1 εισέρχεται στη βαθμίδα Β3 για τη διαδικασία της προσκόμισης ορισμάτων εντολής (ΠΟΕ1) ενώ η Β2 που απελευθερώνεται θα δεχτεί την δεύτερη εντολή για την Αποκωδικοποίηση Εντολής (ΑΕ2) η οποία με τη σειρά της θα ελευθερώσει την Β1 για την έναρξη της διαδικασίας της εντολής 3 (Προσκόμιση Κώδικα Εντολής 3 – ΠΚΕ3).

Σχήμα 13.6 Βήμα 4. (Εντολή 3, Β2), (Εντολή 2, Β3), (Εντολή 1, Β4)

Βαθμίδα								
Β1	ΠΚΕ1	ΠΚΕ2	ΠΚΕ3	ΠΚΕ4				
Β2		ΑΕ1	ΑΕ2	ΑΕ3				
Β3			ΠΟΕ1	ΠΟΕ2				
Β4				ΕΚΕ1				
Χρόνος	t_0	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
Αποτέλεσμα				$1_ο$				

Στο βήμα 4 (χρόνος $4t_0$) ο αγωγός έχει γεμίσει (όλες οι βαθμίδες είναι δεσμευμένες). Έτσι, η πρώτη εντολή θα παράγει το πρώτο αποτέλεσμα σε χρόνο $4t_0$. Θυμηθείτε στην ακολουθιακή εκτέλεση ότι η πρώτη εντολή παράγει αποτέλεσμα σε χρόνο $T_{ακ}=1*p(\varepsilon)*t_0=4t_0$. Άρα για την εκτέλεση μιας μόνο εντολής δεν υπάρχει διαφορά στις δύο προσεγγίσεις (ακολουθιακή, αγωγός) αναφορικά με το χρόνο παραγωγής αποτελέσματος-εκτέλεσης εντολής.

Όμως αφού γεμίσει ο αγωγός θα υπάρχει παραγωγή αποτελεσμάτων κάθε t_0. Αυτό σημαίνει ότι μετά το γέμισμα του αγωγού κάθε εντολή παράγει αποτέλεσμα σε χρόνο t_0 και όχι σε $4t_0$ που ισχύει στην ακολουθιακή εκτέλεση. Χαρακτηριστικό είναι το σχήμα 13.7.

Σχήμα 13.7 Παραγωγή αποτελεσμάτων

Βαθμίδα								
B1	ΠΚΕ1	ΠΚΕ2	ΠΚΕ3	ΠΚΕ4	ΠΚΕ5			
B2		ΑΕ1	ΑΕ2	ΑΕ3	ΑΕ4			
B3			ΠΟΕ1	ΠΟΕ2	ΠΟΕ3			
B4				ΕΚΕ1	ΕΚΕ2			
Χρόνος	t_0	$2t_0$	$3t_0$	$4t_0$	$5t_0$	$6t_0$	$7t_0$	$8t_0$
Αποτέλεσμα				$1_ο$	$2_ο$			

Παρατηρώντας το σχήμα 13.7 διαπιστώνουμε ότι:

- Ο αγωγός γεμίζει μετά από χρόνο $(p(\varepsilon)-1)*t_0$, δηλαδή $3t_0$
- Μετά το γέμισμα του αγωγού παράγονται αποτελέσματα σε χρονικά διαστήματα t_0
- Το πρώτο αποτέλεσμα παράγεται σε χρόνο $p(\varepsilon)*t_0$, δηλαδή $4t_0$

Εφόσον το πρώτο αποτέλεσμα παράγεται στο χρόνο $p(\varepsilon)*t_0=4t_0$, τα υπόλοιπα $\nu-1$ αποτελέσματα απαιτούν χρόνο $(\nu-1)*t_0$. Άρα το τελευταίο αποτέλεσμα (ολοκλήρωση διαδικασίας) θα παραχθεί σε χρόνο:

$$p(\varepsilon)*t_0+(\nu-1)*t_0=(p(\varepsilon)-1+\nu)t_0$$

Ο επιπλέον χρόνος που πρέπει να παρέλθει ώστε να ξεκινήσει η εκτέλεση των εντολών συμβολίζεται ως $\alpha(\varepsilon)$ και είναι ο χρόνος που χρειάζεται ο αγωγός για να γεμίσει και εξαρτάται από τη διανυσματική εντολή ε. Έτσι, ο συνολικός χρόνος για την εκτέλεση διανυσματικής εντολής μήκους ν διαμορφώνεται ως εξής:

$$T_{\alpha\gamma}(\varepsilon,\nu)=(\alpha(\varepsilon)-1+p(\varepsilon)+\nu)t_0 \quad (13.2)$$

Θέλοντας να κάνουμε μια συνολική σύγκριση μεθόδων εκτέλεσης εντολών θα θεωρήσουμε και μια τρίτη περίπτωση σύμφωνα με την οποία η εκτέλεση των εντολών γίνεται από κ υπολογιστές ή κ ΚΜΕ. Σε μια τέτοια περίπτωση οι εντολές μοιράζονται στις κ υπολογιστικές μονάδες.

Ο πίνακας 13.1 συνοψίζει τους τύπους που δίνουν το συνολικό χρόνο εκτέλεσης διανυσματικής εντολής μήκους ν για τις τρεις προσεγγίσεις.

Πίνακας 13.1 Συνολικοί χρόνοι εκτέλεσης

Ακολουθιακή εκτέλεση	$T_{\alpha\kappa}(\varepsilon,\nu) = \nu * p(\varepsilon) * t_0$
Τεχνική αγωγού	$T_{\alpha\gamma}(\varepsilon,\nu)=(\alpha(\varepsilon)-1+p(\varepsilon)+\nu)t_0$
Υπολογιστικές μονάδες	$T_{\kappa}(\varepsilon,\nu)=(\nu/K)*p(\varepsilon)*t_0$

Περίληψη κεφαλαίου

Η βελτιστοποίηση αναφέρεται σε παρεμβάσεις ή και επανασχεδιασμό της υφιστάμενης αρχιτεκτονικής προκειμένου να βελτιωθούν οι επιδόσεις σε συγκεκριμένες διαδικασίες.

Σε μια πιο τυπική προσέγγιση υιοθετούνται συγκεκριμένα στατιστικά μέτρα και αντικειμενικές συναρτήσεις προκειμένου να γίνεται αξιολόγηση του επιπέδου της βελτιστοποίησης.

Σε αυτό το κεφάλαιο έγινε παρουσίαση της τεχνικής του αγωγού που αποτελεί μια από τις δημοφιλέστερες προσεγγίσεις μέχρι σήμερα. Η αρχιτεκτονική και η υλοποίηση του αγωγού βασίζεται στη διάσπαση της διαδικασίας των τεσσάρων βημάτων για την εκτέλεση των εντολών σε ισάριθμες ανεξάρτητες βαθμίδες. Κάθε βαθμίδα υλοποιεί και από ένα βήμα της διαδικασίας ανεξάρτητα από τις άλλες. Αυτό σημαίνει ότι οι βαθμίδες μπορεί να περιέχουν τμήματα της διαδικασίας αλλά για διαφορετικές εντολές που είναι ανεξάρτητες μεταξύ τους. Έτσι, σε κάθε μονάδα του χρόνου πολλές εντολές μαζί και βήμα προς βήμα προχωρούν προς την τελική υλοποίησή τους.

Η τεχνική του αγωγού μειώνει το χρόνο παραγωγής αποτελεσμάτων των εντολών κάτι που δεν μπορεί να γίνει στην κλασική αρχιτεκτονική όπου κάθε εντολή πρέπει να περιμένει την ολοκλήρωση (ως προς την εκτέλεση) της προηγούμενης.

ΛΥΜΕΝΕΣ ΑΣΚΗΣΕΙΣ

Άσκηση 1 Να σχεδιαστεί πίνακας που να περιέχει τους συνολικούς χρόνους εκτέλεσης διανυσματικών εντολών μήκους ν=1,2,5,10,100 χρησιμοποιώντας την ακολουθιακή εκτέλεση και την τεχνική του αγωγού.
Σημείωση: Θεωρήστε ότι α(ε)=0 και t_0=1sec.

Λύση

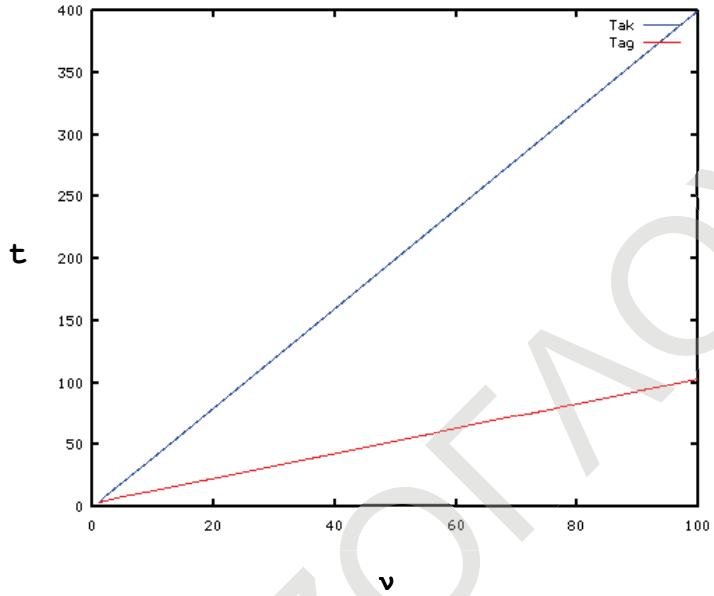
Για τον υπολογισμό των αντίστοιχων χρόνων θα χρησιμοποιήσουμε τους ακόλουθους τύπους:

$T_{\alpha\kappa}(\varepsilon,\nu) = \nu * p(\varepsilon) * t_0$
$T_{\alpha\gamma}(\varepsilon,\nu) = (\alpha(\varepsilon)-1+p(\varepsilon)+\nu)t_0$

v	1	2	5	10	100
$T_{\alpha\kappa}$	4	8	20	40	400
$T_{\alpha\gamma}$	4	5	8	13	103
	Συνολικοί χρόνοι				

Άσκηση 2 Να σχεδιαστούν σε διάγραμμα οι χρόνοι εκτέλεσης που υπολογίστηκαν στην προηγούμενη άσκηση ώστε να φανεί και οπτικά η διαφοροποίηση της απόδοσης των δύο προσεγγίσεων.

Λύση



Tak=ακολουθιακός χρόνος, Tag=χρόνος αγωγού

ΑΛΥΤΕΣ ΑΣΚΗΣΕΙΣ

Άσκηση 1 Να λυθούν οι δύο προηγούμενες ασκήσεις συνυπολογίζοντας και το συνολικό χρόνο εκτέλεσης με την τρίτη προσέγγιση (κ υπολογιστικές μονάδες).

Άσκηση 2 Με ποιο μήκος διανυσματικής εντολής ν έχει νόημα η χρήση της τεχνικής του αγωγού;

Άσκηση 3 Υπολογίστε το ρυθμό μεταβολής του συνολικού χρόνου εκτέλεσης των δύο προσεγγίσεων βάσει του διαγράμματος της λυμένης άσκησης 2.