Ενότητα 1η : *Είσοδος / Έξοδος - Η απόδοση ενός συστήματος*

Σκοπός **Ο σκοπός της ενότητας αυτής είναι να τονίσει την αναγκαιότητα της Εισόδου / Εξόδου σε ένα υπολογιστικό σύστημα, καθώς επίσης και να ορίσει τις έννοιες του χρόνου απόκρισης και του ρυθμού διαμεταγωγής.**

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

κατονομάζετε συσκευές Εισόδου/Εξόδου,

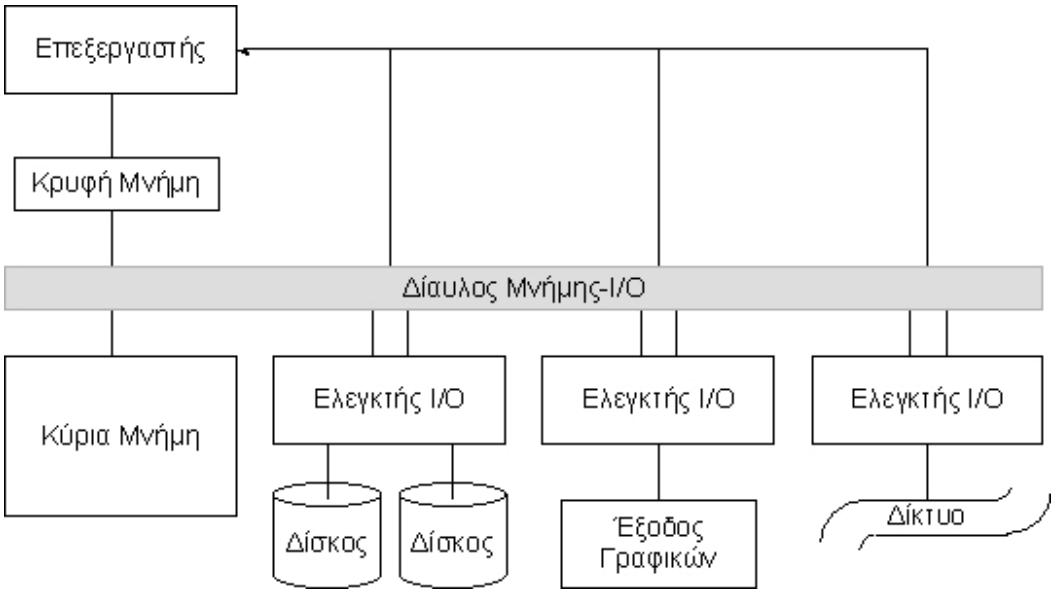
εξηγείτε τους λόγους για τους οποίους οι συσκευές I/O έχουν παραμεληθεί σε σχέση με τα υπόλοιπα τμήματα ενός υπολογιστή,

υπολογίζετε τον χρόνο απόκρισης και το ρυθμό διαμεταγωγής.

συσκευή εισόδου/εξόδου, χρόνος απόκρισης, ρυθμός διαμεταγωγής, χρόνος εισόδου, χρόνος απόκρισης συστήματος, χρόνος σκέψης, χρόνος διεξαγωγής, ρυθμός άφιξης, αλγόριθμος ουράς

Είσοδος / έξοδος (I/O) και η χρησιμότητά της

Η Είσοδος/Εξοδος (θα την αποκαλούμε από εδώ και στο εξής εν συντομία I/O - Input/Output), αποτελεί το τμήμα του υπολογιστικού συστήματος με το οποίο το σύστημα είναι σε θέση να ανταλλάξει δεδομένα με το περιβάλλον του, είτε με άλλους υπολογιστές είτε με τον άνθρωπο-χρήστη. Οι συσκευές εκείνες που απαρτίζουν το τμήμα I/O του υπολογιστή και με τις οποίες πραγματοποιείται η ανταλλαγή των δεδομένων ονομάζονται συσκευές εισόδου/εξόδου (συσκευές I/O).



Σχήμα 5.1.1 -Τυπική συλλογή συσκευών I/O. Παρουσιάζεται η δομή ενός συστήματος με τις συσκευές I/O του. Οι συνδέσεις ανάμεσα στις συσκευές I/O, τον επεξεργαστή και τη μνήμη ονομάζονται αρτηρίες (buses).

Η οπτική γωνία απ΄ την οποία αντιμετωπίζουμε το τμήμα I/O μπορεί να διαφέρει, π.χ. οι ηλεκτρολόγοι μηχανικοί το βλέπουν σαν ένα σύνολο από τσιπ (ολοκληρωμένα κυκλώματα), καλώδια, τροφοδοτικά, ηλεκτροκινητήρες και άλλα φυσικά στοιχεία από τα οποία αποτελείται. Οι προγραμματιστές το βλέπουν σύμφωνα με τον τρόπο που αυτό επικοινωνεί με το λογισμικό, δηλαδή τις εντολές που δέχεται, τις λειτουργίες που εκτελεί και τα σφάλματα που μπορεί να αναφέρει στο λογισμικό.
Συγκεκριμένα, το σύστημα αρχείων (file system) ασχολείται με συσκευές I/O αφήνοντας το τμήμα που εξαρτάται από την κάθε συσκευή σε λογισμικό χαμηλότερου επιπέδου, τους οδηγούς συσκευών (device drivers).

 **Οι μονάδες I/O συνήθως αποτελούνται από το μηχανικό και το ηλεκτρονικό μέρος (ελεγκτής συσκευής ή προσαρμογέας). Τις περισσότερες φορές το λειτουργικό σύστημα συνεργάζεται με τον ελεγκτή ή προσαρμογέα και όχι με τη συσκευή.**

 **Το I/O είναι απαραίτητο και αναπόσπαστο τμήμα ενός υπολογιστικού συστήματος.** Μπορούμε να υποθέσουμε ότι ένας υπολογιστής χωρίς συσκευές εισόδου/ εξόδου είναι σαν ένα αυτοκίνητο χωρίς ρόδες. Γι' αυτό το λόγο, η εξέλιξη και η τελειοποίηση των συσκευών I/O ώστε να γίνουν πιο γρήγορες και αποδοτικές είναι επιτακτική. Άλλωστε, ο χρήστης ενός υπολογιστή κυρίως ενδιαφέρεται για όσο δυνατό μικρότερο χρόνο απόκρισης στις εργασίες που του έχει αναθέσει.

Επιπλέον, σε μία εποχή που οι μηχανές - από τους προσωπικούς υπολογιστές της χαμηλής κατηγορίας (low-end PCs) έως τους μεγάλους υπολογιστές (mainframes) και ακόμα τους υπερυπολογιστές (supercomputers) - κατασκευάζονται με την ίδια βασική τεχνολογία μικροεπεξεργαστών, οι δυνατότητες του συστήματος I/O είναι συχνά ένα από τα χαρακτηριστικά εκείνα που κάνει ένα σύστημα ξεχωριστό. Σχεδόν όλοι οι μικρο-υπολογιστές και οι μίνι-υπολογιστές χρησιμοποιούν το μοντέλο μονής αρτηρίας (single bus) για την επικοινωνία ανάμεσα στην ΚΜΕ (CPU) και στις συσκευές I/O. Οι μεγάλοι υπολογιστές (large mainframes) χρησιμοποιούν ένα διαφορετικό μοντέλο, με πολλαπλές αρτηρίες και εξειδικευμένους υπολογιστές I/O που καλούνται δίαυλοι I/O, απαλλάσσοντας την ΚΜΕ από ένα μέρος του φορτίου αυτού.
Πολλές από τις πρόσφατες υλοποιήσεις στη βιομηχανία των υπολογιστών είναι συναρπαστικές, τόσο για τις νέες δυνατότητές τους σε I/O, όσο και για την ισχύ του επεξεργαστή τους. Αυτό συμβαίνει επειδή οι μηχανές αλληλεπιδρούν με τους ανθρώπους μέσω του I/O.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Είναι γεγονός πως η τεχνολογική ανάπτυξη των συσκευών I/O ακολουθεί πιο αργούς ρυθμούς σε σύγκριση με τη εξέλιξη των υπολοίπων τμημάτων ενός υπολογιστικού συστήματος. Μπορείτε να εξηγήσετε σε 10 το πολύ γραμμές τους λόγους για τους οποίους συμβαίνει αυτό; Να συγκρίνετε την απάντησή σας με την παράγραφο που ακολουθεί: «Η παραμέληση της Εισόδου / Εξόδου».

Η παραμέληση της Εισόδου/Εξόδου

Η παραμέληση των συσκευών I/O οφείλεται σε διάφορους λόγους όπως:

- Η ανάπτυξη του I/O πραγματοποιείται με βραδείς ρυθμούς σε σχέση με τα υπόλοιπα τμήματα ενός υπολογιστικού συστήματος.

- Οι δυσκολίες που παρουσιάζουν τόσο η εκτίμηση της απόδοσης όσο και ο σχεδιασμός ενός συστήματος I/O έχουν κλονίσει το κύρος του I/O.
- Η έρευνα επικεντρώνεται στο σχεδιασμό των επεξεργαστών, ενώ η προκατάληψη εναντίον του I/O μορφοποιείται στο ευρύτερα χρησιμοποιούμενο μέτρο για την απόδοση ενός υπολογιστικού συστήματος, το χρόνο επεξεργαστή ή χρόνο ΚΜΕ, του οποίου η τιμή δεν εξαρτάται από την απόδοση του συστήματος I/O.

Ωστόσο, η ποιότητα του I/O ενός υπολογιστή δε μπορεί να μετρηθεί με το χρόνο ΚΜΕ, επειδή αυτός εξορισμού αγνοεί το I/O. Η σχέση του χρόνου ΚΜΕ και του ολικού χρόνου φαίνεται στο αμέσως παρακάτω, ενδεικτικό παράδειγμα. Η δεύτερης-κλάσης αντιμετώπιση που τυγχάνει το I/O είναι εμφανής ακόμα και στην ονομασία των συσκευών I/O, που συχνά αποκαλούνται "περιφερειακές". Η μελέτη του νόμου του Amdahl θα πρέπει να μας υπενθυμίσει πως η παραμέληση του I/O είναι επικίνδυνη.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Θυμάστε τι είναι ο χρόνος της ΚΜΕ και με ποιον τρόπο υπολογίζεται; Για περισσότερες λεπτομέρειες καλό θα ήταν να ανατρέξετε στο 1^{ο} κεφάλαιο: «Βασικές αρχές υπολογιστικών συστημάτων».

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2

Ο χρόνος της ΚΜΕ είναι ο χρόνος που χρειάζεται για να εκτελεστούν οι υπολογισμοί της ΚΜΕ, μη συμπεριλαμβανομένου του χρόνου που η ΚΜΕ περιμένει για είσοδο / έξοδο ή αυτού που η ΚΜΕ τρέχει άλλα προγράμματα.
Ο χρόνος της ΚΜΕ ενός προγράμματος μπορεί να υπολογιστεί ως εξής:

$$\text{Χρόνος ΚΜΕ} = \left(\sum_{i=1}^{n} CPI_i \times IC_i \right) \times \text{κύκλος ρολογιού}$$

Όπου CPI: αριθμός κύκλων ρολογιού ανά εντολή
IC: πόσες φορές εκτελείται η εντολή i σε ένα πρόγραμμα

Παράδειγμα

Υποθέστε πως έχουμε ένα πρόγραμμα μετρήσεων (benchmark) που εκτελείται σε 100 δευτερόλεπτα, από τα οποία τα 90 είναι χρόνος της ΚΜΕ και τα υπόλοιπα είναι χρόνος I/O. Αν ο χρόνος της ΚΜΕ βελτιώνεται κατά 50% ανά έτος για τα επόμενα

πέντε έτη, αλλά ο χρόνος του I/O δε βελτιώνεται, πόσο ταχύτερα θα εκτελείται το πρόγραμμά μας στο τέλος της περιόδου των πέντε ετών;

Απάντηση:

Γνωρίζουμε ότι:

Ολικός χρόνος = χρόνος ΚΜΕ + χρόνος I/O = 90 + χρόνος I/O = 100

χρόνος I/O = 10 sec

Η ποιότητα του I/O ενός υπολογιστή δεν μπορεί να μετρηθεί με το χρόνο ΚΜΕ, επειδή αυτός εξ' ορισμού αγνοεί το I/O. Κατά συνέπεια, οι νέοι χρόνοι ΚΜΕ και οι χρόνοι εκτέλεσης που προκύπτουν έχουν υπολογιστεί στον ακόλουθο πίνακα:

Μετά n έτη	Χρόνος ΚΜΕ	Χρόνος I/O	Ολικός χρόνος	% χρόνου I/O
0	90sec	10 sec	100 sec	10%
1	90/1.5 = 60 sec	10 sec	70 sec	14%
2	60/1.5 = 40 sec	10 sec	50 sec	20%
3	40/1.5 = 27 sec	10 sec	37 sec	27%
4	27/1.5 = 18 sec	10 sec	28 sec	36%
5	18/1.5 = 12 sec	10 sec	22 sec	45%

Πίνακας 1 - Η βελτίωση στην απόδοση της ΚΜΕ μέσα σε 5 έτη είναι: 90/12 = 7.5 Όμως η βελτίωση στο ***συνολικό χρόνο*** είναι μόλις: 100/22 = 4.5 και ο ***χρόνος του I/O*** έχει αυξηθεί από το 10% στο 45% του ***συνολικού χρόνου***.

Η *βελτίωση στην απόδοση της ΚΜΕ* μέσα σε πέντε έτη είναι: **90/12=7.5**
Όμως, *η βελτίωση στο συνολικό χρόνο* είναι μόλις: **100/22=4.5**
και ο *χρόνος του I/O* έχει αυξηθεί από το 10% στο 45% του συνολικού χρόνου.

 Η απόδοση του I/O μπορεί να καθορίσει σε σημαντικό βαθμό και κατά συνέπεια να περιορίσει την ολική απόδοση και την αποτελεσματικότητα του συστήματος. Η απόδοση του συστήματος εισόδου/εξόδου μπορεί να ρυθμίζει έως και το μισό χρόνο λειτουργίας του συστήματος.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Να εξηγήσετε το λόγο για τον οποίο ο χρόνος ΚΜΕ είναι ένα σημαντικό επιχείρημα που χρησιμοποιείται εναντίον της ανάπτυξης του I/O.

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3

Ο χρόνος ΚΜΕ είναι ένα μέτρο μέτρησης της απόδοσης του συστήματος που εξαρτάται αποκλειστικά από το χρόνο που ο επεξεργαστής παραμένει απασχολημένος. Δηλαδή ο χρόνος ΚΜΕ είναι ανεξάρτητος του συστήματος I/O και της απόδοσής του. Έτσι, αφού η συνολική απόδοση του συστήματος καθορίζεται από το χρόνο ΚΜΕ και δεν έχει σχέση με το πόσο αποδοτικό ή όχι είναι το σύστημα I/O, δεν ενδιαφερόμαστε για την είσοδο/έξοδο.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Υποθέστε ότι έχουμε μία διαφορά μεταξύ του χρόνου του επεξεργαστή και του χρόνου απόκρισης κατά 10% , και ότι επιταχύνουμε τον επεξεργαστή 10 φορές, ενώ παραμελούμε το I/O. Κατά πόσο θα επιταχυνθεί στην πραγματικότητα το σύστημά μας και τι ποσοστό της δύναμης του επεξεργαστή θα χαθεί; Τι θα γινόταν αν επιταχύναμε τον επεξεργαστή 100 φορές;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

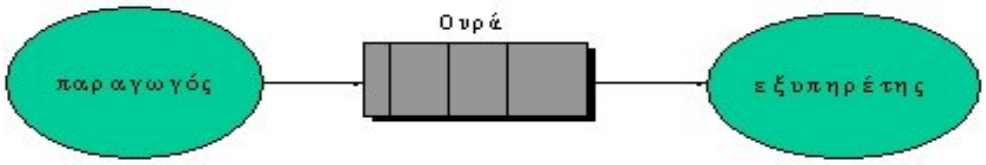
Αν υποθέσουμε ότι ο συνολικός χρόνος απόκρισης (δηλαδή ο χρόνος λειτουργίας του επεξεργαστή και ο χρόνος I/O) είναι 100 δευτερόλεπτα , τότε ο επεξεργαστής θα καταναλώνει τα 90 δευτερόλεπτα και άρα το σύστημα I/O τα υπόλοιπα 100 - 90 = 10 δευτερόλεπτα. Αν επιταχύνουμε τον επεξεργαστή 10 φορές αυτός θα απαιτεί πλέον χρόνο 90/10 = 9 δευτερολέπτων. Ο συνολικός χρόνος απόκρισης θα είναι τώρα 9 + 10 = 19 δευτερόλεπτα (αφού ο χρόνος I/O Δε βελτιώνεται). Επομένως, η ταχύτητα του συστήματος θα βελτιωθεί κατά 100/19 = 5 φορές. Δηλαδή, παρόλο που η ταχύτητα του επεξεργαστή αυξήθηκε κατά 10 φορές, η συνολική ταχύτητα του συστήματος αυξήθηκε μόνο κατά 5. Επίσης, ο επεξεργαστής στην αρχή είχε το 90% του χρόνου δικό του. Μετά την επιτάχυνση έχει το 9/19 = 0,473 = 47,3% , χάθηκε έτσι η μισή περίπου ισχύς του σε σχέση με το συνολικό χρόνο του συστήματος.

Χρόνος απόκρισης εναντίον ρυθμού διαμεταγωγής

Η Είσοδος/ Έξοδος αποτελεί το τμήμα του υπολογιστικού συστήματος με το οποίο το σύστημα είναι σε θέση να ανταλλάξει δεδομένα με το περιβάλλον του (είτε με άλλους υπολογιστές είτε με τον άνθρωπο-χρήστη). Οι συσκευές εκείνες που απαρτίζουν το τμήμα I/O του υπολογιστή και με τις οποίες πραγματοποιείται η ανταλλαγή των δεδομένων ονομάζονται **συσκευές εισόδου/εξόδου** (συσκευές I/O).

Δύο αντιφατικά μέτρα – Το Μοντέλο παραγωγού-εξυπηρέτη

Τα δύο επόμενα σχήματα (βλέπε σχ. 5.1.2 και σχ. 5.1.3), δείχνουν με ποιο τρόπο συσχετίζονται ο χρόνος απόκρισης και ο ρυθμός διαμεταγωγής. Το σχήμα 5.1.2 μας δείχνει το απλό μοντέλο παραγωγού-καταναλωτή.



Σχήμα 5.1.2 – Το μοντέλο παραγωγού – εξυπηρέτη. Ο παραγωγός δημιουργεί και τοποθετεί διεργασίες στην ενδιαμέση μνήμη, τις οποίες παραλαμβάνει ο εξυπηρέτης (server). Ο διεργασίες του που δημιουργεί ο παραγωγός, πρέπει να εκτελεστούν στην ενδιάμεση μνήμη, που λειτουργεί ως μια δομή ουράς. Παράλληλα ο εξυπηρέτης παίρνει διεργασίες από την ενδιάμεση μνήμη και τις εκτελεί.

Ο *χρόνος απόκρισης* ορίζεται ως ο χρόνος που χρειάζεται μία διεργασία από τη στιγμή που τοποθετείται στην ουρά (μνήμη) μέχρι τη στιγμή που ο εξυπηρέτης την ολοκληρώνει (τα αποτελέσματα εμφανίζονται στην έξοδο).
Ο *ρυθμός διαμεταγωγής* είναι ο μέσος αριθμός διεργασιών που θα εκτελεστούν από τον εξυπηρέτη σε ένα χρονικό διάστημα.

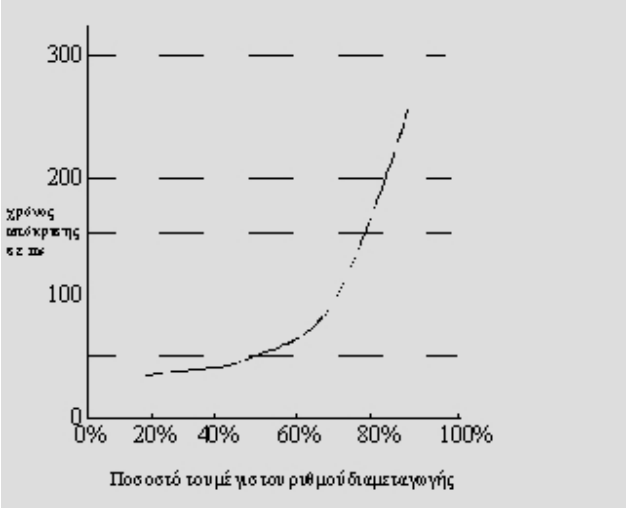
ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Μπορείτε να εξηγήσετε τον τρόπο με τον οποίο σχετίζονται ο ρυθμός διαμεταγωγής και ο χρόνος ΚΜΕ ενός συστήματος;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Ο χρόνος ΚΜΕ αποτελεί το ποσοστό του χρόνου που ο επεξεργαστής είναι απασχολημένος. Όσο περισσότερο χρόνο παραμένει ο επεξεργαστής απασχολημένος, τόσες περισσότερες διεργασίες θα εκτελεί. Έτσι, ο ρυθμός διαμεταγωγής αυξάνεται ή μειώνεται ανάλογα με το χρόνο ΚΜΕ.

Για να έχουμε το μέγιστο δυνατό ρυθμό διαμεταγωγής, ο εξυπηρέτης δε θα πρέπει ποτέ να είναι αδρανής (idle), και γι' αυτό το λόγο η ενδιάμεση μνήμη δε θα πρέπει να είναι ποτέ άδεια. Ο χρόνος απόκρισης, από την άλλη μεριά, στην ουσία μετρά το χρόνο που ξοδεύεται στην ενδιάμεση μνήμη, και γι' αυτό ο χρόνος απόκρισης ελαχιστοποιείται όταν αυτή είναι άδεια.
Η σύγκρουση των δύο μέτρων είναι καταφανής!



Σχήμα 5.1.3 – Ο ρυθμός διαμεταγωγής εναντίον χρόνου απόκρισης. Στη γραφική παράσταση απεικονίζεται η σχέση μεταξύ του χρόνου απόκρισης και του ρυθμού διαμεταγωγής σ'ένα τυπικό σύστημα εισόδου/εξόδου. Η μεγάλη κλίση στην καμπύλη αποτελεί την περιοχή όπου λίγο περισσότερος ρυθμός διαμεταγωγής έχει ως αποτέλεσμα πολύ μεγαλύτερο χρόνο απόκρισης ή , αντίστροφα , λίγο μικρότερος χρόνος απόκρισης έχει ως αποτέλεσμα πολύ χαμηλότερο ρυθμό διαμεταγωγής.

Η χρησιμότητα του χρόνου απόκρισης

Το ενδιαφέρον των χρηστών για το χρόνο απόκρισης οφείλεται στη δημιουργία του διαδραστικού (interactive) λογισμικού, των σταθμών εργασίας και των προσωπικών

υπολογιστών. Μπορεί επίσης να είναι ασύμφορο να περιμένουμε από το σύστημα να εκτελεί άλλες διεργασίες και παράλληλα να αναμένει την είσοδο/έξοδο. Αυτό συμβαίνει επειδή η κύρια μνήμη πρέπει να είναι μεγάλη, αλλιώς η συχνή σελιδοποίηση (paging) από την εναλλαγή διεργασιών στην πραγματικότητα θα αύξανε τη διαδικασία εισόδου/εξόδου. Τέλος, κάποιες εφαρμογές, όπως η διαδικασία συναλλαγών, θέτουν αυστηρά όρια για το χρόνο απόκρισης, ως τμήμα της ανάλυσης της απόδοσης του συστήματος. Σε μία συνδιαλλαγή η αλληλεπίδραση (interaction or transaction) με έναν υπολογιστή διακρίνουμε τρία χρονικά μεγέθη :

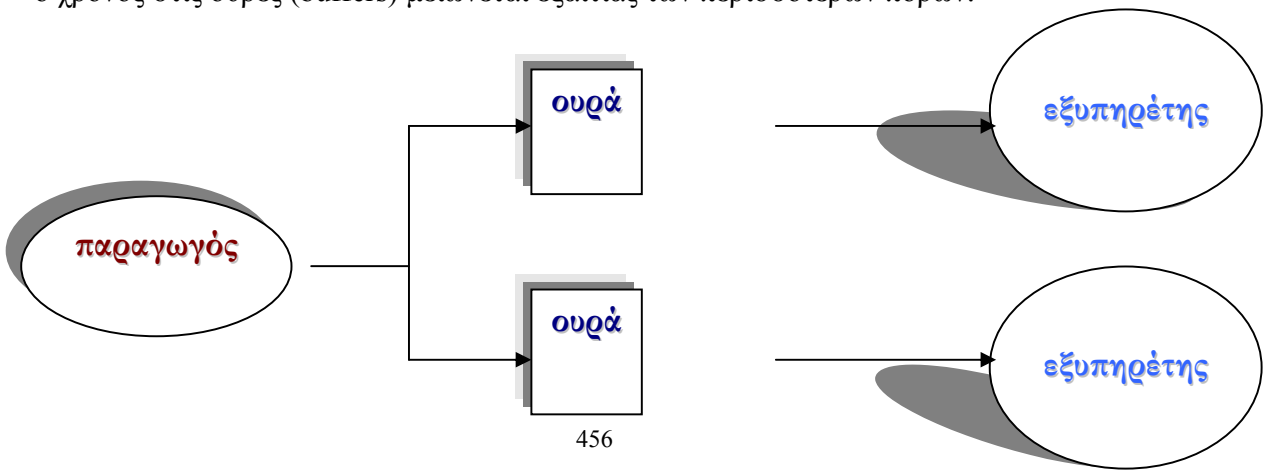
- **Χρόνος εισόδου** - ο χρόνος που θέλει ο χρήστης για να δώσει την εντολή.
- **Χρόνος απόκρισης συστήματος** - ο χρόνος μεταξύ της στιγμής που ο χρήστης δίνει την εντολή και της στιγμής που η πλήρης απόκριση παρουσιάζεται.
- **Χρόνος σκέψης** - ο χρόνος από την υποδοχή της απόκρισης μέχρι ο χρήστης να ξεκινήσει να δίνει την επόμενη εντολή.

Το άθροισμά τους ονομάζεται **χρόνος διεξαγωγής** (της δοσοληψίας). Έρευνες έχουν δείξει ότι η μείωση στο χρόνο απόκρισης στην πραγματικότητα μειώνει τον χρόνο διεξαγωγής πράξεων κατά περισσότερο από μόνο την μείωση του χρόνου απόκρισης. Αυτό συμβαίνει επειδή οι άνθρωποι χρειάζονται λιγότερο χρόνο για να σκεφτούν όταν παίρνουν απαντήσεις συντομότερα.

Ο Ρυθμός διαμεταγωγής είναι εξίσου απαραίτητος

Υπάρχουν εφαρμογές στις οποίες το ενδιαφέρον μας επικεντρώνεται στο ρυθμό διαμεταγωγής και όχι τόσο στο χρόνο απόκρισης. Παράδειγμα ενός τέτοιου περιβάλλοντος μπορεί να είναι ένα γραφείο επεξεργασίας φορολογικών στοιχείων του Υπουργείου Οικονομικών. Η Υπηρεσία ενδιαφέρεται κυρίως για την επεξεργασία ενός μεγάλου αριθμού από φορολογικές δηλώσεις σε δεδομένο χρονικό διάστημα (κάθε φορολογική δήλωση αποθηκεύεται ξεχωριστά και είναι αρκετά μικρή). Σε αυτή την περίπτωση ενδιαφερόμαστε για την επεξεργασία όσο το δυνατόν περισσότερων δηλώσεων σε όσο το δυνατό μικρότερο χρονικό διάστημα.
Όλα θα ήταν πιο απλά, αν το να βελτιώνεις την απόδοση σήμαινε πάντοτε βελτίωση σε χρόνο απόκρισης και σε ρυθμό διαμεταγωγής.

Προσθέτοντας έναν ακόμα εξυπηρέτη (server) αυξάνουμε το ρυθμό διαμεταγωγής, αφού μοιράζουμε τα δεδομένα σε 2 δίσκους αντί σε 1 και έτσι οι διεργασίες μπορούν να ικανοποιούνται ή να εξυπηρετούνται παράλληλα (βλέπε σχ. 5.1.4) .Δυστυχώς αυτό δεν βοηθά το χρόνο απόκρισης ***εκτός εάν το πλήθος*** των διεργασιών παραμένει σταθερό και ο χρόνος στις ουρές (buffers) μειώνεται εξαιτίας των περισσότερων πόρων.



Σχήμα 5.1.4 – Η προσθήκη ενός ακόμα εξυπηρέτη, σε σχέση με το μοντέλο παραγωγού καταναλωτή, αυξάνει το ρυθμό διαμεταγωγής.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 6

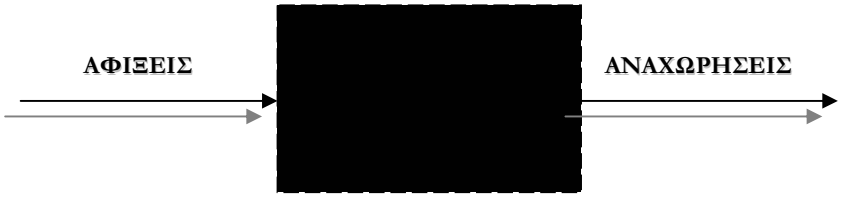
Σε ένα μεγάλο αριθμό εφαρμογών απαιτούνται ταυτόχρονα και υψηλός ρυθμός διαμεταγωγής και μικροί χρόνοι απόκρισης. Να δώσετε ένα παράδειγμα υπολογιστικού συστήματος που συμβαίνει αυτό. Να συγκρίνετε την απάντησή σας με την παράγραφο που ακολουθεί: «Συνύπαρξη ρυθμού διαμεταγωγής και χρόνου απόκρισης».

Συνύπαρξη ρυθμού διαμεταγωγής και χρόνου απόκρισης

Σε ένα μεγάλο αριθμό εφαρμογών απαιτούνται ταυτόχρονα και υψηλός ρυθμός διαμεταγωγής και μικροί χρόνοι απόκρισης. Ενδεικτικά αναφέρουμε τις αυτόματες ταμειακές μηχανές (Automatic Teller Machines – ATMs), τα συστήματα κράτησης αεροπορικών θέσεων, τις εισαγωγές παραγγελιών και τα συστήματα παρακολούθησης καταλόγων, τους εξυπηρέτες αρχείων (file servers) και τις μηχανές καταμερισμού χρόνου (timesharing).
Σε τέτοια περιβάλλοντα ενδιαφερόμαστε *τόσο για το χρόνο που χρειάζεται κάθε εργασία για να εκτελεστεί όσο και για το πλήθος των εργασιών που μπορούμε να επεξεργαστούμε σ' ένα χρονικό διάστημα.* **Γιατί συμβαίνει αυτό;**
Για παράδειγμα, αν μία αυτόματη ταμειακή μηχανή χρειάζεται 15-20 λεπτά για να επεξεργαστεί την αίτηση ενός πελάτη τότε δεν θα έχει και μεγάλη σημασία το πλήθος των αιτήσεων που μπορεί να επεξεργαστεί η μηχανή ανά ώρα – δε θα απομείνει κανείς πελάτης στην τράπεζα!
Παρόμοια αν μπορεί να επεξεργάζεται μια αίτηση γρήγορα, αλλά η μηχανή έχει τη δυνατότητα να χειρίζεται μόνο ένα μικρό πλήθος αιτήσεων ταυτόχρονα ,δεν θα μπορεί να εξυπηρετήσει πολλές αιτήσεις ή το κόστος του συστήματος ανά αίτηση θα είναι πολύ υψηλό.

Θεωρήματα για υπολογισμό του *χρόνου απόκρισης* και του *ρυθμού διαμεταγωγής*

Εάν ένα σύστημα είναι σε σταθερή κατάσταση, τότε ο αριθμός των διεργασιών που μπαίνουν στο σύστημα πρέπει να είναι ίσος με τον αριθμό των διεργασιών που «φεύγουν» από αυτό.



Σχήμα 5.1.5 - Το I/O σύστημα ως μαύρο κουτί.

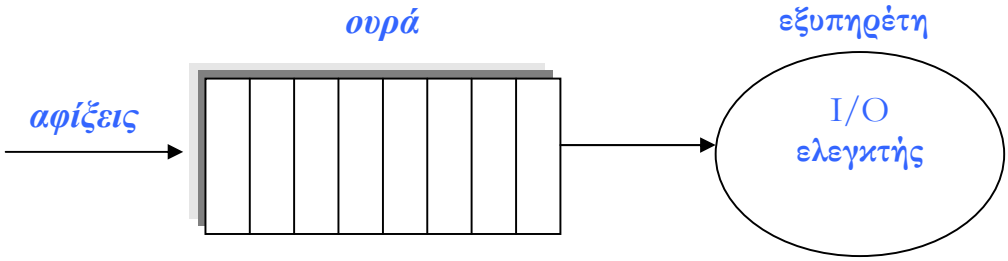
Συνήθως ενδιαφερόμαστε για την πολύ μελλοντική ή την σταθερή κατάσταση ενός ολόκληρου I/O συστήματος περισσότερο από τις αρχικές συνθήκες με τις οποίες ξεκινάμε. Εκτός εάν κάνουμε την απλούστευση ότι μελετάμε συστήματα σε ισορροπία : *Ο ρυθμός της εισόδου πρέπει να είναι ίσος με τον ρυθμό της εξόδου.*

Αυτό μας οδηγεί στον **Little's Law (Νόμο του Little)**, ο οποίος συνδυάζει τον μέσο αριθμό των διεργασιών στο σύστημα, τον μέσο ρυθμό άφιξης των νέων διεργασιών και τον μέσο χρόνο εκτέλεσης μιας διεργασίας.

Μέσος αριθμός διεργασιών στο σύστημα = *ρυθμός άφιξης * μέσο χρόνο απόκρισης*.

Ο Little's Law αναφέρεται σε κάθε σύστημα σε ισορροπία εφόσον δεν δημιουργούνται νέες διεργασίες ή δεν καταστρέφονται αυτές που υπάρχουν.

.



Σχήμα 5.1.6 - Το μοντέλο του μονού εξυπηρέτη. Σε αυτήν την περίπτωση μια αίτηση I/O «φεύγει» με την ολοκλήρωσή της από τον εξυπηρέτη

Ο Little's Law και μια σειρά ορισμών οδηγούν σε αρκετές χρήσιμες εξισώσεις. Ειδικότερα, δίνονται οι ακόλουθοι ορισμοί:

•*Χρόνος(εξυπηρέτη):* ο μέσος χρόνος για να εξυπηρετηθεί μια διαδικασία =
Ρυθμός εξυπηρέτησης 1/ Χρόνος(εξυπηρέτη): (συμβολίζεται με μ).

•*Χρόνος (ουράς) :* ο μέσος χρόνος για κάθε διαδικασία στην ουρά.

•*Χρόνος(συστήματος): ο* μέσος χρόνος ανά διεργασία στο σύστημα ή
ο χρόνος απόκρισης (Χρόνος (ουράς)+Χρόνος (εξυπηρέτη))

•*Ρυθμός άφιξης:* μέσος αριθμός διεργασιών που φτάνουν ανά δευτερόλεπτο (συμβολίζεται με λ).

•*Μήκος (εξυπηρέτη):* μέσος αριθμός διεργασιών σε εξυπηρέτηση.

•*Μήκος (ουράς):* μέσο μήκος ουράς.

•*Μήκος (συστήματος):* μέσος αριθμός διεργασιών στο σύστημα (Μήκος (εξυπηρέτη) +Μήκος (συστήματος))

 Οι ακόλουθες διευκρινήσεις δίνονται προς αποφυγή παρανοήσεων:
Η πρόταση: "*πόσο χρόνο πρέπει να περιμένει στην ουρά μια διεργασία πριν εξυπηρετηθεί", αναφέρεται στο Χρόνο(ουράς),* ενώ η ακόλουθη: "*πόσο χρόνο χρειάζεται μια διεργασία μέχρι να ολοκληρωθεί" αναφέρεται στο* Χρόνο(συστήματος)

Χρησιμοποιώντας τους παραπάνω ορισμούς ο Little's Law μπορεί να επαναδιατυπωθεί ως εξής:

Μήκος (συστήματος) = Ρυθμός άφιξης* Χρόνος (συστήματος)

Μπορούμε ακόμα να βρούμε ***πόσο*** απασχολημένο είναι ένα σύστημα από τον ακόλουθο τύπο:

Απασχόληση Εξυπηρέτη (ρ) = Ρυθμός άφιξης / Ρυθμός εξυπηρέτησης

Η τιμή του ***ρ*** πρέπει να είναι μεταξύ 0 και 1, διαφορετικά θα έφταναν περισσότερες διεργασίες από αυτές που θα μπορούσαν να εξυπηρετηθούν, οπότε το σύστημα δεν θα ήταν σε ισορροπία.

 ***Ο τρόπος με τον οποίο η ουρά παραδίδει διεργασίες στον εξυπηρέτη (server) καλείται αλγόριθμος ουράς (queue discipline).***

Ο πιο απλός και κοινός αλγόριθμος είναι η first-in-first-out (FIFO). Υιοθετώντας τον αλγόριθμο FIFO μπορούμε συνδέσουμε τον ***χρόνο αναμονής*** στην ουρά με τον ***μέσο αριθμό διεργασιών*** στην ουρά:

Χρόνος (συστήματος) = Μήκος (ουράς) * Χρόνος (εξυπηρέτη) +

(Μέσος χρόνος εξυπηρέτησης διεργασιών μετά την άφιξη μιας νέας διεργασίας.)

Για να υπολογίσουμε τον τελευταίο όρο της παραπάνω εξίσωσης χρειαζόμαστε δύο στοιχεία:

1ο) Weighted mean time =

$$\frac{f_1*T_1 + f_2*T_2 + \ldots + f_v*T_v}{\sum_{i=1}^{v} f_i}$$

Όπου $\mathbf{T}_i$ είναι ο χρόνος για μια διεργασία i και $\mathbf{f}_i$ είναι η συχνότητα της διεργασίας i .

2ο) Variance =

$$\frac{f_1*T_1^2 + f_2*T_2^2 + \ldots + f_v*T_v^2}{\sum_{i=1}^{v} f_i}$$

- Weighted mean time

Προκειμένου να απλοποιήσουμε τις μονάδες, χρησιμοποιούμε τον τύπο:
squared coefficient of variance: **C** = *variance / Weighted mean time [(2)/(1)]*

Ολοκληρώνοντας, ο χρόνος αναμονής μιας νέας διεργασίας προκειμένου ο εξυπηρέτης (server) να ολοκληρώσει την προηγούμενη από αυτή διεργασία, καλείται ***Average residual service time*** και δίνεται από τον ακόλουθο τύπο:

*Average residual service time=(1/2)** Weighted mean time*(1+ C)

Η θεωρία ουρών (*queuing theory)* είναι πιο αξιόπιστη όταν δεν χρειάζονται ακριβείς απαντήσεις. Τα αληθινά συστήματα είναι πολύ περίπλοκα ώστε η δοθείσα μέθοδο να επιτύχει μία ακριβή ανάλυση. Κατά συνέπεια, καταλήγουμε σε υποθέσεις για το μοντέλο της ουράς:
•Το σύστημα είναι σε ισορροπία.
•Ο αριθμός των αιτήσεων είναι άπειρος.
•Ο εξυπηρέτης μπορεί να ασχοληθεί με τον επόμενο πελάτη αμέσως μετά το τέλος της ενασχόλησής του με τον προηγούμενο.
•Δεν υπάρχει όριο στο μήκος της ουράς και αυτή ακολουθεί τον αλγόριθμο FIFO.
•Όλες οι διεργασίες πρέπει να ολοκληρωθούν.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 7

Έστω ότι ο μέσος χρόνος για να ικανοποιηθεί μια αίτηση του δίσκου είναι 50ms και ένα σύστημα I/O με πολλούς δίσκους παίρνει περίπου 200 I/O αιτήσεις το δευτερόλεπτο. Να βρείτε ποιος είναι ο μέσος αριθμός των I/O αιτήσεων στον εξυπηρέτη του δίσκου;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 7

Εφαρμόζοντας τον Little's Law έχουμε:

Μήκος(εξυπηρέτη) = Ρυθμός άφιξης* Χρόνος(ουράς)

= 200*0.05 δευτερόλεπτα/δευτερόλεπτα =10 (διεργασίες)

Άρα υπάρχουν 10 αιτήσεις κατά μέσο όρο στον εξυπηρέτη.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 8

Να βρείτε ένα τύπο για το μέσο χρόνο αναμονής στην ουρά (*Χρόνος (ουράς)*) σε σχέση με το μέσο χρόνο εξυπηρέτησης (*Χρόνος (εξυπηρέτη)*), την απασχόληση του εξυπηρέτη (server utilization) και την *squared coefficient of variance.*

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 8

Όλες οι διεργασίες που προηγούνται στην ουρά (Μήκος (ουράς)) από τη νέα διεργασία, πρέπει να εκτελεστούν προτού αυτή εξυπηρετηθεί. Κάθε μία για να εκτελεστεί, απαιτεί ένα μέσο χρόνο εξυπηρέτησης. Αν μια διαδικασία είναι στον εξυπηρέτη τότε χρειάζεται *Average residual service time για να ολοκληρωθεί.* Η πιθανότητα ο εξυπηρέτης να είναι απασχολημένος είναι η *server utilization,* επομένως ο χρόνος που περιμένουμε να απαιτείται για εξυπηρέτηση είναι:

server utilization* Average residual service time.

Άρα έχουμε:

$$Χρόνος_{ουράς} = Μήκος_{ουράς} * Χρόνος_{εξυπηρέτη} + \underline{server\ utilization} * \underline{Average\ residual\ service\ time}$$

Αντικαθιστώντας το $Μήκος_{ουράς}$ με *$Ρυθμό\ άφιξης * Χρόνος_{ουράς}$* και την *Average residual service time* με το ίσο της έχουμε:

$$Χρόνος_{ουράς} = server\ utilization * (1/2 * Χρόνος_{εξυπηρέτη} * (1+C)) + (Ρυθμό\ άφιξης * Χρόνος_{ουράς}) * Χρόνος_{εξυπηρέτη}$$

Το οποίο γίνεται :

$$= server\ utilization * (1/2 * Χρόνος_{εξυπηρέτη} * (1+C)) + server\ utilization * Χρόνος_{ουράς}$$

Μετά παίρνουμε:

$$Χρόνος_{ουράς} - server\ utilization * Χρόνος_{ουράς} = server\ utilization * (1/2 * Χρόνος_{εξυπηρέτη} * (1+C)).$$

Κατόπιν:

$$Χρόνος_{ουράς} * (1 - server\ utilization) = (1/2) * Χρόνος_{εξυπηρέτη} * (1+C) * server\ utilization$$

Και τελικά έχουμε:

$$Χρόνος_{ουράς} = Χρόνος\ (εξυπηρέτη) * (1+C) * server\ utilization^2 * (1 - server\ utilization)$$

Ανακεφαλαιώνοντας λοιπόν...

Η Είσοδος/Εξόδος είναι το τμήμα ενός υπολογιστικού συστήματος, με το οποίο το σύστημα είναι σε θέση να ανταλλάξει δεδομένα με το περιβάλλον του, δηλαδή είτε με άλλους υπολογιστές είτε με τον άνθρωπο-χρήστη.

Η συστηματική παραμέληση της ανάπτυξης του συστήματος εισόδου/εξόδου αποτελεί τροχοπέδη για την συνολική απόδοση του συστήματος.

Η απόδοση ενός συστήματος μπορεί να μετρηθεί με τη βοήθεια τόσο του χρόνου απόκρισης , όσο και του ρυθμού διαμεταγωγής.

Τις περισσότερες φορές οι δύο παραπάνω έννοιες είναι συγκρουόμενες: μικρός χρόνος απόκρισης μπορεί να σημαίνει μεγάλο ρυθμό διαμεταγωγής και το αντίστροφο.

Η εκάστοτε εφαρμογή καθορίζει σε ποίο από τα δύο μέτρα θα δώσουμε σημασία. Υπάρχουν και εφαρμογές που απαιτούν καλό χρόνο απόκρισης και ρυθμό διαμεταγωγής ταυτόχρονα.

Ένα σύστημα είναι σε ισορροπία όταν ο *ρυθμός της εισόδου είναι ίσος με τον ρυθμό της εξόδου.* Αυτό μας οδηγεί στον Little's Law, ο οποίος συνδέει τον μέσο αριθμό των διεργασιών στο σύστημα, τον μέσο ρυθμό άφιξης των νέων διεργασιών και τον μέσο χρόνο εκτέλεσης μιας διεργασίας.