

Κεφάλαιο



Διαχείριση ΚΜΕ στα Λειτουργικά Συστήματα

Σκοπός του κεφαλαίου αυτού είναι να σου γνωρίσει τον τρόπο με τον οποίο ένα λειτουργικό σύστημα διαχειρίζεται τις διεργασίες του, με τέτοιο τρόπο ώστε να εξυπηρετούνται όλες οι ανάγκες των χρηστών του με ικανοποιητικό τρόπο.

Όταν ολοκληρώσεις το κεφάλαιο αυτό, θα μπορείς:

- ♦ Να γνωρίζεις πώς εναλλάσσονται οι διεργασίες στην ΚΜΕ από το χρονοδρομολογητή.
- ♦ Να γνωρίζεις τα διάφορα είδη και τους αλγόριθμους χρονοδρομολόγησης.
- ♦ Να εφαρμόζεις τους αλγόριθμους χρονοδρομολόγησης σε πρακτικά παραδείγματα και να υπολογίζεις διάφορους δείκτες απόδοσης.

Μαθήματα**8.1** Χρονοδρομολόγηση**8.2** Αλγόριθμοι Χρονοδρομολόγησης

Μάθημα

8.1

Χρονοδρομολόγηση

Σκοπός του μαθήματος αυτού είναι να εξηγήσει το ρόλο και τη λειτουργία της χρονοδρομολόγησης σε ένα λειτουργικό σύστημα.

Σκοπός του μαθήματος

Όταν ολοκληρώσεις το μάθημα αυτό, θα μπορείς:

- ◆ Να ορίζεις τη χρονοδρομολόγηση της ΚΜΕ
- ◆ Να απαριθμείς τα επίπεδα χρονοδρομολόγησης που υπάρχουν
- ◆ Να περιγράφεις τον κύκλο εκτέλεσης των διεργασιών

Τι θα μάθεις;

Για να εκτελούνται οι διάφορες διεργασίες και να παράγεται υπολογιστικό έργο, πρέπει πρώτα απ' όλα το ΛΣ να αποφασίζει ποια διεργασία θα εκτελείται ανά πάσα στιγμή στην ΚΜΕ του υπολογιστή.

Η ενότητα του ΛΣ που λαμβάνει διάφορες αποφάσεις σχετικά με την ανάθεση της ΚΜΕ στις διάφορες διεργασίες ονομάζεται *χρονοδρομολόγηση ΚΜΕ (CPU scheduling)*.

Όταν σχεδιάζεται ένα ΛΣ, η επιλογή των μεθόδων που αυτό θα χρησιμοποιεί για τη χρονοδρομολόγηση της ΚΜΕ είναι πολύ σημαντική για την καλή απόδοση του συστήματος. Ανάλογα με την οργάνωση του υλικού και τις υπολογιστικές ανάγκες που το σύστημα (υλικό + λογισμικό) θα καλύψει, μπορούν να ακολουθηθούν διάφορες μεθοδολογίες.

Η χρονοδρομολόγηση της ΚΜΕ αντιστοιχεί με την επιλογή από το ζαχαροπλάστη του γλυκού που παρασκευάζει κάθε φορά. Επειδή γενικά οι παραγγελίες είναι πολλές, δεν τις εκτελεί όλες ταυτόχρονα. Επιλέγει μια ομάδα από αυτές, και τις εκτελεί μαζί· κάθε φορά που μια από αυτές ολοκληρώνεται, την αντικαθιστά με μια νέα. Ανάλογα με τις ανάγκες των πελατών, αλλά και τις δυνατότητες που του παρέχει η κουζίνα του, ο ζαχαροπλάστης πρέπει να λάβει διάφορες αποφάσεις:

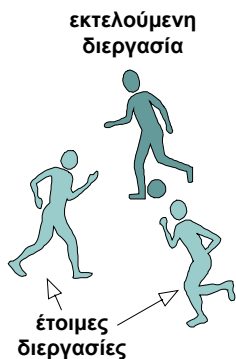
- ◆ Πόσος χρόνος θα διατίθεται για κάθε συνταγή μέχρι αυτή να διακοπεί για να πάρει κάποια άλλη τη θέση της;
- ◆ Ποια συνταγή από αυτές που μπορούν να συνεχιστούν θα επιλέξει κάθε φορά ο ζαχαροπλάστης για να συνεχίσει;
- ◆ Ποιες από όλες τις διαθέσιμες παραγγελίες για γλυκά θα επιλέξει για να αρχίσει να παρασκευάζει ταυτόχρονα;

- ◇ Πότε πρέπει να αλλάξει την ομάδα των συνταγών που εκτελεί ταυτόχρονα, έτσι ώστε να πετύχει την καλύτερη δυνατή εκμετάλλευση του χρόνου και των σκευών του;

Βλέπουμε λοιπόν ότι η χρονοδρομολόγηση της ΚΜΕ γίνεται σε δυο επίπεδα:

1. Στο πρώτο επίπεδο, οι διεργασίες που πρέπει να εκτελεστούν βρίσκονται αποθηκευμένες όλες μαζί σε κάποια μονάδα μαζικής αποθήκευσης (π.χ. το σκληρό δίσκο). Από αυτές επιλέγεται μια ομάδα διεργασιών, οι οποίες θα εναλλάσσονται εκτελούμενες στην ΚΜΕ.
2. Στο δεύτερο επίπεδο βρίσκονται μόνο οι διεργασίες οι οποίες έχουν επιλεγεί από το πρώτο. Οι διεργασίες αυτές βρίσκονται πλέον στην κύρια μνήμη, και είναι καταχωρημένες στη λίστα έτοιμων διεργασιών. Η χρονοδρομολόγηση πλέον περιλαμβάνει την επιλογή σε τακτά χρονικά διαστήματα μιας νέας διεργασίας για να εκτελεστεί από την ΚΜΕ.

Τα δυο επίπεδα χρονοδρομολόγησης της ΚΜΕ μπορούν να αντιστοιχιστούν με έναν ποδοσφαιρικό αγώνα. Αρχικά υπάρχουν δυο δεκαεξάδες παικτών, από τις οποίες όμως οι προπονητές επιλέγουν δυο ενδεκάδες για να αγωνιστούν. Η επιλογή αυτή αντιστοιχεί στο πρώτο στάδιο χρονοδρομολόγησης, όπου κάποιες διεργασίες επιλέγονται για να παίξουν ενεργό ρόλο, ενώ οι υπόλοιπες παραμένουν «ανεργές». Το δεύτερο στάδιο χρονοδρομολόγησης αντιστοιχεί στην εναλλαγή της μπάλας μεταξύ των παικτών (όπου η μπάλα παριστάνει την ΚΜΕ): μόνο ένας παίκτης μπορεί να έχει στην κατοχή του τη μπάλα, και μάλιστα ένας από τους εικοσιδύο ενεργούς παίκτες που συμμετέχουν στο παιχνίδι. Οι αναπληρωματικοί πρέπει να μπουν πρώτα στον αγώνα για να αποκτήσουν τη δυνατότητα να πάρουν τη μπάλα.



Είδη χρονοδρομολόγησης

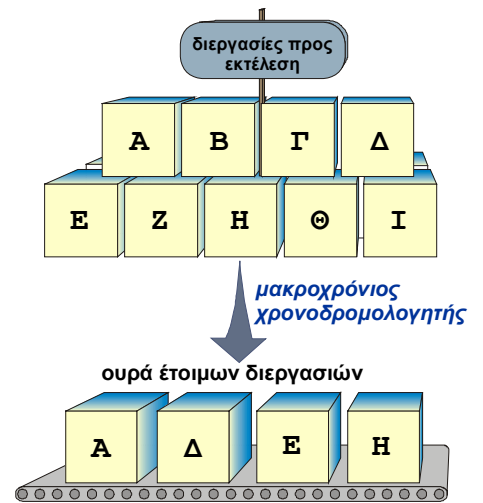
Τα δυο αυτά επίπεδα χρονοδρομολόγησης γίνονται ανεξάρτητα· υπάρχουν λοιπόν στο ΛΣ δυο χρονοδρομολογητές (schedulers), ένας για κάθε επίπεδο.

- ✦ Ο *μακροχρόνιος χρονοδρομολογητής* ή *χρονοδρομολογητής εργασιών* (long term scheduler ή job scheduler), καθορίζει ποιες διεργασίες (προγράμματα έτοιμα να εκτελεστούν) εισέρχονται στο σύστημα για εκτέλεση. Από όλες τις διεργασίες που έχουν υποβληθεί από τους χρήστες για εκτέλεση, και βρίσκονται σε κάποια μονάδα μαζικής αποθήκευσης, ο μακροχρόνιος χρονοδρομολογητής επιλέγει ορισμένες και τις φορτώνει στη μνήμη για επεξεργασία.
- ✦ Ο *βραχυχρόνιος χρονοδρομολογητής* ή *χρονοδρομολογητής ΚΜΕ* (short term scheduler ή CPU scheduler), επιλέγει για να εκτελέσει μια από τις διεργασίες οι οποίες είναι φορτωμένες στη μνήμη και είναι έτοιμες να χρησιμοποιήσουν την ΚΜΕ. Οι διεργασίες αναμένουν στη λίστα έτοιμων διεργασιών (τη λίστα του ΛΣ με τις διεργασίες που μπορούν να εκτελεστούν από την ΚΜΕ), από όπου κάνει την επιλογή του ο βραχυχρόνιος χρονοδρομολογητής.

Η κύρια διαφοροποίηση μεταξύ των δυο αυτών χρονοδρομολογητών είναι η συχνότητα με την οποία εκτελούνται. Ο μακροχρόνιος χρονοδρομολογητής εκτελείται σε αραιά χρονικά διαστήματα, ενώ αντίθετα ο βραχυχρόνιος εκτελείται πολύ συχνά

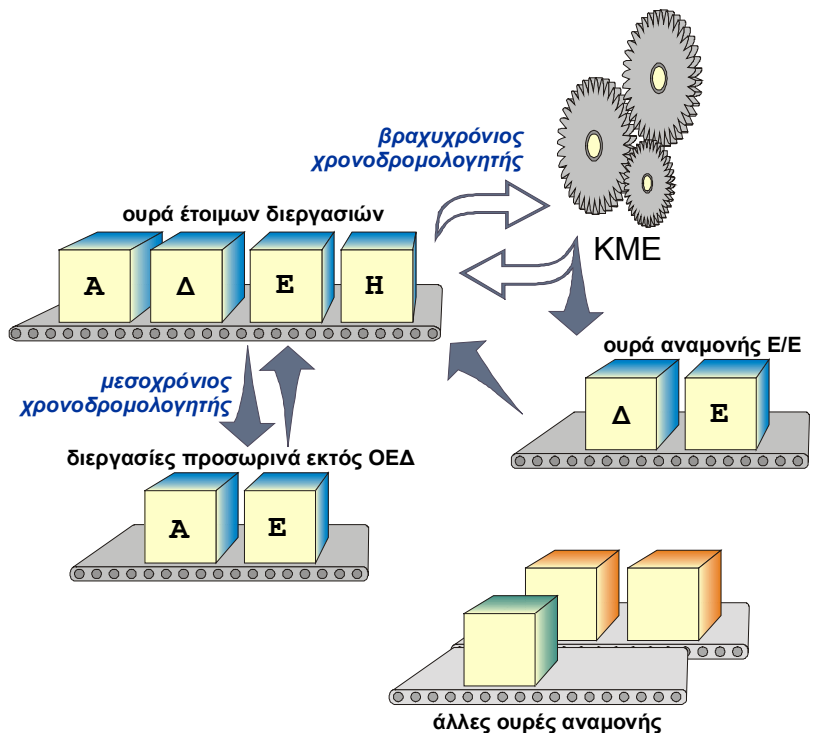
(π.χ. κάθε 10 msec). Για να μην επιβαρύνει λοιπόν το σύστημα, ο βραχυχρόνιος χρονοδρομολογητής πρέπει να είναι πολύ γρήγορος στην επιλογή της διεργασίας που θα εκτελεστεί.

Εκτός από τους δυο αυτούς χρονοδρομολογητές, πολλές φορές υπάρχει και ένα ενδιάμεσο επίπεδο χρονοδρομολόγησης, το οποίο αναλαμβάνει ο *μεσοχρόνιος χρονοδρομολογητής* (mid term scheduler). Αυτός ελέγχει σε τακτά χρονικά διαστήματα τη λίστα έτοιμων διεργασιών, και επιλέγει μερικές φορές από αυτή κάποιες διεργασίες οι οποίες διακόπτονται προσωρινά. Σε άλλες περιπτώσεις ενεργοποιεί ξανά κάποιες διεργασίες που είχε διακόψει παλαιότερα και τις επιστρέφει στη λίστα έτοιμων διεργασιών για εκτέλεση. Με τον τρόπο αυτό, επιδιώκεται να είναι ομοιόμορφα κατανομημένο το υπολογιστικό φορτίο κατά τη λειτουργία του υπολογιστή και ο χρόνος απόκρισης να μην ξεφεύγει από κάποια όρια αποδεκτής λειτουργίας.



Στα σχήματα παριστάνονται τα διάφορα στάδια της χρονοδρομολόγησης από τα οποία περνά μια διεργασία. Η είσοδος της στην ουρά των έτοιμων διεργασιών γίνεται από το μακροχρόνιο χρονοδρομολογητή, και η μετακίνησή της από και προς την ΚΜΕ από το βραχυχρόνιο. Ο μεσοχρόνιος χρονοδρομολογητής μετακινεί διεργασίες προς και από μια άλλη ουρά διεργασιών, οι οποίες βρίσκονται προσωρινά εκτός του ανταγωνισμού για χρήση της ΚΜΕ.

Όταν μια διεργασία αφήνει την ΚΜΕ μπορεί να επιστρέψει στη λίστα έτοιμων διεργασιών, αλλά υπάρχει και η περίπτωση να βρεθεί σε μια ουρά αναμονής, π.χ. για είσοδο/έξοδο δεδομένων από κάποια συσκευή. Η *είσοδος/έξοδος* -αλλιώς E/E - *δεδομένων* (data input/output, I/O) περιλαμβάνει λειτουργίες όπως ανάγνωση από το σκληρό δίσκο, αποστολή δεδομένων στον εκτυπωτή μέσω της παράλληλης θύρας, αποστολή δεδομένων από την κάρτα δικτύου. Γενικά η διεκπεραίωση E/E είναι πολύ πιο αργή από την εκτέλεση εντολών της ΚΜΕ· κατά τη διάρκεια μιας μόνο ανάγνωσης από το σκληρό δίσκο μπορούν να εκτελεστούν χιλιάδες εντολές από την ΚΜΕ. Όσο λοιπόν μια διεργασία αναμένει την ολοκλήρωση μιας λειτουργίας E/E μπορεί να παραχωρήσει την ΚΜΕ σε άλλες, και αυτή περιμένει σε μια ουρά την ολοκλήρωση της λειτουργίας E/E. Η παράλληλη εκτέλεση E/E από μια διεργασία και εκτέλεση στην ΚΜΕ από μια άλλη βελτιώνει την απόδοση του συστήματος.



Κύκλος εκτέλεσης διεργασιών

Η ζωή μιας διεργασίας είναι ένας συνεχής κύκλος από εκτέλεση στην ΚΜΕ και αναμονή για ολοκλήρωση μιας λειτουργίας E/E. Κάθε συνεχόμενη περίοδος εκτέλεσης στην ΚΜΕ ονομάζεται *έκρηξη ΚΜΕ* (CPU burst) ενώ μια συνεχόμενη περίοδος αναμονής για E/E ονομάζεται *έκρηξη E/E* (I/O burst). Η ζωή τής διεργασίας λοιπόν

ξεκινά με μια έκρηξη ΚΜΕ, η οποία ακολουθείται από μια έκρηξη Ε/Ε, κλπ. Ολοκληρώνεται με μια τελευταία έκρηξη ΚΜΕ κατά την οποία η διεργασία τερματίζεται.

Αν και οι διάρκειες των εκρήξεων διαφέρουν πολύ μεταξύ διαφορετικών υπολογιστών ή διεργασιών, γενικά σε κάθε διεργασία παρατηρείται ένας μεγάλος αριθμός από σύντομες εκρήξεις ΚΜΕ και ένας μικρός αριθμός από πολύ μεγάλες.

Οι διεργασίες μπορούν να κατηγοριοποιηθούν ανάλογα με το συνολικό χρόνο που περνούν σε εκρήξεις του ενός ή του άλλου είδους. Έτσι έχουμε τις διεργασίες που είναι *προσανατολισμένες στην Ε/Ε (I/O bound)* και περνούν τον περισσότερο χρόνο τους σε εκρήξεις Ε/Ε, και αυτές που είναι *προσανατολισμένες στην ΚΜΕ (CPU bound)* και απασχολούν περισσότερο την ΚΜΕ.

Ο μακροχρόνιος χρονοδρομολογητής πρέπει να κάνει ιδιαίτερα προσεκτική επιλογή των διεργασιών που θα εισαγάγει στο σύστημα για εκτέλεση. Αν διαλέξει πολλές διεργασίες προσανατολισμένες σε Ε/Ε τότε η ΚΜΕ θα υποαπασχολείται. Αν επιλέξει πολλές διεργασίες προσανατολισμένες στην ΚΜΕ, τότε οι συσκευές Ε/Ε θα μένουν ανενεργές ενώ η ΚΜΕ θα απασχολείται ασταμάτητα.



Η χρονοδρομολόγηση σε ένα ΛΣ αποφασίζει ποια διεργασία θα χρησιμοποιεί ανά πάσα στιγμή την ΚΜΕ. Συνήθως σε ένα ΛΣ υπάρχει ο μακροχρόνιος χρονοδρομολογητής, ο οποίος επιλέγει μια ομάδα διεργασιών για να εκτελεστούν ταυτόχρονα, και ο βραχυχρόνιος χρονοδρομολογητής που επιλέγει μια συγκεκριμένη διεργασία και της αναθέτει την ΚΜΕ. Ανάμεσα σε αυτούς τους δυο μπορεί να παρεμβληθεί και ο μεσοχρόνιος χρονοδρομολογητής. Μια διεργασία μπορεί να εκτελείται στην ΚΜΕ ή να περιμένει για κάποια λειτουργία Ε/Ε. Οι χρονοδρομολογητές πρέπει να εξασφαλίζουν ότι θα υπάρχει ισορροπία ανάμεσα στις διεργασίες που απαιτούν πολύ χρόνο ΚΜΕ και εκείνες που απαιτούν πολύ χρόνο Ε/Ε έτσι ώστε το σύστημα να έχει καλή απόδοση και χρησιμοποίηση τόσο της ΚΜΕ όσο και των συσκευών Ε/Ε.



<i>Βραχυχρόνιος Χρονοδρομολογητής</i>	<i>Short Term Scheduler</i>
<i>Διεργασία Προσανατολισμένη σε Ε/Ε</i>	<i>I/O Bound Process</i>
<i>Διεργασία Προσανατολισμένη στην ΚΜΕ</i>	<i>CPU Bound Process</i>
<i>Είσοδος/Εξόδος Δεδομένων - Ε/Ε</i>	<i>Data Input/Output - I/O</i>
<i>Έκρηξη Ε/Ε</i>	<i>I/O Burst</i>
<i>Έκρηξη ΚΜΕ</i>	<i>CPU Burst</i>
<i>Μακροχρόνιος Χρονοδρομολογητής</i>	<i>Long Term Scheduler</i>
<i>Μεσοχρόνιος Χρονοδρομολογητής</i>	<i>Mid Term Scheduler</i>
<i>Χρονοδρομολόγηση ΚΜΕ</i>	<i>CPU Scheduling</i>
<i>Χρονοδρομολογητής</i>	<i>Scheduler</i>
<i>Χρονοδρομολογητής Εργασιών</i>	<i>Job Scheduler</i>
<i>Χρονοδρομολογητής ΚΜΕ</i>	<i>CPU Scheduler</i>

Ερωτήσεις

- ? Ποιο είναι το έργο του χρονοδρομολογητή σε ένα ΛΣ; Νομίζεις ότι είναι σημαντικό;
- ? Πόσα και ποια επίπεδα χρονοδρομολόγησης υπάρχουν; Είναι όλα τους απαραίτητα;
- ? Τι είναι η «έκρηξη ΚΜΕ»; Πόσες εκρήξεις ΚΜΕ μπορεί να έχει μία διεργασία;

Μάθημα

8.2

Αλγόριθμοι Χρονοδρομολόγησης

Σκοπός του μαθήματος αυτού είναι να περιγράψει το ρόλο και την αξιολόγηση των αλγορίθμων χρονοδρομολόγησης, και να παρουσιάσει τους κυριότερους τέτοιους αλγορίθμους.

Σκοπός του μαθήματος

Όταν ολοκληρώσεις το μάθημα αυτό, θα μπορείς:

- ◆ Να εξηγείς τη λειτουργία των αλγορίθμων χρονοδρομολόγησης
- ◆ Να περιγράφεις πώς αξιολογούμε τους αλγορίθμους χρονοδρομολόγησης
- ◆ Να ορίζεις τι είναι διακοπτός και μη διακοπτός αλγόριθμος
- ◆ Να απαριθμείς τους κυριότερους μη διακοπτούς αλγόριθμους χρονοδρομολόγησης
- ◆ Να απαριθμείς τους κυριότερους διακοπτούς αλγόριθμους χρονοδρομολόγησης

Τι θα μάθεις;

Στο μάθημα 7.2 είδαμε ότι οι διεργασίες μπορούν να μοιράζονται μεταξύ τους την ΚΜΕ. Στο μάθημα αυτό θα δούμε τους τρόπους που μπορεί να γίνει αυτό, εξετάζοντας τους διάφορους αλγορίθμους χρονοδρομολόγησης.

Η μεθοδολογία την οποία χρησιμοποιεί κάθε χρονοδρομολογητής (βραχυχρόνιος, μεσοχρόνιος ή μακροχρόνιος) βασίζεται στον *αλγόριθμο χρονοδρομολόγησης* (scheduling algorithm) τον οποίο εφαρμόζει.

Ο αλγόριθμος αυτός περιγράφει με σαφήνεια τον τρόπο με τον οποίο ο χρονοδρομολογητής θα επιλέξει μια διεργασία για να την προωθήσει στην ΚΜΕ ή να την απομακρύνει από αυτή, πόσο χρόνο θα απασχολεί κάθε διεργασία την ΚΜΕ κλπ. Οι παράμετροι που λαμβάνει υπόψη του ο αλγόριθμος είναι το πλήθος των διεργασιών, η διάρκεια που αναμένεται να έχουν οι εκρήξεις ΚΜΕ και Ε/Ε για τις διεργασίες αυτές, κλπ.

Οι βασικοί στόχοι ενός αλγορίθμου χρονοδρομολόγησης είναι δυο:

- ⇒ Το υπολογιστικό σύστημα πρέπει να έχει όσο το δυνατόν καλύτερη απόδοση σύμφωνα με κάποια κριτήρια αξιολόγησης
- ⇒ Ο ίδιος ο αλγόριθμος πρέπει να λαμβάνει τις αποφάσεις του όσο πιο γρήγορα γίνεται, ιδιαίτερα όταν το πλήθος των διεργασιών είναι μεγάλο, για να προσθέτει την ελάχιστη επιβάρυνση στο σύστημα

Ο δεύτερος στόχος είναι ιδιαίτερα σημαντικός για το βραχυχρόνιο χρονοδρομολογητή, ο οποίος εκτελείται πολύ συχνά.

Κριτήρια αξιολόγησης για αλγόριθμους χρονοδρομολόγησης

Τα ιδιαίτερα χαρακτηριστικά των αλγορίθμων χρονοδρομολόγησης μπορεί να ευνοούν κάποια ομάδα διεργασιών έναντι άλλων. Για να αξιολογηθεί λοιπόν η επίδοσή τους, χρειάζονται κάποια «αντικειμενικά» κριτήρια, τα οποία μετρούν τις επιδόσεις του υπολογιστικού συστήματος. Υπάρχουν διάφορα τέτοια κριτήρια, αλλά συνήθως δεν είναι δυνατόν ένας αλγόριθμος να είναι αποδοτικός σε όλες τις περιπτώσεις. Έτσι, ανάλογα με το σκοπό για τον οποίο θα χρησιμοποιηθεί ένα υπολογιστικό σύστημα, επιλέγονται τα κριτήρια που ταιριάζουν στο σκοπό αυτό και ο αλγόριθμος χρονοδρομολόγησης που τα ικανοποιεί.

Τα πιο συνηθισμένα κριτήρια αξιολόγησης είναι:

1. **Ο βαθμός χρησιμοποίησης της ΚΜΕ (CPU utilization).**
Πρόκειται για το ποσοστό του χρόνου που η ΚΜΕ είναι απασχολημένη. Αν το υπολογιστικό σύστημα λειτουργεί για χρονικό διάστημα $T_{ολ}$ και η ΚΜΕ έχει απασχοληθεί για χρόνο $T_{απ}$, ο βαθμός χρησιμοποίησής της είναι $T_{απ} / T_{ολ}$. Οι καλοί βαθμοί χρησιμοποίησης είναι κοντά στο 1 (όπου η ΚΜΕ είναι συνεχώς απασχολημένη).
2. **Η ρυθμαπόδοση (throughput).**
Είναι το πλήθος των εργασιών που ολοκληρώνονται στη μονάδα του χρόνου. Αν π.χ. μέσα σε μια ώρα ολοκληρωθούν 1800 εργασίες, τότε η ρυθμαπόδοση του συστήματος είναι 0,5 εργασίες/sec. Όσο πιο μεγάλη είναι η ρυθμαπόδοση, τόσο καλύτερη γίνεται η απόδοση του συστήματος.
3. **Ο χρόνος ανακύκλωσης (turnaround time).**
Είναι ο συνολικός χρόνος που απαιτείται για την πλήρη εκτέλεση μιας εργασίας, από τη στιγμή που αυτή υποβάλλεται στο σύστημα μέχρι τη στιγμή που ολοκληρώνεται. Αυτός ο χρόνος περιλαμβάνει την αρχική αναμονή της εργασίας μέχρι να επιλεγεί για φόρτωση στην κύρια μνήμη, το χρόνο εκτέλεσης στην ΚΜΕ, το χρόνο E/E και το χρόνο αναμονής σε διάφορες ουρές του συστήματος.
4. **Ο χρόνος αναμονής (waiting time).**
Πρόκειται για το χρόνο που πέρασε η διεργασία στη λίστα έτοιμων διεργασιών αναμένοντας την εκτέλεσή της. Ο χρόνος αναμονής είναι ένας «καλύτερος» δείκτης της καθυστέρησης που επιβάλλει το σύστημα σε μια διεργασία από το χρόνο ανακύκλωσης, γιατί ο δεύτερος περιέχει και το χρόνο που εκτελείται η διεργασία στην ΚΜΕ ή εκτελεί λειτουργίες E/E, ο οποίος είναι σταθερός για κάθε διεργασία και δεν εξαρτάται από το ΛΣ.
5. **Ο χρόνος απόκρισης (response time).**
Σε διαλογικά συστήματα ενδιαφέρει πολλές φορές ο χρόνος που απαιτείται μέχρι το σύστημα να δώσει την πρώτη έξοδο-απόκριση για μια διεργασία, που ονομάζεται χρόνος απόκρισης. Αυτός διαφέρει από το χρόνο ανακύκλωσης κατά το χρονικό διάστημα που απαιτείται για την ολοκλήρωση της εξόδου μιας διεργασίας, και συνεπώς δεν εξαρτάται από την ταχύτητα της συσκευής εξόδου. Αν π.χ. μια διεργασία μετά το τέλος των υπολογισμών της εκτυπώνει τα αποτελέσματα, ο χρόνος απόκρισης μετριέται μέχρι τη στιγμή που αρχίζει η εκτύπωση· ο χρόνος της εκτύπωσης δεν συνυπολογίζεται.

Κατηγορίες αλγορίθμων χρονοδρομολόγησης

Οι αλγόριθμοι χρονοδρομολόγησης μπορούν να ταξινομηθούν σε δυο μεγάλες κατηγορίες:

- ♦ **Μη διακοπτοί (non preemptive) αλγόριθμοι:** Μια διεργασία που επιλέγεται να πάρει τον έλεγχο της ΚΜΕ, τον διατηρεί έως ότου αυτή ολοκληρωθεί, ή απαιτήσει κάποια άλλη λειτουργία (π.χ. Ε/Ε).
- ♦ **Διακοπτοί (preemptive) αλγόριθμοι:** Αφού μια διεργασία χρησιμοποιήσει την ΚΜΕ εκτελούμενη για κάποιο χρονικό διάστημα (το κβάντο χρόνου), το ΛΣ τη διακόπτει, ανεξάρτητα από το αν έχει ολοκληρωθεί, για να δώσει την ΚΜΕ σε κάποια άλλη.

Σε ένα μη διακοπτό αλγόριθμο δηλαδή, κάθε διεργασία εκτελεί χωρίς διακοπή ολόκληρες τις εκρήξεις ΚΜΕ, ενώ σε ένα διακοπτό τις εκτελεί τμηματικά.

Μη διακοπτοί αλγόριθμοι χρονοδρομολόγησης

Οι πιο διαδεδομένοι μη διακοπτοί αλγόριθμοι χρονοδρομολόγησης είναι τρεις, και βασίζονται, αντίστοιχα, στη σειρά άφιξης των διεργασιών, τη διάρκειά τους και το λόγο απόκρισής τους. Θα εφαρμόσουμε κάθε έναν από τους αλγορίθμους στο ίδιο παράδειγμα για να συγκρίνουμε τις επιδόσεις τους. Ας υποθέσουμε λοιπόν ότι οι τέσσερις διεργασίες προς εξυπηρέτηση είναι οι $\delta_1, \delta_2, \delta_3$ και δ_4 . Η δ_1 φθάνει στη λίστα έτοιμων διεργασιών τη χρονική στιγμή 0 και έχει διάρκεια 12 χρονικές μονάδες, η δ_2 φθάνει τη στιγμή 6 και έχει διάρκεια 31, η δ_3 φθάνει τη στιγμή 8 και έχει διάρκεια 3 και η δ_4 φθάνει τη στιγμή 14 και έχει διάρκεια 11. Επιπλέον γίνεται η παραδοχή ότι οι διεργασίες απασχολούν μόνο την ΚΜΕ και δεν εκτελούν λειτουργίες Ε/Ε.

Διεργασία	Άφιξη	Διάρκεια
δ_1	0	12
δ_2	6	31
δ_3	8	3
δ_4	14	11

Εξυπηρέτηση με βάση τη σειρά άφιξης

Ο πιο απλός τρόπος επιλογής του επόμενου πελάτη του ζαχαροπλαστείου που θα εξυπηρετηθεί είναι να επιλέγεται αυτός που περιμένει περισσότερη ώρα. Όση ώρα το γλυκό του παρασκευάζεται οι υπόλοιποι περιμένουν τη σειρά τους. Για να βρίσκεται εύκολα ο επόμενος πελάτης που θα εξυπηρετηθεί, όποιος μπαίνει στο ζαχαροπλαστείο περιμένει στην ουρά για να έρθει η σειρά του.

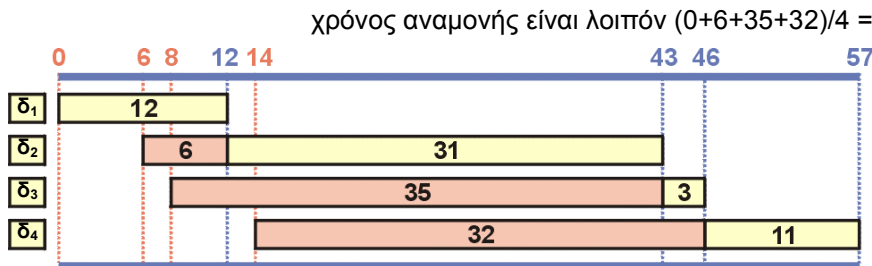
Ο αντίστοιχος αλγόριθμος χρονοδρομολόγησης ΚΜΕ ονομάζεται *αλγόριθμος με βάση τη σειρά άφιξης* (First Come First Served - FCFS). Η υλοποίηση του γίνεται πολύ εύκολα· η λίστα έτοιμων διεργασιών λειτουργεί όπως μια ουρά αναμονής.

Κάθε νέα διεργασία που ετοιμάζεται να εκτελέσει την επόμενη έκρηξή της στην ΚΜΕ εισάγεται στο τέλος της λίστας έτοιμων διεργασιών. Όταν έρθει η ώρα να εισαχθεί στην ΚΜΕ μια νέα διεργασία, επιλέγεται αυτή που είναι στην αρχή της λίστας, προωθείται στην ΚΜΕ και εκεί εκτελεί ολόκληρη την έκρηξή της στην ΚΜΕ.

Η επίδοση του αλγορίθμου αυτού αν κριθεί με βάση το *χρόνο ανακύκλωσης* (turnaround time) είναι χαμηλή, γιατί εξαναγκάζει ακόμα και διεργασίες με πολύ σύντομες εκρήξεις ΚΜΕ να αναμένουν για την εκτέλεση άλλων. Παρόμοιες επιδόσεις πετυχαίνει και με βάση τα υπόλοιπα κριτήρια αξιολόγησης που αναφέρθηκαν στο μάθημα αυτό· η χρησιμοποίηση π.χ. της ΚΜΕ μπορεί να είναι χαμηλή.

Το βασικό πλεονέκτημα της εξυπηρέτησης με βάση τη σειρά άφιξης είναι η απλότητά της. Ο αλγόριθμος είναι πάρα πολύ απλός να προγραμματιστεί και καταναλώνει ελάχιστο χρόνο για την εκτέλεσή του.

Στο σχήμα φαίνεται το σχεδιάγραμμα εκτέλεσης των τεσσάρων διεργασιών του παραδείγματος. Οι χρόνοι αναμονής είναι 0 για την πρώτη διεργασία, $12-6=6$ για τη δεύτερη, $12+31-8=35$ για την τρίτη και $12+31+3-14=32$ για την τέταρτη. Ο μέσος



χρόνος αναμονής είναι λοιπόν $(0+6+35+32)/4 = 18,25$. Παρατηρούμε επίσης ότι η διεργασία δ_3 , η οποία είχε διάρκεια μόνο 3, εξαναγκάστηκε να περιμένει για 35 χρονικές μονάδες για να εκτελεστεί, είχε δηλαδή τεράστια αναμονή σε σχέση με τη διάρκειά της.

Οι χρόνοι απόκρισης (χρόνος αναμονής + χρόνος εκτέλεσης) είναι: 12 για τη δ_1 , 37 για τη δ_2 , 38 για τη δ_3 και 43 για τη δ_4 . Έτσι ο μέσος χρόνος απόκρισης είναι $(12+37+38+43)/4 = 32,5$.

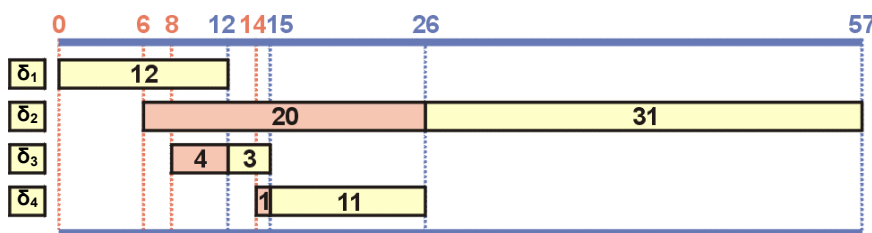
Εξυπηρέτηση με βάση τη διάρκεια

Καλό είναι οι πελάτες που επιθυμούν την παρασκευή κάποιων απλών γλυκών που απαιτούν λίγο χρόνο, να μην περιμένουν πολλή ώρα μέχρι να εκτελεστούν κάποιες άλλες, χρονοβόρες, συνταγές. Αντί αυτού, είναι προτιμότερο να περιμένουν πολλή ώρα αυτοί που έχουν παραγγείλει μια χρονοβόρα συνταγή. Γενικά, όταν προηγούνται αυτοί που θα εξυπηρετηθούν σύντομα, κανείς δεν περιμένει άδικα για ένα υπερβολικό χρονικό διάστημα*.

Ο αλγόριθμος εξυπηρέτησης με βάση τη διάρκεια (Shortest Job First - SJF), κατατάσσει τις διεργασίες κατά αύξουσα σειρά διάρκειας. Πρώτα εκτελείται η συντομότερη διεργασία, μετά αυτή με την αμέσως μεγαλύτερη διάρκεια κ.ο.κ. Τελευταία εκτελείται αυτή με τη μεγαλύτερη διάρκεια.

Μια διεργασία που εισάγεται στη λίστα έτοιμων διεργασιών, όταν μια άλλη ήδη εκτελείται, θα τοποθετηθεί στο κατάλληλο σημείο ανάλογα με τη διάρκεια της έκρηξής της στην ΚΜΕ, και θα εκτελεστεί πριν από κάποια άλλη που είχε εισαχθεί στη λίστα νωρίτερα αλλά είχε μεγαλύτερη διάρκεια.

Όσον αφορά το χρόνο που οι διεργασίες βρίσκονται σε αναμονή, ο αλγόριθμος με βάση τη διάρκεια είναι βέλτιστος, δηλαδή κανένας άλλος αλγόριθμος δεν μπορεί να επιτύχει καλύτερο μέσο χρόνο αναμονής.



Η εκτέλεση των διεργασιών του παραδείγματος με τον αλγόριθμο εξυπηρέτησης με βάση τη διάρκεια φαίνεται στο σχήμα. Η σειρά εκτέλεσης των διεργασιών είναι δ_1 , δ_3 , δ_4 , δ_2 . Η δ_1 επειδή είναι μόνη της πρώτη δεν περιμένει καθόλου, η δ_2

περιμένει για $(12+3+11)-6 = 20$, η δ_3 για $12-8 = 4$ και η δ_4 για $(12+3)-12 = 3$. Ο μέσος χρόνος αναμονής είναι λοιπόν $(0+20+4+1)/4 = 6,25$ · πολύ μικρότερος από τον αντίστοιχο για την εξυπηρέτηση με βάση τη σειρά άφιξης, και ο μικρότερος δυνατός.

Οι χρόνοι απόκρισης είναι: 12 για τη δ_1 , 51 για τη δ_2 , 7 για τη δ_3 και 12 για τη δ_4 . Ο μέσος χρόνος απόκρισης είναι $(12+51+7+12)/4 = 20,5$, πολύ καλύτερος επίσης από την εξυπηρέτηση με βάση τη σειρά άφιξης.

* Η αρχή αυτή εφαρμόζεται σε ορισμένα μεγάλα καταστήματα όπου οι πελάτες που αγοράζουν μικρό αριθμό αντικειμένων εξυπηρετούνται σε ιδιαίτερα ταμεία.

Η δυσκολία στην υλοποίηση του αλγορίθμου αυτού είναι η εκτίμηση της διάρκειας για την έκρηξη ΚΜΕ κάθε διεργασίας. Στην περίπτωση του μακροχρόνιου χρονοδρομολογητή, οι χρήστες μπορούν να έχουν θέσει εκ των προτέρων ανώτατα όρια στο χρόνο εκτέλεσης του προγράμματός τους, με βάση τα οποία να γίνεται η επιλογή της συντομότερης διεργασίας. Όμως στο βραχυχρόνιο χρονοδρομολογητή δεν είναι εύκολο να υπολογιστεί εκ των προτέρων η επόμενη έκρηξη ΚΜΕ για μια διεργασία· μπορεί βέβαια να εκτιμηθεί στατιστικά με βάση τις προηγούμενες εκρήξεις ΚΜΕ, αλλά η εκτίμηση αυτή δεν είναι ακριβής και επιπλέον απαιτεί υπολογιστικό χρόνο.

Εξυπηρέτηση με βάση το λόγο απόκρισης

Στους δυο αλγορίθμους που παρουσιάστηκαν οι παράμετροι που λαμβάνονται υπόψη είναι δυο:

- ♦ Στην εξυπηρέτηση με βάση τη σειρά άφιξης, ο αλγόριθμος εξετάζει τον πραγματικό χρόνο αναμονής για κάθε μια διεργασία και επιλέγει αυτή με το μεγαλύτερο χρόνο αναμονής.
- ♦ Στην εξυπηρέτηση με βάση τη διάρκεια, ο αλγόριθμος ελέγχει τη διάρκεια της επόμενης έκρηξης ΚΜΕ και επιλέγει τη διεργασία με τη μικρότερη έκρηξη.

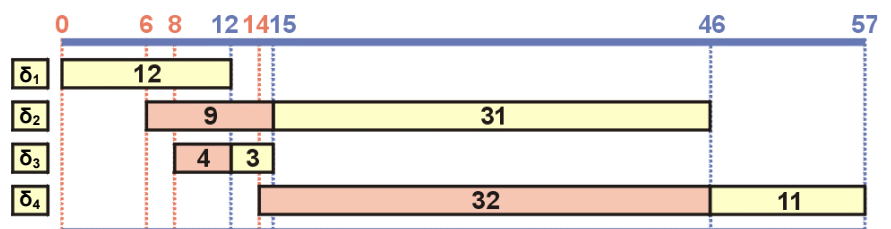
Βλέπουμε λοιπόν ότι οι δυο αλγόριθμοι είναι μάλλον μονόπλευροι ως προς τη χρονοδρομολόγηση ΚΜΕ. Μια μέση λύση θα προσπαθούσε να λάβει υπόψη και τις δυο αυτές παραμέτρους, συνδυάζοντάς τις σε μια. Αυτή η παράμετρος είναι ο *λόγος απόκρισης* (response ratio) και είναι:

$$\text{Λόγος Απόκρισης} = \frac{\text{Χρόνος Απόκρισης}}{\text{Χρόνος Εκτέλεσης}} = \frac{\text{Χρόνος Εκτέλεσης} + \text{Χρόνος Αναμονής}}{\text{Χρόνος Εκτέλεσης}} = 1 + \frac{\text{Χρόνος Αναμονής}}{\text{Χρόνος Εκτέλεσης}}$$

Ο αλγόριθμος που βασίζεται στο λόγο απόκρισης ονομάζεται αλγόριθμος *εξυπηρέτησης με βάση το λόγο απόκρισης* (Highest Response Ratio First, HRF), και επιλέγει για εκτέλεση εκείνη τη διεργασία που έχει το μεγαλύτερο λόγο απόκρισης.

Στο παράδειγμά μας, η διεργασία δ₁ που φθάνει πρώτη για εκτέλεση καταλαμβάνει την ΚΜΕ. Όσο αυτή εκτελείται, φθάνουν οι δ₂ και δ₃. Όταν η δ₁ ολοκληρωθεί, η δ₂ θα έχει χρόνο αναμονής 6 και η δ₃ θα έχει 4. Ο λόγος απόκρισης για τη δ₂ θα είναι λοιπόν 1+6/31 ≅ 1,19 και για τη δ₃ θα είναι 1+4/3 ≅ 2,33. Επιλέγεται λοιπόν η δ₃ για εκτέλεση, κατά τη διάρκεια της οποίας φθάνει η δ₄. Όταν και η δ₃ τελειώσει τη χρονική στιγμή 15, η δ₂ θα έχει περάσει 15-6 = 9 χρονικές μονάδες σε αναμονή και η δ₄ 1. Ο λόγος απόκρισης για τη δ₂ θα είναι τώρα 1+9/31 ≅ 1,29 και για τη δ₄ θα είναι 1+1/11 ≅ 1,09. Αυτή τη φορά λοιπόν επιλέγεται η δ₂ για εκτέλεση και τελευταία εκτελείται η δ₄. Η ακολουθία εκτέλεσης φαίνεται στο σχήμα.

Ο χρόνος αναμονής για τη δ₂ είναι (12+3)-6 = 9, για τη δ₃ είναι 12-8 = 4 και για τη δ₄ είναι (12+3+31)-14 = 32. Έτσι ο μέσος χρόνος αναμονής είναι (0+9+4+32)/4 = 11,25· μεγαλύτερος από το χρόνο της



εξυπηρέτησης με βάση τη διάρκεια, αλλά καλύτερος από την εξυπηρέτηση με βάση τη σειρά άφιξης.

Οι χρόνοι απόκρισης εδώ είναι 12 για τη δ_1 , 40 για τη δ_2 , 7 για τη δ_3 και 43 για τη δ_4 . Έτσι ο μέσος χρόνος απόκρισης γίνεται $(12+40+7+43)/4 = 25,5$.

Συνοπτικά, τα αποτελέσματα για τους τρεις αλγόριθμους φαίνονται στον πίνακα που ακολουθεί. Καλύτερος από τους τρεις για το συγκεκριμένο παράδειγμα είναι ο αλγόριθμος εξυπηρέτησης με βάση τη διάρκεια και για τα δυο κριτήρια αξιολόγησης.

	Με βάση τη σειρά άφιξης (FCFS)	Με βάση τη διάρκεια (SJF)	Με βάση το λόγο απόκρισης (HRF)
Μέσος χρόνος αναμονής	18,25	6,25	11,25
Μέσος χρόνος απόκρισης	32,50	20,50	24,50

Εξυπηρέτηση με βάση την προτεραιότητα

Οι τρεις μη διακοπτοί αλγόριθμοι που περιγράφηκαν στις προηγούμενες παραγράφους αποτελούν ειδική περίπτωση ενός γενικότερου αλγόριθμου, ο οποίος δρομολογεί διεργασίες με βάση την προτεραιότητά (priority) τους.

Στον αλγόριθμο αυτό, σε κάθε διεργασία αντιστοιχίζεται μια αριθμητική τιμή, που είναι η προτεραιότητά της, και η διεργασία με τη μεγαλύτερη προτεραιότητα επιλέγεται κάθε φορά για εκτέλεση. Όταν δυο ή περισσότερες διεργασίες έχουν την ίδια προτεραιότητα, τότε επιλέγεται πρώτα η παλαιότερη, δηλαδή χρονοδρομολογούνται με βάση τη σειρά άφιξης.

Ας δούμε πώς εκφράζονται οι τρεις προηγούμενοι αλγόριθμοι με βάση την προτεραιότητα:

- ⇒ Στην εξυπηρέτηση με βάση τη σειρά άφιξης, όλες οι διεργασίες έχουν την ίδια προτεραιότητα, π.χ. 1. Έτσι κάθε φορά επιλέγεται για εκτέλεση η παλαιότερη.
- ⇒ Στην εξυπηρέτηση με βάση τη διάρκεια, θέλουμε να εκτελείται πρώτη, να έχει δηλαδή τη *μεγαλύτερη* προτεραιότητα, η διεργασία με τη *μικρότερη* έκρηξη ΚΜΕ. Πρέπει δηλαδή η προτεραιότητα να είναι μια ποσότητα που μεγαλώνει όταν η έκρηξη ΚΜΕ μικραίνει. Η ποσότητα $1/(\text{έκρηξη ΚΜΕ})$ έχει αυτή την ιδιότητα, αφού είναι αντιστρόφως ανάλογη της έκρηξης ΚΜΕ και επιλέγεται για να παραστήσει την προτεραιότητα.
- ⇒ Στην εξυπηρέτηση με βάση το λόγο απόκρισης, θέλουμε να έχει τη μεγαλύτερη προτεραιότητα η διεργασία με το μεγαλύτερο λόγο απόκρισης. Έτσι πολύ απλά μπορούμε να επιλέξουμε το λόγο απόκρισης για να παίξει το ρόλο της προτεραιότητας.

Στους δυο πρώτους αλγόριθμους, η προτεραιότητα μιας διεργασίας καθορίζεται μια φορά, κατά την είσοδό της στη λίστα έτοιμων διεργασιών και δεν αλλάζει ξανά. Η προτεραιότητα που δεν αλλάζει κατά τη διάρκεια της ζωής μιας διεργασίας ονομάζεται *στατική προτεραιότητα* (static priority). Αντίθετα, στον αλγόριθμο εξυπηρέτησης με βάση το λόγο απόκρισης, οι προτεραιότητες (δηλαδή οι λόγοι απόκρισης) των διεργασιών υπολογίζονται κάθε φορά που πρέπει να επιλεγεί μια διεργασία για εκτέλεση. Η προτεραιότητα που αλλάζει κατά τη ζωή μιας διεργασίας καλείται *δυναμική προτεραιότητα* (dynamic priority).

Ένα πολύ ουσιαστικό πρόβλημα σε ορισμένους αλγόριθμους χρονοδρομολόγησης είναι ότι κάποια διεργασία με αρκετά μικρή προτεραιότητα μπορεί να αναμένει για την εκτέλεσή της επ' αόριστο. Κυρίως όταν οι προτεραιότητες είναι στατικές (όπως στην

εξυπηρέτηση με βάση τη διάρκεια), σε ένα υπερφορτωμένο σύστημα, μια διεργασία με χαμηλή προτεραιότητα μπορεί πρακτικά να είναι αποκλεισμένη από την ΚΜΕ γιατί συνεχώς την υπερπηδούν άλλες, υψηλότερης προτεραιότητας.

Μια λύση που μπορεί να δοθεί στο πρόβλημα αυτό είναι η τεχνική της *γήρανσης* (aging), η οποία αυξάνει την προτεραιότητα μιας διεργασίας με την πάροδο του χρόνου, ώστε κάποια στιγμή να εκτελεστεί. Αυτή την τεχνική εφαρμόζει ουσιαστικά η εξυπηρέτηση με βάση το λόγο απόκρισης: όσο περνά ο χρόνος και μια διεργασία δεν εκτελείται, ο λόγος απόκρισής της (άρα και η προτεραιότητά της) αυξάνει.

Διακοπτοί αλγόριθμοι χρονοδρομολόγησης

Στους διακοπτούς αλγόριθμους μια διεργασία δεν καταλαμβάνει την ΚΜΕ για όλη τη διάρκεια της έκρηξης ΚΜΕ. Το ΛΣ μπορεί να τη διακόψει και να παραχωρήσει την ΚΜΕ σε κάποια άλλη διεργασία, η οποία (όπως κρίνει το ΛΣ) πρέπει να προηγηθεί στην εκτέλεσή της.

Την ώρα που ο ζαχαροπλάστης (ΚΜΕ) εκτελεί ένα στάδιο μιας συνταγής (έκρηξη ΚΜΕ), μπαίνει στο κατάστημα ένας πελάτης που βιάζεται πολύ. Οι οδηγίες (ΛΣ) προς το ζαχαροπλάστη για την εξυπηρέτηση των πελατών προβλέπουν ότι κάθε φορά που ένας νέος πελάτης έρχεται, εξετάζεται η συνταγή και οι ανάγκες του. Αν ικανοποιεί κάποιες προϋποθέσεις, τον εξυπηρετεί αμέσως, διακόπτοντας αυτόν που εξυπηρετούσε μέχρι τότε. Οι προϋποθέσεις που πρέπει να ικανοποιεί ο πελάτης για να του παραχωρηθεί η προτεραιότητα αποτελούν το συγκεκριμένο διακοπτό αλγόριθμο χρονοδρομολόγησης.

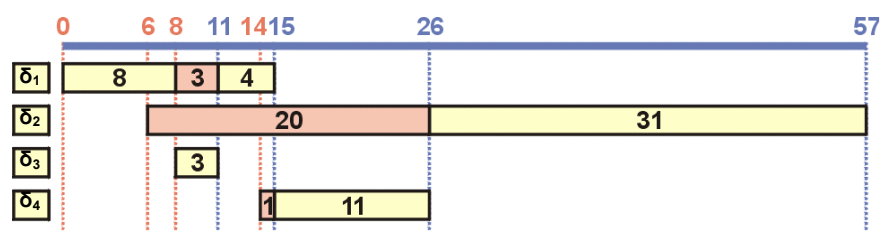
Διακοπτή εξυπηρέτηση με βάση τη διάρκεια

Κάθε φορά που ένας νέος πελάτης μπαίνει στο ζαχαροπλαστείο, ο ζαχαροπλάστης υπολογίζει το χρόνο που χρειάζεται για να τον εξυπηρετήσει, και το συγκρίνει με το χρόνο που χρειάζεται για να ολοκληρώσει τη συνταγή την οποία εκτελεί εκείνη τη στιγμή. Αν ο νέος πελάτης απαιτεί λιγότερο χρόνο, τότε σταματά τη συνταγή που εκτελούσε και ξεκινά τη δική του.

Στο διακοπτό αλγόριθμο εξυπηρέτησης *με βάση τη διάρκεια* (Preemptive Shortest Job First), κάθε φορά που μια νέα διεργασία εισέρχεται στη λίστα έτοιμων διεργασιών για εκτέλεση, ελέγχεται αν η διάρκεια της έκρηξης ΚΜΕ που θα εκτελέσει είναι μικρότερη από το χρόνο που απομένει στη διεργασία η οποία εκτελείται. Στην περίπτωση αυτή, η διεργασία που εκτελείται διακόπτεται αμέσως και η νέα διεργασία παίρνει τη θέση της. Η διεργασία που διακόπηκε τοποθετείται πάλι στη λίστα έτοιμων διεργασιών.

Όταν μια διεργασία ολοκληρώσει την εκτέλεσή της, επιλέγεται από τη λίστα έτοιμων διεργασιών εκείνη με τη συντομότερη έκρηξη ΚΜΕ και της παραχωρείται η ΚΜΕ, όπως ακριβώς και στη μη διακοπτή παραλλαγή του αλγορίθμου.

Στο παράδειγμα που χρησιμοποιήθηκε και προηγουμένως, η διεργασία δ_1 φθάνει τη χρονική στιγμή 0 και αρχίζει να εκτελείται. Τη χρονική στιγμή 6 φθάνει η διεργασία δ_2 , που θα έχει διάρκεια 31 μονάδες. Στη δ_1 απομένουν 6 μονάδες για να τελειώσει, λιγότερες από το χρόνο της δ_2 , οπότε η δ_1 συνεχίζει να



εκτελείται. Τη χρονική στιγμή 8 όμως φθάνει η διεργασία δ_3 με διάρκεια 3, ενώ στην δ_1 απομένουν 4 χρονικές μονάδες εκτέλεσης. Έτσι η δ_1 διακόπτεται και αρχίζει να εκτελείται η δ_3 , η οποία ολοκληρώνεται τη στιγμή 11. Τότε υπάρχουν δυο διεργασίες για εκτέλεση: η δ_1 , στην οποία απομένουν 4 χρονικές μονάδες για να ολοκληρωθεί, και η δ_2 η οποία απαιτεί 31 χρονικές μονάδες. Επιλέγεται βέβαια η δ_1 και αρχίζει να εκτελείται. Τη χρονική στιγμή 14, και ενώ η δ_1 εκτελείται ακόμα, φθάνει στη λίστα έτοιμων διεργασιών η δ_4 που απαιτεί 11 χρονικές μονάδες. Αφού στη δ_1 απομένει μόνο μια χρονική μονάδα το ΛΣ δεν τη διακόπτει και την αφήνει να ολοκληρωθεί. Όταν τελειώσει, οι υποψήφιες διεργασίες είναι: η δ_2 με διάρκεια 31 και η δ_4 με διάρκεια 11. Επιλέγεται η δ_4 και αφού ολοκληρωθεί εκτελείται και η δ_2 .

Η δ_1 περνά σε αναμονή για 3 χρονικές μονάδες, κατά τις οποίες εκτελείται η δ_3 (που δεν περιμένει καθόλου). Η δ_2 περιμένει για 20 χρονικές μονάδες και η δ_4 για μια. Ο μέσος χρόνος αναμονής είναι $(3+20+0+1)/4 = 6$. Αυτός είναι καλύτερος και από τον καλύτερο μη διακοπτό αλγόριθμο, τον SJF, που είχε μέσο χρόνο αναμονής 6,25.

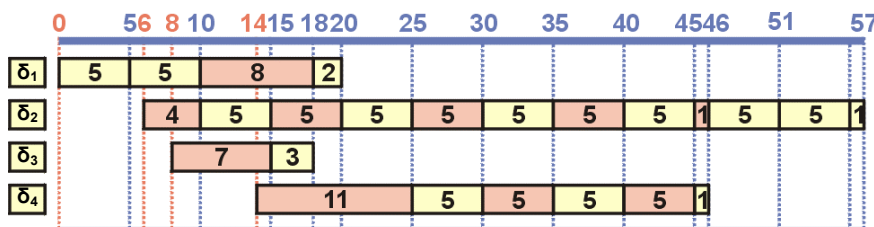
Η διεργασία δ_1 έχει χρόνο απόκρισης 15, η δ_2 51, η δ_3 3 και η δ_4 12. Έτσι, ο μέσος χρόνος απόκρισης είναι $(15+51+3+12)/4 = 20,25$, καλύτερος και αυτός από όλους τους μη διακοπτούς αλγορίθμους (αν και η διαφορά του από το μη διακοπτό SJF είναι ελάχιστη). Βέβαια στους υπολογισμούς μας δεν έχουμε λάβει υπόψη την επιβάρυνση που προκαλεί η εκτέλεση του χρονοδρομολογητή.

Χρονοδρομολόγηση κυκλικής επαναφοράς

Ο αλγόριθμος κυκλικής επαναφοράς (Round Robin - RR) έχει σχεδιασθεί ειδικά για συστήματα καταμερισμού χρόνου (time sharing) όπου δίνεται ένα μικρό ποσό χρόνου εναλλάξ σε κάθε διεργασία. Αυτό το μικρό ποσό χρόνου, όπως ήδη έχουμε πει, ονομάζεται *κβάντο χρόνου* (time quantum) και έχει συνήθως διάρκεια της τάξης των 10 msec. Η λίστα έτοιμων διεργασιών είναι η ουρά στην οποία περιμένουν οι διεργασίες. Κάθε φορά που περνά ένα κβάντο χρόνου, η ΚΜΕ διακόπτει την τρέχουσα διεργασία και την τοποθετεί στο τέλος της λίστας έτοιμων διεργασιών. Αφαιρεί στη συνέχεια τη διεργασία που βρίσκεται στην αρχή της ουράς και την παραχωρεί στην ΚΜΕ για το επόμενο κβάντο χρόνου.

Αν μια διεργασία ολοκληρώσει την εκτέλεσή της προτού τελειώσει το κβάντο χρόνου που της έχει παραχωρηθεί, απελευθερώνει οικειοθελώς την ΚΜΕ και το ΛΣ παραχωρεί την ΚΜΕ στην επόμενη έτοιμη διεργασία.

Αν υπάρχει μόνο μια διεργασία στο σύστημα, τότε της παραχωρούνται διαδοχικά κβάντα χρόνου.



Το πλάνο εκτέλεσης του παραδείγματος με τον αλγόριθμο κυκλικής επαναφοράς φαίνεται στο σχήμα, για κβάντο 5 μονάδων χρόνου. Παρατηρούμε ότι οι διεργασίες που εισάγονται στη λίστα έτοιμων διεργασιών την ώρα που

κάποια άλλη εκτελείται πρέπει να περιμένουν να έρθει η σειρά τους για εκτέλεση μετά από 1, 2 ή και 3 κβάντα χρόνου. Αυτό συμβαίνει γιατί κάθε νέα διεργασία τοποθετείται στο τέλος της λίστας έτοιμων διεργασιών, οπότε πρέπει πρώτα να εκτελεστούν όλες όσες προηγούνται (για ένα κβάντο χρόνου η κάθε μια) και μετά αυτή.

Επίσης βλέπουμε ότι, αν μια διεργασία ολοκληρωθεί πριν τελειώσει το κβάντο χρόνου της (π.χ. η δ_3 τελειώνει τη χρονική στιγμή 18 ενώ το κβάντο χρόνου τελείωνε στην 20),

η ΚΜΕ παραχωρείται αμέσως για ένα ολόκληρο κβάντο χρόνου στην επόμενη διεργασία.

Όπως φαίνεται στο σχήμα, η δ_1 περνά συνολικά 8 χρονικές μονάδες σε αναμονή, η δ_2 περνά 20 χρονικές μονάδες, η δ_3 7 και η δ_4 20. Ο μέσος χρόνος αναμονής είναι λοιπόν $(3+20+7+20)/4 = 12,5$ μονάδες χρόνου. Αυτός είναι ένας μεγάλος μέσος χρόνος αναμονής, αλλά αντισταθμίζεται από το γεγονός ότι στους χρήστες δίνεται η εντύπωση ότι οι διεργασίες τους εκτελούνται όντως παράλληλα.

Από το σχήμα επίσης μπορούμε να υπολογίσουμε τους χρόνους απόκρισης για τις τέσσερις διεργασίες: 20 για τη δ_1 , 51 για τη δ_2 , 10 για τη δ_3 και 31 για τη δ_4 . Ο μέσος χρόνος απόκρισης είναι λοιπόν $(20+51+10+31)/4 = 28$.

Η επίδοση του αλγορίθμου κυκλικής επαναφοράς επηρεάζεται σημαντικά από την επιλογή της διάρκειας που θα έχει το κβάντο χρόνου.

Αν αυτό έχει πολύ μεγάλη τιμή, τότε ο αλγόριθμος πρακτικά εκφυλίζεται στον αλγόριθμο εξυπηρέτησης με βάση τη σειρά άφιξης (FCFS), ο οποίος όπως είδαμε δεν έχει πολύ καλές επιδόσεις. Αν όμως, από την άλλη, το κβάντο χρόνου είναι πολύ μικρό, γίνονται πολλές εναλλαγές από τη μια διεργασία στην άλλη, οι οποίες επιβαρύνουν αρκετά το σύστημα. Πρέπει λοιπόν η τιμή του κβάντου χρόνου να είναι τέτοια ώστε η επιβάρυνση της εναλλαγής μεταξύ των διεργασιών να μην είναι σπουδαία. Ένας καλός κανόνας για τον προσδιορισμό ενός ικανοποιητικού κβάντου χρόνου απαιτεί να είναι μεγαλύτερο σε διάρκεια από το 80% των εκρήξεων ΚΜΕ των διαφόρων διεργασιών.

Το κυριότερο πλεονέκτημα του αλγορίθμου κυκλικής επαναφοράς είναι ότι διευκολύνει τις μικρές διεργασίες να εκτελεστούν, διότι αυτές ολοκληρώνονται σχετικά γρήγορα, χωρίς να περάσουν πολύ χρόνο σε αναμονή.



Ανακεφαλαίωση

Η μεθοδολογία που ακολουθεί το ΛΣ για να κάνει τη χρονοδρομολόγηση, δηλαδή ο αλγόριθμος χρονοδρομολόγησης που χρησιμοποιεί, μπορεί να αξιολογηθεί με διάφορα κριτήρια: το χρόνο απόκρισης, ανακύκλωσης ή αναμονής, τη ρυθμαπόδοση ή τη χρησιμοποίηση της ΚΜΕ. Οι αλγόριθμοι, ανάλογα με το αν επιτρέπουν στις διεργασίες να εκτελέσουν ακέραιες τις εκρήξεις ΚΜΕ ή όχι διακρίνονται σε μη διακοπτούς και διακοπτούς αντίστοιχα. Οι τρεις κυριότεροι μη διακοπτοί αλγόριθμοι χρονοδρομολόγησης επιλέγουν τη διεργασία που θα εκτελεστεί στην ΚΜΕ (για όλη την έκρηξη ΚΜΕ αφού είναι μη διακοπτοί) με βάση τη σειρά άφιξης των διεργασιών στη λίστα έτοιμων διεργασιών, τη διάρκεια της έκρηξης ΚΜΕ ή το λόγο απόκρισης. Και οι τρεις μπορούν να θεωρηθούν εφαρμογές της χρονοδρομολόγησης με βάση την προτεραιότητα. Οι διακοπτοί αλγόριθμοι χρονοδρομολόγησης επιτρέπουν τη διακοπή μιας διεργασίας προτού αυτή ολοκληρώσει την έκρηξη ΚΜΕ της, για να παραχωρηθεί η ΚΜΕ σε κάποια άλλη διεργασία. Οι πιο διαδεδομένοι διακοπτοί αλγόριθμοι είναι ο διακοπτός SJF (με βάση τη διάρκεια) και ο αλγόριθμος κυκλικής επαναφοράς που χρησιμοποιείται στα συστήματα καταμερισμού χρόνου.



Γλωσσάριο όρων

Αλγόριθμος Κυκλικής Επαναφοράς	Round Robin - RR
Αλγόριθμος Χρονοδρομολόγησης	Scheduling Algorithm
Βαθμός Χρησιμοποίησης της ΚΜΕ	CPU Utilization
Γήρανση	Aging
Διακοπτή Εξυπηρέτηση με Βάση τη Διάρκεια	Preemptive Shortest Job First
Διακοπτός Αλγόριθμος	Preemptive Algorithm
Δυναμική Προτεραιότητα	Dynamic Priority
Εξυπηρέτηση με Βάση τη Διάρκεια	Shortest Job First - SJF
Εξυπηρέτηση με Βάση τη Σειρά Άφιξης	First Come First Served - FCFS
Εξυπηρέτηση με Βάση το Λόγο Απόκρισης	Highest Response Ratio First - HRF
Λόγος Απόκρισης	Response Ratio
Μη Διακοπτός Αλγόριθμος	Non Preemptive Algorithm
Προτεραιότητα	Priority
Ρυθμαπόδοση	Throughput
Στατική Προτεραιότητα	Static Priority
Χρόνος Ανακύκλωσης	Turnaround Time
Χρόνος Αναμονής	Waiting Time
Χρόνος Απόκρισης	Response Time



Ερωτήσεις

- ? Ποια είναι τα βασικά κριτήρια αξιολόγησης για αλγορίθμους χρονοδρομολόγησης;
- ? Ποιες είναι οι βασικές κατηγορίες αλγορίθμων χρονοδρομολόγησης;
- ? Ανάφερε τους αλγορίθμους χρονοδρομολόγησης που υπάρχουν σε μη διακοπτή και σε διακοπτή έκδοση.
- ? Ανάφερε τους αλγορίθμους χρονοδρομολόγησης που υπάρχουν μόνο σε διακοπτή έκδοση.

Τι
μάθαμε
σε
αυτό
το
κεφάλαιο

- ♦ Η χρονοδρομολόγηση σε ένα ΛΣ αποφασίζει ποια διεργασία θα χρησιμοποιεί ανά πάσα στιγμή την ΚΜΕ.
- ♦ Η χρονοδρομολόγηση μπορεί να γίνει στο βραχυχρόνιο, το μακροχρόνιο και το μεσοχρόνιο επίπεδο.
- ♦ Οι αλγόριθμοι χρονοδρομολόγησης διακρίνονται σε διακοπτούς και μη διακοπτούς.
- ♦ Μερικοί αλγόριθμοι χρονοδρομολόγησης είναι η εξυπηρέτηση με βάση τη διάρκεια, το λόγο απόκρισης, την άφιξη, την προτεραιότητα και η κυκλική εναλλαγή.

Βιβλιογραφία – Πηγές

-  Παπακωσταντίνου Γ., Ν. Μπιλάλη, Π. Τσανάκα, Λειτουργικά Συστήματα: Αρχές Λειτουργίας, Συμμετρία, 1989.
-  Silberschatz A., Peterson J., Galvin P., Operating System Concepts, Addison - Wesley, 1991.