

Ενότητα 8 (Κεφάλαιο 8) Ιδεατή Μνήμη

Οι διαφάνειες αυτές έχουν συμπληρωματικό και επεξηγηματικό χαρακτήρα και σε καμία περίπτωση δεν υποκαθιστούν το βιβλίο

Γιάννης Α. Παπαδόπουλος
Τμήμα Πληροφορικής
Πανεπιστήμιο Κύπρου



1

Περιεχόμενα

- Τεχνικές υλοποίησης ιδεατής μνήμης.
- Στρατηγικές διαχείρισης ιδεατής μνήμης.
- Διαχείριση μνήμης στο UNIX και Solaris.
- Διαχείριση μνήμης στο Linux.
- Διαχείριση μνήμης στα Windows.
- Διαχείριση μνήμης στο Android.



2

Ορολογία

Table 8.1 Virtual Memory Terminology

Virtual memory	A storage allocation scheme in which secondary memory can be addressed as though it were part of main memory. The addresses a program may use to reference memory are distinguished from the addresses the memory system uses to identify physical storage sites, and program-generated addresses are translated automatically to the corresponding machine addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of secondary memory available and not by the actual number of main storage locations.
Virtual address	The address assigned to a location in virtual memory to allow that location to be accessed as though it were part of main memory.
Virtual address space	The virtual storage assigned to a process.
Address space	The range of memory addresses available to a process.
Real address	The address of a storage location in main memory.



3

Βασικά χαρακτηριστικά αποθήκευσης διεργασιών στη μνήμη

- Οι τεχνικές σελιδοποίησης και κατάτμησης που εξετάστηκαν έχουν δύο βασικά χαρακτηριστικά:
 - Οι αναφορές σε διευθύνσεις μνήμης είναι λογικές και αποκωδικοποιούνται δυναμικά για να δώσουν τις αντίστοιχες απόλυτες (φυσικές) τιμές τους.
 - Μία διεργασία μπορεί να απομακρυνθεί από τη μνήμη και κατόπιν να επανέλθει σε αυτήν και όχι κατ' ανάγκη στον ίδιο χώρο που χρησιμοποίησε στο παρελθόν.
 - Τα μέρη από τα οποία αποτελείται ένα πρόγραμμα (είτε είναι σελίδες είτε είναι τμήματα) δεν χρειάζεται να βρίσκονται στη μνήμη σε συνεχόμενες θέσεις. Αυτό επιτυγχάνεται με χρήση τεχνικών μετατόπισης.



4

Άμεσο επακόλουθο των προηγούμενων χαρακτηριστικών

- Αν ισχύουν τα ανωτέρω χαρακτηριστικά, τότε δεν είναι απαραίτητο όλες οι σελίδες ή τμήματα μίας διεργασίας να βρίσκονται στη μνήμη την ίδια στιγμή.
- Αν η επόμενη εντολή και τα δεδομένα που εμπλέκονται στην εκτέλεσή της βρίσκονται σε κάποια σελίδα ή πλαίσιο που είναι ήδη στη μνήμη, η εκτέλεση της διεργασίας μπορεί να συνεχίσει τουλάχιστον για κάποιο χρονικό διάστημα.
- Η παρατήρηση αυτή έγινε από τον [John Fotheringham](#) το 1961 και αποτελεί τη βάση της ανάπτυξης της τεχνικής της ιδεατής μνήμης.




5

Στρατηγική εκτέλεσης μίας διεργασίας — 1

- Αν πρέπει να φορτωθεί στη μνήμη μία νέα διεργασία, το Λ.Σ. μεταφέρει σε αυτήν μόνο ένα μέρος της.
 - Το οποίο είναι το υπάρχον στη μνήμη μέρος (resident set).
- Όσο οι αναφορές σε λογικές διευθύνσεις βρίσκονται στο υπάρχον στη μνήμη μέρος της διεργασίας (κάτι το οποίο μπορεί να ελέγξει το Λ.Σ. από τις πληροφορίες που βρίσκονται στους πίνακες σελίδων ή τμημάτων), η εκτέλεση γίνεται κανονικά.
- Αν πρέπει να γίνει αναφορά σε λογική διεύθυνση η οποία δεν ανήκει στο υπάρχον στη μνήμη μέρος της διεργασίας, τότε δημιουργείται ένα λάθος πρόσβασης μνήμης και η διεργασία τίθεται υπό αναστολή.
- Το λάθος πρόσβασης μνήμης μπορεί να είναι σφάλμα σελίδας (page fault) ή σφάλμα τμήματος (segment fault), ανάλογα με το αν χρησιμοποιείται σελιδοποίηση ή κατάτμηση.




6




Στρατηγική εκτέλεσης μίας διεργασίας — 2

- Το Λ.Σ. πρέπει τώρα να μεταφέρει στη μνήμη το μέρος εκείνο της διεργασίας στο οποίο ανήκει η λογική διεύθυνση η οποία δημιούργησε το λάθος πρόσβασης μνήμης.
- Αυτό επιτυγχάνεται με μία εντολή Ε/Ε στο δίσκο.
- Όσο διαρκεί η ενέργεια αυτή, το Λ.Σ. επιλέγει μία άλλη διεργασία για να εκτελεσθεί.
- Όταν το μέρος της αρχικής διεργασίας που βρισκόταν στο δίσκο φορτωθεί στη μνήμη, η διεργασία αυτή επανέρχεται σε κατάσταση έτοιμη για εκτέλεση.




7




Επιπτώσεις αυτής της στρατηγικής

- Περισσότερες διεργασίες μπορούν να βρίσκονται στην κύρια μνήμη.
 - Μόνο ένα μέρος της κάθε διεργασίας χρειάζεται να βρίσκεται στη μνήμη.
 - Περισσότερες πιθανότητες να υπάρχει πάντα κάποια διεργασία έτοιμη για εκτέλεση και επομένως να γίνεται πιο αποδοτική χρήση της ΚΜΕ.
- Το μέγεθος σε μνήμη μίας διεργασίας μπορεί να είναι μεγαλύτερο από τη διαθέσιμη κύρια μνήμη.
 - Λαμβάνοντας υπ' όψη ότι η περιφερειακή μνήμη είναι πολύ μεγάλη, δεν υπάρχει ουσιαστικά κανένας περιορισμός για το μέγεθος μίας διεργασίας.




8




Πραγματική και ιδεατή μνήμη

- Πραγματική μνήμη.
 - Η κύρια μνήμη.
- Ιδεατή μνήμη.
 - Η περιφερειακή μνήμη του δίσκου.
- Ο πίνακας 8.2 συγκρίνει τα χαρακτηριστικά των τεχνικών της σελιδοποίησης και της κατάτμησης με και χωρίς χρήση ιδεατής μνήμης.




9




Ατέρμονη προσκόμιση πληροφοριών στη μνήμη

- Στη συνήθη περίπτωση, σχεδόν όλη η κύρια μνήμη θα είναι καταλυμένη από διεργασίες.
- Επομένως, όταν πρέπει να προσκομισθεί στη μνήμη ένα μέρος μίας διεργασίας, κάποιο άλλο μέρος πρέπει να απομακρυνθεί από εκεί.
- Αν το μέρος που επιλεχθεί να απομακρυνθεί ζητηθεί αμέσως μετά την απομάκρυνσή του και αυτό το φαινόμενο συνεχισθεί για κάποιο χρονικό διάστημα, τότε το σύστημα θα σπαταλάει τον περισσότερο από το χρόνο του στο να μεταφέρει πληροφορίες από και προς την κύρια μνήμη αντί να εκτελεί εντολές (thrashing).
- Η αποφυγή αυτού του φαινομένου ήταν σημαντικό ερευνητικό πρόβλημα τη δεκαετία του '70 και οδήγησε στο σχεδιασμό αριθμού αλγορίθμων που βασίζονται στην προσπάθεια πρόγνωσης για τις πιθανότητες ενός μέρους διεργασίας να χρειασθεί στο άμεσο μέλλον αναλύοντας το ιστορικό της διεργασίας.




10




Τοπικότητα των αναφορών

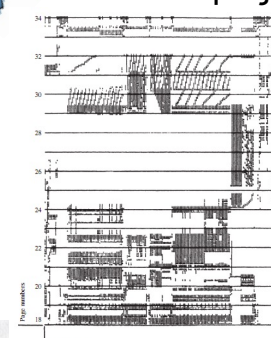
- Η αρχή της τοπικότητας των αναφορών (principle of locality) ορίζει ότι ανά πάσα χρονική στιγμή η ροή εκτέλεσης ενός προγράμματος εστιάζεται μόνο σε ένα μέρος του κώδικα και των δεδομένων του προγράμματος.
- Κατ' επέκταση, μόνο το μέρος της διεργασίας στο οποίο βρίσκονται η ομάδα των τρεχουσών εκτελούμενων εντολών και τα σχετιζόμενα με αυτές δεδομένα χρειάζεται να βρίσκεται στη μνήμη για το άμεσο μέλλον.
- Επιπλέον, είναι δυνατόν να κάνουμε ευφύεις προβλέψεις για τα μέρη μίας διεργασίας που θα χρειασθούν στο άμεσο μέλλον, ώστε να μην τα απομακρύνουμε από τη μνήμη και να αποφύγουμε το φαινόμενο του thrashing.
- Η τοπικότητα των αναφορών είναι άμεση θετική επίπτωση της χρήσης αρθρωτών και δομημένων τεχνικών προγραμματισμού (και για αυτόν το λόγο δεν γίνεται συχνή αναφορά σε συγκεκριμένη «κρυφή» εντολή στις γλώσσες προγραμματισμού — πιο;)




11



Εικονική αναπαράσταση της τοπικότητας των αναφορών



- Παρατηρήστε ότι κατά τη διάρκεια εκτέλεσης της διεργασίας, η αναφορά σε διευθύνσεις εστιάζεται σε υποσύνολο των σελίδων.



12

Υλοποίηση της ιδεατής μνήμης

- Για την αποτελεσματική υλοποίηση της τεχνικής της ιδεατής μνήμης, απαιτούνται δύο προϋποθέσεις:
 - Το υλικό του συστήματος πρέπει να υποστηρίζει σελιδοποίηση ή/και κατάτμηση.
 - Το Λ.Σ. πρέπει να υποστηρίζει μηχανισμούς μεταφοράς σελίδων ή/και τμημάτων μεταξύ κύριας και περιφερειακής μνήμης.

13

Σελιδοποίηση με χρήση ιδεατής μνήμης

- Ο μηχανισμός παραμένει λίγο πολύ ο ίδιος άλλα τώρα θα πρέπει να ληφθεί υπ' όψη ότι οι σελίδες εναλλάσσονται μεταξύ κύριας και περιφερειακής μνήμης.
- Έτσι κάθε εγγραφή στον πίνακα σελίδων έχει επιπλέον πληροφορίες ελέγχου:
 - Το bit παρουσίας/απουσίας δηλώνει αν η σελίδα βρίσκεται στην κύρια ή στην περιφερειακή μνήμη.
 - Το bit τροποποίησης (modified bit) δηλώνει αν η σελίδα έχει υποστεί αλλαγές από τότε που εισήλθε στην κύρια μνήμη.
 - Το bit αναφοράς (referenced bit) δηλώνει αν η σελίδα έχει χρησιμοποιηθεί από το πρόγραμμα για γράψιμο ή για διάβασμα.
 - Τέλος, υπάρχουν άλλα bits ελέγχου όπως bits προστασίας, κλπ.

14

Πίνακας σελίδων

Virtual Address

Page Number	Offset
-------------	--------

Page Table Entry

Other Control Bits	Frame Number
--------------------	--------------

(a) Paging only

15

Αποκωδικοποίηση διευθύνσεων

Figure 8.2 Address Translation in a Paging System

16

Υλοποίηση πίνακα σελίδων

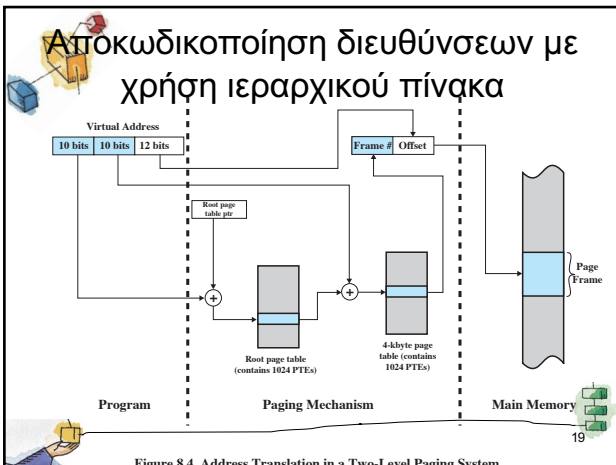
- Από τη στιγμή που ο αριθμός των σελίδων δεν περιορίζεται από το μέγεθος της κύριας μνήμης, ο πίνακας σελίδων τείνει να γίνεται μεγάλος.
 - Το να υπάρχουν πίνακες της τάξης του 1 εκατομμυρίου εγγραφών (δηλαδή 1 εκατομμύριο σελίδες) είναι σύνθηδες φαινόμενο. Σημειωτέον ότι κάθε διεργασία έχει το δικό της πίνακα!
- Αυτό δημιουργεί δύο προβλήματα:
 - Γρήγορη προσπέλαση στα περιεχόμενά του, μία πράξη η οποία είναι συνεχής (για αποκωδικοποίηση διευθύνσεων και εναλλαγή σελίδων).
 - Δέσμευση σημαντικού μέρους της μνήμης για την αποθήκευση των πινάκων σελίδων των διεργασιών που εκτελούνται.
- Έτσι ακόμα και ο πίνακας σελίδων πρέπει να χωρισθεί σε σελίδες και το μεγαλύτερο μέρος του να βρίσκεται στην περιφερειακή μνήμη.
- Αυτό οδηγεί στην οργάνωση του πίνακα σε πολλαπλά (περισσότερα από ενός) επίπεδα και την αποθήκευση μερικών από αυτών στο υλικό.

17

Ιεραρχική, δύο επιπέδων, υλοποίηση του πίνακα σελίδων

Figure 8.3 A Two-Level Hierarchical Page Table

18



19

Μέγεθος του ιεραρχικού πίνακα σελίδων

- Ένα μειονέκτημα του συγκεκριμένου είδους πίνακα σελίδων είναι ότι το μέγεθός του αυξάνεται αναλογικά με το μέγεθος της ιδεατής μνήμης.
 - Επομένως, δυνητικά μπορεί να είναι πολύ μεγάλος.
- Μία εναλλακτική λύση είναι η χρήση ανεστραμμένων πινάκων σελίδων (inverted page tables).

20

Ανεστραμμένοι πίνακες σελίδων

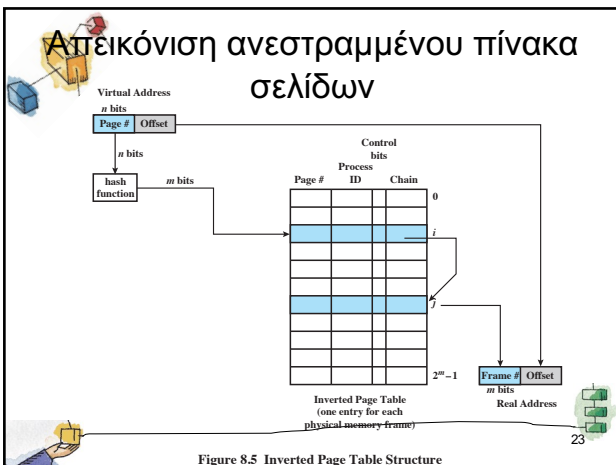
- Χρησιμοποιούνται στις αρχιτεκτονικές PowerPC, UltraSPARC, και IA-64.
- Οι αριθμοί των σελίδων που βρίσκονται στην ιδεατή μνήμη αντιστοιχίζονται σε ένα κωδικό τεμαχίου (hash value) μέσω μιας αντίστοιχης συνάρτησης (hash function).
- Ο κωδικός αυτός είναι ένας δείκτης στον ανεστραμμένο πίνακα που περιέχει τις πληροφορίες για τις σελίδες.
- Ο ανεστραμμένος πίνακας έχει μία εγγραφή για κάθε φυσικό πλαίσιο σελίδας του συστήματος (αντί για κάθε υπάρχουσα σελίδα), γι' αυτό και ονομάστηκε ανεστραμμένος.
 - Επομένως το μέγεθός του είναι σταθερό και ανάλογο με το μέγεθος της κύριας μνήμης.
- Επειδή περισσότερες από μία ιδεατές διευθύνσεις πιθανόν να αντιστοιχούν σε ένα κωδικό τεμαχίου, δημιουργούνται αλυσίδες οι οποίες όμως είναι συνήθως μικρού μεγέθους (1-2 εγγραφές).

21

Περιεχόμενα ενός ανεστραμμένου πίνακα σελίδων

- Αριθμός σελίδας.
 - Είναι το μέρος της ιδεατής διεύθυνσης μνήμης της σελίδας που αναφέρεται στον αριθμό της.
- Ταυτότητα της διεργασίας.
 - Η διεργασία στην οποία ανήκει η σελίδα.
 - Ο συνδυασμός αριθμού σελίδας και ταυτότητας διεργασίας αποτελεί μοναδική αναφορά μίας σελίδας μέσα στο ιδεατό χώρο διευθύνσεων μίας διεργασίας.
- Bits ελέγχου.
 - Παρόμοια με αυτά που χρησιμοποιούνται στον κλασικό πίνακα σελίδων.
- Δείκτης αλυσίδας.
 - Δείχνει στην επόμενη εγγραφή στην αλυσίδα.

22



23

Ενδιάμεση Μνήμη Μετάφρασης

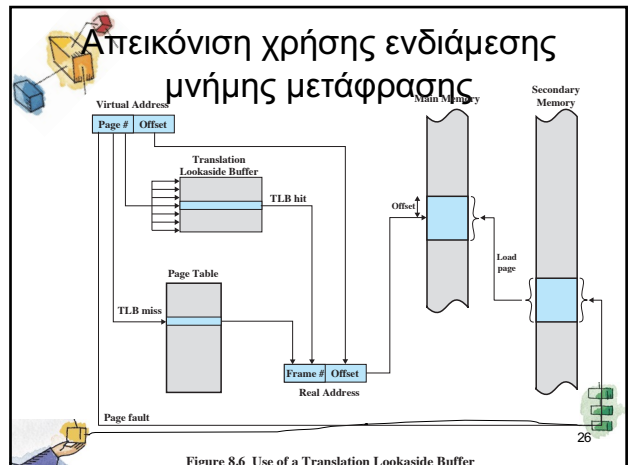
- Για την απεικόνιση μίας ιδεατής διεύθυνσης στην αντίστοιχη της φυσική διεύθυνση, απαιτούνται δύο προσβάσεις στη μνήμη:
 - Μία για το διάβασμα της αντίστοιχης εγγραφής στον πίνακα σελίδων.
 - Μία για την προσαγωγή των σχετικών δεδομένων.
- Για την επιτάχυνση της διαδικασίας αυτής χρησιμοποιείται ένα είδος πολύ γρήγορης μνήμης που λέγεται ενδιάμεση μνήμη μετάφρασης (translation lookaside buffer).
- Περιέχει τις εγγραφές του πίνακα σελίδων για τις σελίδες εκείνες που χρησιμοποιήθηκαν πιο πρόσφατα.

24

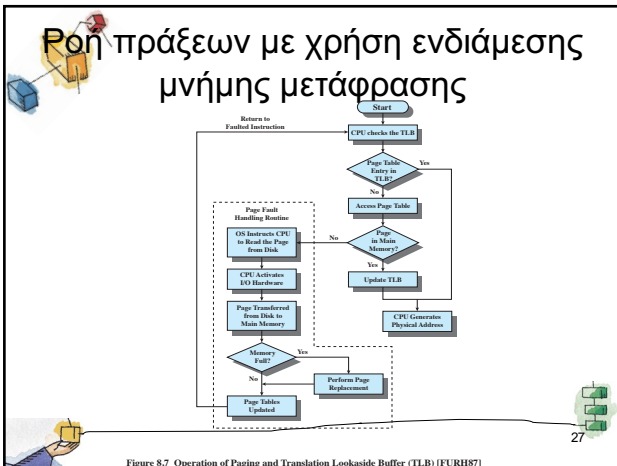
Χρήση ενδιάμεσης μνήμης μετάφρασης

- Για την αποκωδικοποίηση μιας ιδεατής διεύθυνσης μνήμης εξετάζονται πρώτα τα περιεχόμενα της ενδιάμεσης μνήμης μετάφρασης.
- Αν η σχετική εγγραφή του πίνακα σελίδων βρίσκεται εκεί, τότε ο αριθμός του αντίστοιχου πλαισίου σελίδων ανακτάται και χρησιμοποιείται για τον καθορισμό της αντίστοιχης φυσικής διεύθυνσης μνήμης.
- Διαφορετικά, ο αριθμός της σελίδας (από την ιδεατή διεύθυνση μνήμης) χρησιμοποιείται για να βρεθεί η σχετική εγγραφή στον πίνακα σελίδων.
- Αν η σελίδα δεν βρίσκεται στην κύρια μνήμη, τότε δημιουργείται σφάλμα σελίδας.
- Σε κάθε περίπτωση, τα περιεχόμενα της ενδιάμεσης μνήμης μετάφρασης ενημερώνονται με τις σχετικές πληροφορίες της νέας σελίδας.

25



26

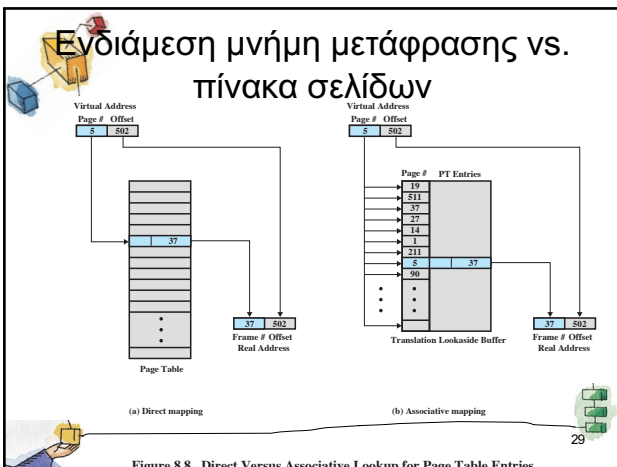


27

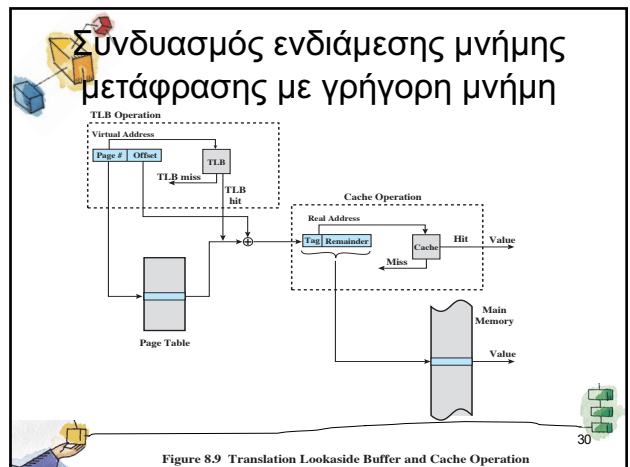
Συσχετιστική Απεικόνιση

- Επειδή η ενδιάμεση μνήμη μετάφρασης περιέχει μόνο μερικές από τις εγγραφές του πίνακα σελίδων, κάθε εγγραφή της περιέχει όχι μόνο τον αριθμό μίας σελίδας αλλά και όλη την εγγραφή για τη σελίδα αυτή.
- Ο επεξεργαστής έχει τη δυνατότητα να κάνει ταυτόχρονη προσπέλαση σε όλες τις εγγραφές της ενδιάμεσης μνήμης μετάφρασης για να βρει αν υπάρχει εκεί η σελίδα που ψάχνει.
- Αυτό λέγεται **συσχετιστική απεικόνιση** (associative mapping) και καθίσταται δυνατόν από την ίδια τη φύση των καταχωρητών που αποτελούν την ενδιάμεση μνήμη μετάφρασης.
- Το **ποσοστό επιτυχίας** (hit ratio) στο να βρεθούν οι ζητούμενες πληροφορίες στην ενδιάμεση μνήμη μετάφρασης παίζει πρωτεύοντα ρόλο στην ταχύτητα αποκωδικοποίησης των ιδεατών διευθύνσεων.

28



29



30

Μέγεθος σελίδας

- Αν είναι σχετικά μικρό ελαττώνει τον εσωτερικό κατακερματισμό της τελευταίας σελίδας ενός προγράμματος.
- Όμως αυξάνει το μέγεθος του πίνακα σελίδων (και κατ' επέκταση το χρόνο προσπέλασής του και το χώρο μνήμης που χρειάζεται για αποθήκευση).
- Αν είναι σχετικά μεγάλο αντιστρέφεται το πρόβλημα.
- Σημειωτέον ότι η περιφερειακή μνήμη σχεδιάζεται με τρόπο που να ευνοεί τη μεταφορά δεδομένων μεγάλου μεγέθους, επομένως είναι καλύτερη η χρήση μεγάλου μεγέθους σελίδας.
- Για να είναι όσο γίνεται πιο εύκολη η αποκωδικοποίηση μίας διεύθυνσης μνήμης το μέγεθος της σελίδας πρέπει να είναι δύναμη του 2 και συνήθως κυμαίνεται μεταξύ 1 K και 4 K (αλλά μπορεί να φτάσει και τα 256 MB).

31

Σχέση του μεγέθους σελίδας με τη συχνότητα σφαλμάτων σελίδας

- Αν το μέγεθος της σελίδας είναι μικρό, τότε υπό κανονικές συνθήκες ένας σχετικά μεγάλος αριθμός σελίδων για κάποια διεργασία θα βρίσκεται στη μνήμη.
- Μετά από κάποιο χρονικό διάστημα οι σελίδες αυτές θα έχουν διευθύνσεις που χρησιμοποιήθηκαν πρόσφατα και επομένως η συχνότητα δημιουργίας σφαλμάτων σελίδων θα είναι μικρή.
- Όσο αυξάνεται το μέγεθος της σελίδας, κάθε σελίδα θα περιέχει διευθύνσεις μακρύτερα από αυτές που χρησιμοποιήθηκαν πρόσφατα.
 - Κατ' επέκταση η αρχή της τοπικότητας των αναφορών δεν ισχύει και αυξάνει η συχνότητα δημιουργίας σφαλμάτων σελίδων.
- Κάποια στιγμή όμως η συχνότητα δημιουργίας σφαλμάτων σελίδων θα αρχίσει να μειώνεται, όταν το μέγεθος της σελίδας πλησιάζει το μέγεθος της διεργασίας (σημείο P στο επόμενο διάγραμμα) και θα γίνει 0 με ταύτιση των δύο μεγεθών.

32

Τυπική συμπεριφορά διεργασίας σε σχέση με τα σφάλματα σελίδας

Figure 8.10 Typical Paging Behavior of a Program

P = size of entire process
 W = working set size
 N = total number of pages in process

33

Παραδείγματα μεγεθών σελίδων

Table 8.3 Example Page Sizes

Computer	Page Size
Atlas	512 48-bit words
Honeywell-Multics	1024 36-bit word
IBM 370/XA and 370/ESA	4 Kbytes
VAX family	512 bytes
IBM AS/400	512 bytes
DEC Alpha	8 Kbytes
MIPS	4 Kbytes to 16 Mbytes
UltraSPARC	8 Kbytes to 4 Mbytes
Pentium	4 Kbytes or 4 Mbytes
IBM POWER	4 Kbytes
Itanium	4 Kbytes to 256 Mbytes

34

Κατάτμηση με χρήση ιδεατής μνήμης

- Επιτρέπει στον προγραμματιστή να βλέπει τη μνήμη ως ένα αριθμό από τμήματα, το καθένα σε διαφορετικό χώρο διεύθυνσεων.
- Τα τμήματα μπορούν να είναι διαφορετικού ή ακόμα και δυναμικού μεγέθους.
- Διευκολύνει τη διαχείριση μεγάλων δομών δεδομένων.
 - Ακόμα και αν δεν είναι γνωστό εξ' αρχής το μέγεθος μίας δομής δεδομένων, μπορεί να της δοθεί ένα τμήμα του οποίου το μέγεθος θα αυξομειώνεται δυναμικά ανάλογα με το μέγεθος της δομής.
 - Το τμήμα αυτό μπορεί να μετακινηθεί σε άλλο μέρος της μνήμης ή και να απομακρυνθεί από αυτήν, για να επανέλθει αργότερα.
- Επιτρέπει την επί μέρους τροποποίηση και μεταγλώττιση μέρους ενός προγράμματος.
- Υποστηρίζει διαμοίραση κώδικα και δεδομένων μεταξύ διεργασιών.
- Υποβοηθάει τους μηχανισμούς προστασίας διότι ένα τμήμα είναι λογικά αυτόνομο και μπορεί να συσχετισθεί με συγκεκριμένα δικαιώματα πρόσβασης στα περιεχόμενά του με χρήση bits προστασίας ή δακτυλίων προστασίας (protection rings).

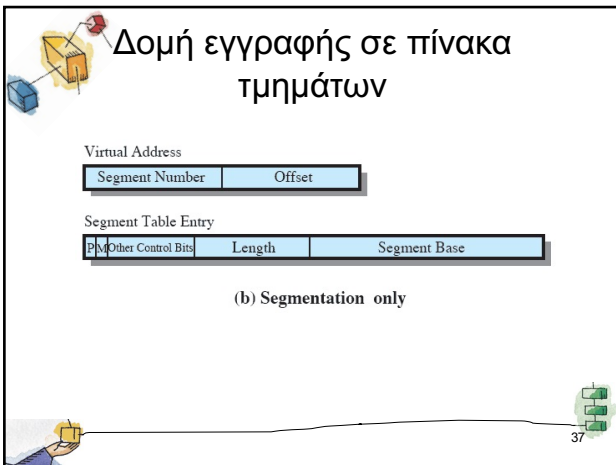
35

Οργάνωση τμημάτων

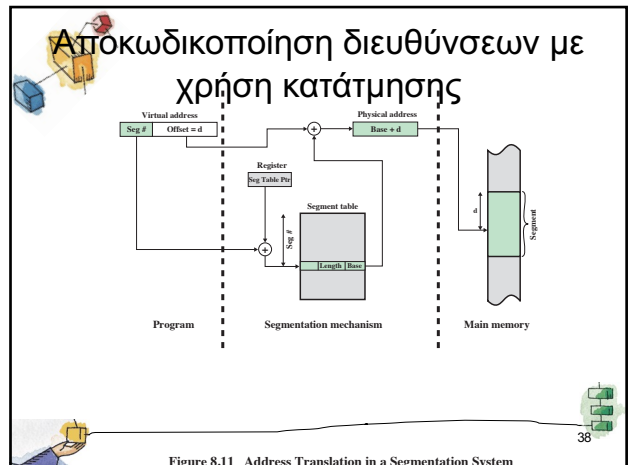
- Όπως και στην περίπτωση που δεν υποστηρίζεται ιδεατή μνήμη:
 - Για κάθε διεργασία υπάρχει ένας πίνακας τμημάτων για τη διεργασία αυτή.
 - Η αναφορά σε διεύθυνση μνήμης ενός τμήματος είναι της μορφής (αριθμός τμήματος, μετατόπιση).
 - Κάθε εγγραφή στον πίνακα τμημάτων περιέχει την πρώτη διεύθυνση του αντίστοιχου τμήματος στη μνήμη και το μέγεθος του τμήματος.
- Επιπλέον όμως υπάρχουν:
 - Ένα bit **προσασίας** το οποίο δηλώνει αν το τμήμα βρίσκεται ή όχι στη μνήμη.
 - Ένα bit **τροποποίησης** το οποίο δηλώνει αν τα περιεχόμενα του τμήματος έχουν υποστεί αλλαγές από την τελευταία φορά που φορτώθηκε το τμήμα από το δίσκο στη μνήμη. Αν δεν υπάρχει τροποποίηση, τότε δεν χρειάζεται να ενημερωθεί το αντίγραφο του τμήματος στο δίσκο, γλυτώνοντας έτσι μία πράξη E/E.

Αλλά bits ελέγχου.

36



37

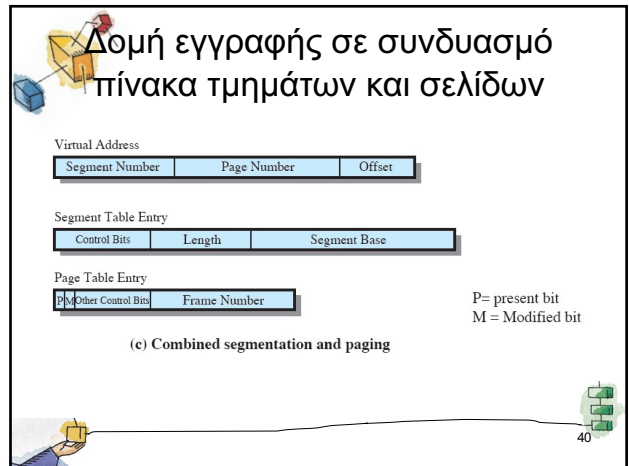


38

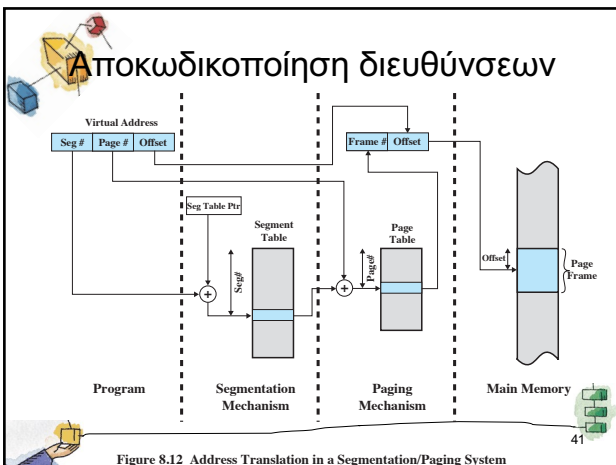
Συνδυασμός σελιδοποίησης και κατάτμησης

- Ο χώρος μνήμης του χρήστη χωρίζεται σε ένα αριθμό από τμήματα (με πρωτοβουλία του προγραμματιστή).
- Το κάθε τμήμα χωρίζεται σε ένα αριθμό από σελίδες, ίδιου μεγέθους με αυτό των πλαισίων σελίδας του συστήματος.
- Αν το τμήμα έχει μέγεθος μικρότερο από μία σελίδα, τότε κάνει χρήση μόνο μίας σελίδας.
- Από το σημείο αναφοράς του προγραμματιστή, μία λογική διεύθυνση έχει τη μορφή (αριθμός τμήματος, μετατόπιση).
- Από το σημείο αναφοράς του συστήματος όμως, μία λογική διεύθυνση έχει τη μορφή (αριθμός τμήματος, αριθμός σελίδας στο τμήμα, μετατόπιση).

39



40

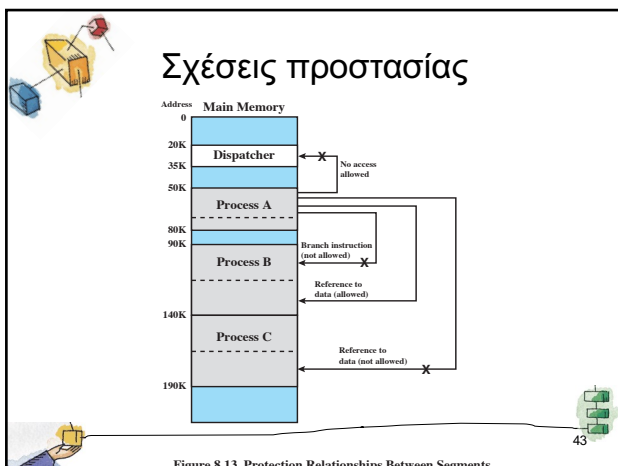


41

Προστασία και διαμοίραση

- Όπως ήδη έχει ειπωθεί, η κατάτμηση υποβοηθά τους μηχανισμούς προστασίας και διαμοίρασης.
- Κάθε εγγραφή στον πίνακα τμημάτων περιλαμβάνει το μέγεθος και την αρχική διεύθυνση του τμήματος.
 - Επομένως, ένα πρόγραμμα δεν μπορεί κατά λάθος να επιχειρήσει πρόσβαση σε μνήμη πέρα από τα όρια του τμήματος.
- Διαμοίραση επιτυγχάνεται μέσω της δυνατότητας να γίνεται αναφορά σε κάποιο τμήμα στον πίνακα τμημάτων διαφορετικών διεργασιών.

42



43

Περιεχόμενα

- Τεχνικές υλοποίησης ιδεατής μνήμης.
- - Στρατηγικές διαχείρισης ιδεατής μνήμης.
- Διαχείριση μνήμης στο UNIX και Solaris.
- Διαχείριση μνήμης στο Linux.
- Διαχείριση μνήμης στα Windows.
- Διαχείριση μνήμης στο Android.

44

Ερωτήματα σχετιζόμενα με τη διαχείριση μνήμης

- Χρήση ή όχι ιδεατής μνήμης.
- Χρήση σελιδοποίησης ή κατάτμησης ή συνδυασμού τους.
- Ποιος αλγόριθμος να χρησιμοποιηθεί για τις διάφορες πτυχές της διαχείρισης μνήμης.
- Οι απαντήσεις στα πρώτα δύο ερωτήματα συνήθως καθορίζονται από τις δυνατότητες της αρχιτεκτονικής του συστήματος.
 - Με ελάχιστες εξαιρέσεις πάντως (π.χ. το DOS) όλα τα Λ.Σ. υποστηρίζουν ιδεατή μνήμη.

45

Πτυχές της διαχείρισης ιδεατής μνήμης

- Πολιτικές προσκόμισης σελίδων στην κύρια μνήμη (fetch policies).
- Πολιτικές τοποθέτησης σελίδων στην κύρια μνήμη (placement policies).
- Πολιτικές αντικατάστασης σελίδων (replacement policies).
- Διαχείριση των σελίδων που βρίσκονται στην κύρια μνήμη (resident set management).
- Πολιτικές απομάκρυνσης σελίδων από την κύρια μνήμη (cleaning policies).
- Έλεγχος του αριθμού των διεργασιών που βρίσκονται στην κύρια μνήμη (load control). Έχει να κάνει περισσότερο με χρονοπρογραμματισμό και δεν θα ασχοληθούμε περισσότερο εδώ με αυτό.

46

Κοινός παρανομαστής των πτυχών αυτών


- Είναι η μεγιστοποίηση της απόδοσης του συστήματος που σε αυτή την περίπτωση σημαίνει την ελαχιστοποίηση του αριθμού των σφαλμάτων σελίδας που δημιουργούνται.
- Και αυτό γιατί η αντιμετώπιση ενός σφάλματος σελίδας είναι μία δαπανηρή σε χρόνο διαδικασία που αναγκάζει το Λ.Σ. μεταξύ άλλων να:
 - αποφασίσει ποια σελίδα πρέπει να απομακρυνθεί από την κύρια μνήμη,
 - εναλλάξει τη σελίδα αυτή με τη σελίδα που πρέπει να φορτωθεί στην κύρια μνήμη,
 - εναλλάξει την τρέχουσα διεργασία που εκτελείται με άλλη (process switching), κάτι που έχει επιπρόσθετο κόστος (η τελευταία αυτή εναλλαγή είναι αναγκαία γιατί η αντικατάσταση σελίδων στην κύρια μνήμη είναι ουσιαστικά πράξη E/E και παίρνει χρόνο να ολοκληρωθεί).
- Σημαντικό ρόλο στην απόδοση του συστήματος παίζουν επίσης παράγοντες όπως το μέγεθος της κύριας μνήμης, η ταχύτητα προσπέλασης στην κύρια και περιφερειακή μνήμη, ο αριθμός και μέγεθος των διεργασιών που εκτελούνται ταυτόχρονα καθώς επίσης και η συμπεριφορά των εκτελούμενων προγραμμάτων.

47

Πολιτικές προσκόμισης σελίδων στην κύρια μνήμη


- Καθορίζουν πότε μία σελίδα θα φορτωθεί στην κύρια μνήμη.
- Υπάρχουν δύο πολιτικές:
 - Απαίτηση σελιδοποίησης (demand paging).
 - Προσελιδοποίηση (prepaging).

48




Απαιτήση σελιδοποίησης

- Βασίζεται στη λογική ότι μόλις αρχίσει να εκτελείται μία καινούργια διεργασία δημιουργούνται στην αρχή πολλά σφάλματα σελίδας αλλά μόλις φορτωθούν οι σελίδες που χρειάζονται, αυτά μειώνονται σημαντικά.
- Επομένως, μετά το αρχικό φόρτωμα σελίδων, ο αριθμός των σφαλμάτων σελίδων θα είναι μικρός.




49




Προσελιδοποίηση

- Φορτώνεται μία ομάδα από σελίδες πριν ζητηθούν.
- Βασίζεται στο γεγονός ότι λόγω του τρόπου που λειτουργούν οι συσκευές αποθήκευσης, είναι πιο αποδοτικό να μεταφερθεί στην κύρια μνήμη μία ομάδα από σελίδες αποθηκευμένες συνεχόμενα, παρά μία μόνο σελίδα.
- Φυσικά, η τεχνική αυτή είναι ωφέλιμη μόνο αν η πλειοψηφία των σελίδων που φορτωθούν στη μνήμη αποδειχθεί τελικά ότι χρειάζονται.
- Σημειώτεον ότι η χρησιμότητα της προσελιδοποίησης δεν έχει αποδειχθεί στατιστικά.




50




Πολιτικές τοποθέτησης σελίδων στην κύρια μνήμη

- Σε συστήματα που χρησιμοποιούν σελιδοποίηση το πρόβλημα αυτό δεν υπάρχει γιατί μία σελίδα μπορεί να φορτωθεί σε οποιοδήποτε μέρος της κύριας μνήμης.
- Μόνο σε συστήματα που χρησιμοποιούν μόνο κατάτμηση πρέπει να χρησιμοποιηθεί κάποιος από τους αλγόριθμους τοποθέτησης.




51




Πολιτικές αντικατάστασης σελίδων

- Σκοπός τους είναι να επιλέξουν την «πιο κατάλληλη» σελίδα που πρέπει να απομακρυνθεί από την κύρια μνήμη για να δημιουργηθεί χώρος για να φορτωθεί κάποια άλλη σελίδα από την περιφερειακή μνήμη.
- Ο ορισμός της «πιο κατάλληλης» σελίδας δίνεται από την αρχή της βελτιστοποίησης (principle of optimality): η σελίδα που πρέπει να αντικατασταθεί είναι αυτή που δεν θα χρειασθεί στο μέλλον για το μεγαλύτερο χρονικό διάστημα.
- Φυσικά, μια και δεν μπορεί να προβλεφθεί το μέλλον, οι αλγόριθμοι που ακολουθούν προσπαθούν να εκτιμήσουν κατά προσέγγιση ποια είναι η καλύτερη σελίδα, λαμβάνοντας υπ' όψη τους το ιστορικό της συμπεριφοράς των σελίδων μίας διεργασίας.
- Σημειώτεον ότι όσο πιο πολύπλοκος είναι ένας αλγόριθμος τόσο περισσότερο κόστος έχει η υλοποίησή του.




52




Κλειδώμα πλαισίων σελίδων

- Ανεξάρτητα από το ποιος αλγόριθμος αντικατάστασης σελίδων εφαρμόζεται, ορισμένα πλαίσια σελίδων στη μνήμη μπορεί να είναι κλειδωμένα.
- Οι σελίδες που βρίσκονται στα κλειδωμένα πλαίσια δεν μπορούν να αντικατασταθούν.
- Τα κλειδωμένα πλαίσια έχουν πληροφορίες που τις χρειάζεται το Λ.Σ. για να λειτουργεί, όπως:
 - Κώδικας του πυρήνα του Λ.Σ.
 - Σημαντικές δομές ελέγχου.
 - Προσωρινή μνήμη.
- Το κλειδώμα επιτυγχάνεται με την ύπαρξη ενός bit κλειδώματος στον πίνακα τμημάτων ή/και σελίδων το οποίο εξετάζει ο αλγόριθμος αντικατάστασης πριν επιλέξει μία σελίδα για να την απομακρύνει από την κύρια μνήμη.




53



Βασικοί αλγόριθμοι αντικατάστασης σελίδων



- Τυχαία αντικατάσταση (random page replacement).
- Βέλτιστος (optimal).
- Μη πρόσφατα χρησιμοποιούμενη σελίδα (not recently used, NRU).
- Λιγότερο πρόσφατα χρησιμοποιούμενη σελίδα (least recently used, LRU).
- Πρώτη εισερχόμενη, πρώτη εξερχόμενη (first in first out, FIFO).
- Ρολόι (Clock).



54

Παράδειγμα



- Για την κατανόηση κάποιων από αυτούς τους αλγορίθμους, θα χρησιμοποιήσουμε ένα παράδειγμα.
- Μία διεργασία έχει 5 σελίδες (1-5) και τις έχουν δοθεί στη μνήμη 3 πλαίσια σελίδων.
- Επίσης, η πρόσβαση στα περιεχόμενα των σελίδων της έχει ως εξής:
 $- 2 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 2.$

55

Τυχαία επιλογή



- Απλά επιλέγει τυχαία μία σελίδα.
- Έχει μικρό κόστος εκτέλεσης αλλά μπορεί κάλλιστα να διαλέξει για αντικατάσταση μία σελίδα που αμέσως μετά θα χρησιμοποιηθεί ξανά.
- Έτσι εφαρμόζεται σπάνια αφού στατιστικά έχει τα χειρότερα αποτελέσματα.
- Η μεγαλύτερη χρησιμότητά του έγκειται στο να αποτελεί μέτρο σύγκρισης για τους υπόλοιπους αλγόριθμους.

56

Βέλτιστος

- Εξετάζει τις εντολές του εκτελούμενου προγράμματος και επιλέγει τη σελίδα εκείνη στην οποία θεωρεί ότι θα γίνει αναφορά όσο γίνεται πιο μακριά στο μέλλον.
- Έχει πάντα τα καλύτερα αποτελέσματα.
- Το πρόβλημα είναι ότι δεν μπορούμε να ξέρουμε τη μελλοντική συμπεριφορά του προγράμματος.
- Επομένως ο αλγόριθμος δεν μπορεί να υλοποιηθεί.
- Όμως, όπως και ο προηγούμενος, μπορεί να χρησιμεύσει σαν μέσο σύγκρισης για τους υπόλοιπους.

57



Παράδειγμα χρήσης του βέλτιστου αλγόριθμου

Page address stream	2	3	2	1	5	2	4	5	3	2	5	2
OPT	2	2	2	2	2	2	4	4	4	4	4	4
							F	F	F	F	F	F

F = page fault occurring after the frame allocation is initially filled

Figure 8.15 Behavior of Four Page Replacement Algorithms



- Με αυτόν τον αλγόριθμο έχουμε 3 σφάλματα, μετά το γέμισμα των πλαισίων.

58

Μη πρόσφατα χρησιμοποιούμενη σελίδα



- Επιλέγει τη σελίδα εκείνη στην οποία δεν έχει γίνει αναφορά για ένα συγκεκριμένο χρονικό διάστημα (ένα κύττο του ρολογιού, δηλαδή γύρω στα 20 msec).
- Ο αλγόριθμος αυτός είναι εύκολα κατανοητός, έχει καλές επιδόσεις και υλοποιείται εύκολα με απλή αναφορά στα bits εκείνα του πίνακα σελίδων που δείχνουν αν έχει γίνει αναφορά ή τροποποίηση σε μία σελίδα.

59

Λιγότερο πρόσφατα χρησιμοποιούμενη σελίδα

- Επιλέγει τη σελίδα εκείνη στην οποία δεν έχει γίνει αναφορά για το μεγαλύτερο χρονικό διάστημα.
- Με βάση το φαινόμενο της τοπικότητας των αναφορών, οι πιθανότητες η σελίδα αυτή να χρειασθεί στο εγγύς μέλλον είναι μικρές.
- Ο αλγόριθμος αυτός έχει καλές επιδόσεις οι οποίες πλησιάζουν αυτές του βέλτιστου.
- Όμως είναι πολυέξοδος στην υλοποίηση γιατί χρειάζεται να καταγράφεται ο χρόνος που έγινε αναφορά σε μία σελίδα κάθε φορά που χρησιμοποιείται για ανάγνωση ή/και τροποποίηση, κάτι που έχει μεγάλο κόστος.

60

Παράδειγμα χρήσης του αλγόριθμου LRU

Page address stream: 2 3 2 1 5 2 4 5 3 2 5 2

LRU:

2	2	2	2	2	2	2	2	3	3	3	3
	3	3	3	5	5	5	5	4	4	4	4
			1	1	1	1	1	2	2	2	2

F = page fault occurring after the frame allocation is initially filled

Figure 8.15 Behavior of Four Page Replacement Algorithms

- Ο LRU έχει σχεδόν τις ίδιες επιδόσεις με αυτές του βέλτιστου.
 - Στο παράδειγμά μας δημιουργεί 4 λάθη σελίδας.

61

Πρώτη εισερχόμενη, πρώτη εξερχόμενη

- Επιλέγει για αντικατάσταση τη σελίδα εκείνη που έχει μείνει για μεγαλύτερο χρονικό διάστημα στην κύρια μνήμη.
- Βασίζεται στη λογική ότι οι πιθανότερες η σελίδα αυτή να συνεχίσει να είναι χρήσιμη είναι μικρές.
- Είναι απλός στην υλοποίηση (χρειάζεται μόνο μία κυκλική ουρά) αλλά η λογική στην οποία βασίζεται δεν είναι συχνά σωστή και έτσι έχει κακή απόδοση.
- Ο λόγος που η λογική δεν είναι πάντα σωστή έγκειται στο γεγονός ότι σε ένα πρόγραμμα υπάρχουν συχνά περιοχές κώδικα ή/και δεδομένων που χρησιμοποιούνται κατ' εξακολούθηση.
 - Σε αυτές τις περιπτώσεις, οι σελίδες που αντιστοιχούν σε αυτές τις περιοχές θα μπαίνουν συνεχία στη μνήμη.

62

Παράδειγμα χρήσης του αλγόριθμου FIFO

Page address stream: 2 3 2 1 5 2 4 5 3 2 5 2

FIFO:

2	2	2	3	3	3	3	3	3	3	3	3
			5	5	5	5	5	5	5	5	5

F = page fault occurring after the frame allocation is initially filled

Figure 8.15 Behavior of Four Page Replacement Algorithms

- Ο αλγόριθμος FIFO δημιουργεί 6 σφάλματα σελίδας.
 - Παρατηρείστε ότι ενώ ο LRU αναγνωρίζει ότι στις σελίδες 2 και 5 γίνεται αναφορά πιο συχνά από ότι στις υπόλοιπες σελίδες, ο FIFO δεν καταφέρνει να κάνει το ίδιο.

63

Το παράδοξο του Belady

- Ο αλγόριθμος FIFO ορισμένες φορές δημιουργεί περισσότερα σφάλματα όταν αυξηθεί ο αριθμός των πλαισίων.
- Θεωρήστε το διπλανό παράδειγμα όπου σε μία διεργασία έχουν δοθεί αρχικά 3 πλαίσια και κατόπιν 4. Η διεργασία αυτή έχει 5 σελίδες, στις οποίες κάνει αναφορά με την εξής σειρά: 0→1→2→3→0→1→4→0→1→2→3→4.
- Με 3 πλαίσια ο αριθμός των σφαλμάτων είναι 9 ενώ με 4 είναι 10.
- Η παρατήρηση αυτή έγινε από τον [Belady](#) το 1969.

0	1	2	3	0	1	4	4	4	2	3	3
	0	1	2	3	0	1	1	1	4	2	2
		0	1	2	3	0	0	0	1	4	4

0	1	2	3	3	4	0	1	2	3	4	
	0	1	2	2	3	4	0	1	2	2	
		0	1	1	1	2	3	4	0	1	2
			0	0	0	1	2	3	4	0	1

64

Ρολόι

- Ο αλγόριθμος του ρολογιού χρησιμοποιεί σε κάθε πλαίσιο σελίδας ένα επιπρόσθετο bit χρήσης.
 - Όταν μία σελίδα φορτωθεί για πρώτη φορά σε κάποιο πλαίσιο, η τιμή του bit αυτού είναι 1.
 - Επίσης, κάθε φορά που γίνεται αναφορά στα περιεχόμενα της σελίδας, η τιμή του bit γίνεται 1.
- Ο αλγόριθμος θεωρεί ότι τα πλαίσια των οποίων οι σελίδες είναι υποψήφιος για αποχώρηση από τη μνήμη δημιουργούν ένα κύκλο με ένα δείκτη να δείχνει κυκλικά από το ένα πλαίσιο στο επόμενο.
 - Όταν μία σελίδα αντικαθιστάται, ο δείκτης μετατοπίζεται στο επόμενο πλαίσιο από αυτό του οποίου η σελίδα αντικαταστάθηκε.
 - Όταν πρέπει να αντικατασταθεί μία σελίδα, ο δείκτης σαρώνει κυκλικά τα πλαίσια για να βρει ένα πλαίσιο με τιμή 0 στο bit χρήσης και διαλέγει το πρώτο από αυτά για να αντικαταστήσει τη σελίδα του πλαισίου.
 - Κάθε φορά που βρίσκεται ένα πλαίσιο με τιμή 1 στο bit χρήσης το θέτει σε 0.
 - Αν όλα τα πλαίσια έχουν τιμή 1 στο bit χρήσης, τότε ο δείκτης θα κάνει έναν πλήρη κύκλο θέτοντας τις τιμές όλων των bits σε 0 και μετά θα διαλέξει το πρώτο από αυτά (το αρχικό πλαίσιο).

65

Παράδειγμα χρήσης του αλγόριθμου του ρολογιού

Page address stream: 2 3 2 1 5 2 4 5 3 2 5 2

CLOCK:

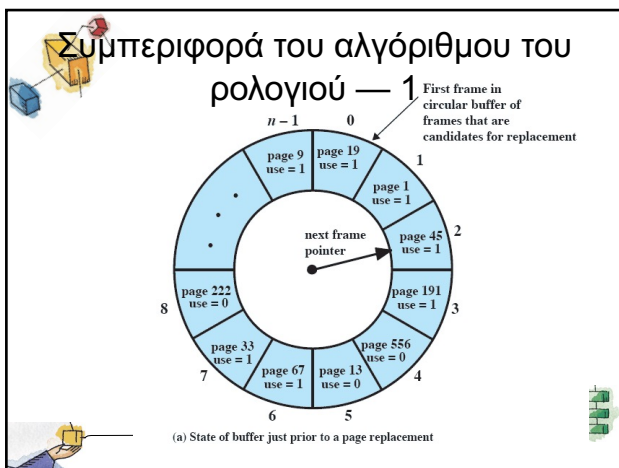
2	2	2	2	5	5	5	5	3	3	3	3
				1	1	1	1	2	2	2	2

F = page fault occurring after the frame allocation is initially filled

Figure 8.15 Behavior of Four Page Replacement Algorithms

- Ο αλγόριθμος είναι παρόμοιος με τον FIFO με εξαίρεση τη χρήση του bit χρήσης.
- Η παρουσία ενός αστερίσκου δηλώνει ότι το bit χρήσης είναι 1 και το βέλος αναπαριστά το δείκτη.
- Σημειώστε ότι ο αλγόριθμος προστατεύει από αντικατάσταση τη σελίδα 2.
- Λειτουργεί και αλγόριθμος της δεύτερης ευκαιρίας (second chance).

66



67

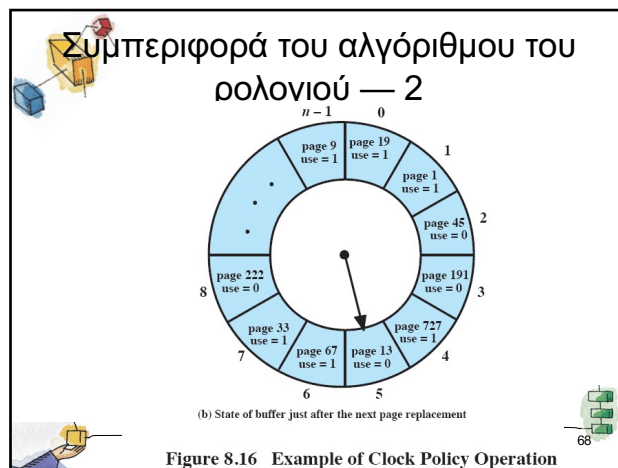
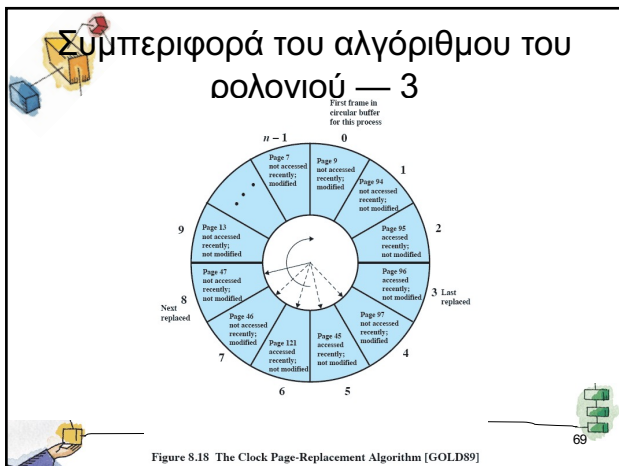


Figure 8.16 Example of Clock Policy Operation

68



69

Σύγκριση των αλγορίθμων αντικατάστασης σελίδων

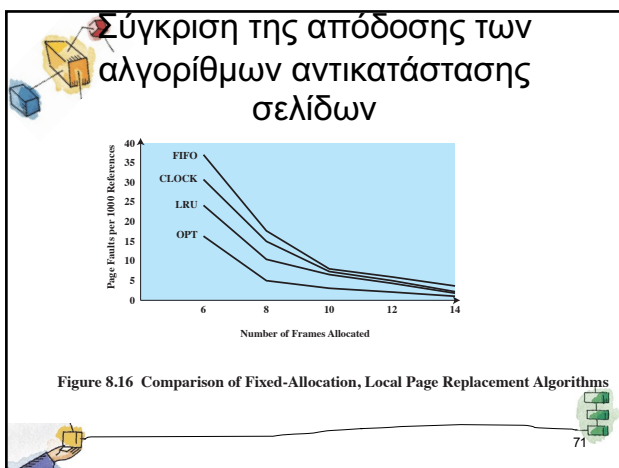
Page address stream: 2 3 2 1 5 2 4 5 3 2 5 2

OPT	2	2	2	2	2	2	4	4	4	2	2	3
	3	3	3	1	1	1	5	5	5	5	5	5
LRU	2	2	2	2	2	2	2	2	3	3	3	3
	3	3	3	1	1	1	4	4	4	4	2	2
FIFO	2	2	2	2	5	5	5	5	3	3	3	3
	3	3	3	1	1	1	4	4	4	4	2	2
CLOCK	2*	2*	2*	2*	5*	5*	5*	5*	3*	3*	3*	3*
	3	3	3	1	1	1	4	4	4	4	2	2

F = page fault occurring after the frame allocation is initially filled

Figure 8.14 Behavior of Four Page-Replacement Algorithms

70



71

Χρήση προσωρινής μνήμης για σελίδες

- Μία ενδιαφέρουσα παραλλαγή του αλγόριθμου FIFO με χρήση προσωρινής μνήμης εφαρμόστηκε στο Λ.Σ. VAX VMS.
- Όταν ο αλγόριθμος επιλέγει μία σελίδα για αντικατάσταση, αυτή δεν μεταφέρεται στο δίσκο αλλά τοποθετείται σε έναν από δύο προσωρινούς χώρους μνήμης (buffers), ανάλογα με το αν έχουν τροποποιηθεί τα περιεχόμενά της ή όχι (αυτό επιτυγχάνεται όχι με φυσική μετατόπιση των σελίδων στη μνήμη αλλά με αλλαγή των σχετικών δεικτών στους πίνακες σελίδων).
- Αν οι προσωρινοί χώροι είναι πλήρεις, τότε περιοδικά κάποιες από τις σελίδες αυτές (ξεκινώντας από αυτές που βρίσκονται στους χώρους αυτούς για το μεγαλύτερο χρονικό διάστημα) απομακρύνονται από αυτούς.
- Τα πλεονεκτήματα της μεθόδου είναι ότι για τις μη τροποποιημένες σελίδες αν χρειασθούν ξανά στο διάστημα που αυτές βρίσκονται στο προσωρινό χώρο δεν χρειάζεται να μεταφερθούν από το δίσκο και για τις τροποποιημένες σελίδες η μεταφορά τους στο δίσκο γίνεται σε ομάδες και όχι μία-μία, μειώνοντας έτσι τις πράξεις E/E.

72

Μέγεθος γρήγορης μνήμης και πολιτική αντικατάστασης σε αυτήν

- Με την αύξηση του μεγέθους της κύριας μνήμης, μειώνεται η τοπικότητα των αναφορών.
- Όμως αυξάνεται επίσης το μέγεθος της πολύ γρήγορης (κρυφής) μνήμης (cache).
- Η έξυπνη τοποθέτηση και αντικατάσταση σελίδων στη μνήμη cache μπορεί να έχει σημαντικό θετικό αντίκτυπο στην απόδοση του συστήματος (μείωση μέχρι και 20% της αποτυχίας εύρεσης μίας σελίδας στη μνήμη cache).

73

Διαχείριση των σελίδων που βρίσκονται στην κύρια μνήμη

- Το Λ.Σ. πρέπει να αποφασίσει πόσες σελίδες από κάθε διεργασία θα φορτώσει στην κύρια μνήμη.
 - Όσο μικρότερος είναι ο αριθμός τόσο περισσότερες διεργασίες θα μπορούν να βρίσκονται στη μνήμη.
 - Όμως, όσο μικρότερος είναι ο αριθμός τόσο μεγαλύτερη θα είναι και η συχνότητα σε σφάλματα σελίδων.
 - Επίσης, από ένα σημείο και μετά, παρέχοντας περισσότερες σελίδες σε μία διεργασία δεν θα επηρεάσει σημαντικά τη συχνότητα δημιουργίας λαθών σελίδας για αυτή τη διεργασία.

74

Προσεγγίσεις στον αριθμό των πλαισίων που κατανέμονται στις διεργασίες

- **Σταθερή κατανομή (fixed allocation).**
 - Διατηρεί σταθερό τον αριθμό των πλαισίων που χορηγούνται σε μία διεργασία.
 - Ο αριθμός αυτός υπολογίζεται από το Λ.Σ. όταν δημιουργείται η διεργασία και βασίζεται στις ιδιότητες της διεργασίας ή σε πληροφορίες από τον προγραμματιστή.
 - Όταν δημιουργείται σφάλμα σελίδας, αντικαθίσταται μία από τις σελίδες αυτών των πλαισίων.
- **Μεταβλητή κατανομή (variable allocation).**
 - Επιτρέπει δυναμικά την αύξηση του αριθμού των πλαισίων κάποιων διεργασιών, κυρίως αυτών που δημιουργούν πολλά σφάλματα σελίδας, αλλά και τη μείωση των αριθμών των πλαισίων για διεργασίες που έχουν περιορισμένη ανάγκη σε μνήμη.

75

Εύρος εφαρμογής των πολιτικών αντικατάστασης σελίδων

- Οι πολιτικές αντικατάστασης σελίδων μπορεί να εφαρμοσθούν σε:
 - **Τοπικό επίπεδο (local replacement policy)** όπου εξετάζονται μόνο οι σελίδες της διεργασίας που δημιούργησε το σφάλμα σελίδας, ή σε
 - **Ολικό επίπεδο (global replacement policy)** όπου εξετάζονται όλες τις σελίδες που βρίσκονται στην κύρια μνήμη, ανεξαρτήτως του αν ανήκουν ή όχι στη διεργασία που δημιούργησε το σφάλμα σελίδας.
- Ο συνδυασμός της σταθερής ή δυναμικής κατανομής και του τοπικού ή ολικού εύρους οδηγεί σε τρεις πιθανούς συνδυασμούς.

76

Σταθερή κατανομή με τοπική αντικατάσταση

- Αποφασίζεται από την αρχή πόσα πλαίσια θα χορηγηθούν σε κάθε διεργασία, με βάση τις ιδιότητες της διεργασίας και τις ανάγκες του προγράμματος.
- Αν συμβεί σφάλμα σελίδας, η αντικατάσταση γίνεται μόνο ανάμεσα στις σελίδες της διεργασίας που βρίσκονται στη μνήμη.
- Αν η κατανομή δεν επαρκεί, τότε θα δημιουργηθούν πολλά σφάλματα σελίδων.
- Αν η κατανομή είναι υπέρ το δέον, τότε θα έχουμε μειωμένο αριθμό διεργασιών στη μνήμη και κατ' επέκταση είτε υποχρησιμοποίηση της ΚΜΕ ή επιπλέον κόστος σε εναλλαγή διεργασιών.

77

Μεταβλητή κατανομή με ολική αντικατάσταση

- Η πιο εύκολη στην υλοποίηση προσέγγιση (και γι' αυτό χρησιμοποιείται σε πολλά Λ.Σ.).
- Το Λ.Σ. διατηρεί μία λίστα από ελεύθερα πλαίσια.
- Όταν μία διεργασία προκαλέσει σφάλμα σελίδας, το Λ.Σ. της παραχωρεί ένα πλαίσιο από αυτήν τη λίστα και κατ' επέκταση επεκτείνεται το μέγεθος της διεργασίας στη μνήμη με επακόλουθο τη σταδιακή μείωση των σφαλμάτων σελίδας για αυτή τη διεργασία.
- Όταν δεν υπάρχουν ελεύθερα πλαίσια, το Λ.Σ. πρέπει να αφαιρέσει ένα από αυτά που διαθέτει κάποια διεργασία και να το παραχωρήσει σε αυτήν που δημιούργησε το σφάλμα σελίδας.
 - Εδώ βρίσκεται το μειονέκτημα της μεθόδου αυτής: να αποφασισθεί σε ποια διεργασία θα υπάρξει η μικρότερη αρνητική επίπτωση αν της αφαιρεθεί ένα πλαίσιο από αυτά που διαθέτει.

78

Μεταβλητή κατανομή με τοπική αντικατάσταση

- Όταν μία διεργασία φορτωθεί στη μνήμη, της παραχωρείται ένας αριθμός πλαισίων (με χρήση προσελιδοποίησης ή απαίτησης σελιδοποίησης) με βάση τις ιδιότητες και ανάγκες της διεργασίας.
- Όταν δημιουργηθεί σφάλμα σελίδας, επιλέγεται για αντικατάσταση μία από τις σελίδες της διεργασίας αυτής.
- Όμως, περιοδικά το Λ.Σ. επανεξετάζει τη συμπεριφορά της κάθε διεργασίας έτσι ώστε να εκτιμηθεί αν χρειάζεται να γίνει ανακατανομή στον αριθμό των πλαισίων που ανήκουν σε κάθε διεργασία.
- Η τεχνική αυτή είναι πιο πολύπλοκη από την προηγούμενη αλλά μπορεί να οδηγήσει σε καλύτερα αποτελέσματα.

79

Εκτίμηση ανακατανομής στον αριθμό των πλαισίων

- Εδώ γίνεται χρήση του λεγόμενου μοντέλου συνόλου εργασίας (working set model) που προσπαθεί να υπολογίσει το μέγεθος της μνήμης που πρέπει να έχει κάθε διεργασία για να εκτελεσθεί χωρίς να δημιουργήσει πολλά σφάλματα σελίδας.
- Συνδυάζεται η χρήση του με αυτή του αλγόριθμου συχνότητας λάθους σελίδας (page fault frequency) που προσπαθεί να εντοπίσει τις διεργασίες εκείνες που δημιουργούν ένα απαράδεκτα υψηλό αριθμό σφαλμάτων σελίδων και ή να τις απομακρύνει προσωρινά από την κύρια μνήμη ή να τις εφοδιάσει με περισσότερα πλαίσια σελίδας.

80

Ανακεφαλαίωση των τεχνικών διαχείρισης σελίδων στη μνήμη

Table 8.5 Resident Set Management

	Local Replacement	Global Replacement
Fixed Allocation	<ul style="list-style-type: none"> Number of frames allocated to process is fixed. Page to be replaced is chosen from among the frames allocated to that process. 	<ul style="list-style-type: none"> Not possible.
Variable Allocation	<ul style="list-style-type: none"> The number of frames allocated to a process may be changed from time to time, to maintain the working set of the process. Page to be replaced is chosen from among the frames allocated to that process. 	<ul style="list-style-type: none"> Page to be replaced is chosen from all available frames in main memory; this causes the size of the resident set of processes to vary.

81

Πολιτικές απομάκρυνσης σελίδων από την κύρια μνήμη

- Καθορίζουν πότε μία σελίδα θα απομακρυνθεί από την κύρια μνήμη.
- Υπάρχουν δύο πολιτικές:
 - Απομάκρυνση μετά από απαίτηση (demand cleaning policy).
 - Προαπομάκρυνση (precleaning policy).
- Οι πολιτικές αυτές μπορούν να συνδυασθούν και με τη χρήση μνήμης cache (διαφάνεια 73).

82

Απομάκρυνση μετά από απαίτηση

- Ενημερώνει το αντίγραφο της σελίδας στην περιφερειακή μνήμη μόνο όταν πρέπει να απομακρυνθεί η σελίδα αυτή από την κύρια μνήμη.

83

Προαπομάκρυνση

- Ενημερώνει το αντίγραφο μίας σελίδας στην περιφερειακή μνήμη πριν χρειασθεί κάποια διεργασία το αντίστοιχο πλαίσιο.
- Αυτό επιτρέπει τη μεταφορά σελίδων στην περιφερειακή μνήμη κατά ομάδες αλλά η ενημέρωση των αντίστοιχων αντιγράφων τους μπορεί να καταστεί άχρηστη αν εν τω μεταξύ οι σελίδες αυτές έχουν τροποποιηθεί.

84

Έλεγχος φόρτου

- Ο έλεγχος φόρτου (load control) καθορίζει το βέλτιστο αριθμό διεργασιών που πρέπει να βρίσκονται στη μνήμη, δηλαδή το βαθμό πολυπρογραμματισμού του συστήματος.
- Είναι σημαντική παράμετρος στην απόδοση του συστήματος γιατί:
 - Αν ο αριθμός των διεργασιών στη μνήμη είναι πολύ μικρός, τότε είτε η ΚΜΕ θα είναι ανενεργή για μεγάλα χρονικά διαστήματα είτε θα υπάρχει μεγάλο κόστος σε εναλλαγή διεργασιών.
 - Αν ο αριθμός των διεργασιών στη μνήμη είναι υπερβολικά μεγάλος, τότε κατά μέσο όρο οι περισσότερες διεργασίες δεν θα έχουν αρκετή μνήμη με αποτέλεσμα να δημιουργούνται πολύ συχνά σφάλματα σελίδας.

85

Πολυπρογραμματισμός

Figure 8.19 Multiprogramming Effects

86

Αναστολή εκτέλεσης διεργασιών

- Αν το Λ.Σ. αποφασίσει ότι πρέπει να μειωθεί ο βαθμός πολυπρογραμματισμού, τότε μία ή περισσότερες από τις διεργασίες που εκτελούνται πρέπει να τεθεί υπό αναστολή και να απομακρυνθεί από τη μνήμη.
- Τίθεται επομένως το ερώτημα με βάση ποια κριτήρια γίνεται η επιλογή των διεργασιών αυτών από το Λ.Σ.
- Εδώ υπάρχουν 6 επιλογές.

87

Πολιτικές αναστολής εκτέλεσης διεργασιών — 1

- Η διεργασία με τη χαμηλότερη προτεραιότητα.
 - Έχει να κάνει με πολιτικές χρονοδρομολόγησης και είναι άσχετη με θέματα απόδοσης.
- Η διεργασία που δημιούργησε σφάλμα σελίδας.
 - Υπάρχει πιθανότητα να μην έχει στη μνήμη τις σελίδες που χρειάζεται για να εκτελεστεί και θα τεθεί έτσι ή αλλιώς υπό αναστολή, επομένως κερδίζουμε σε κόστος Ε/Ε και εναλλαγής σελίδων.
- Η διεργασία που ενεργοποιήθηκε πιο πρόσφατα.
 - Έχει τη μεγαλύτερη πιθανότητα να μην έχει στη μνήμη τις σελίδες που χρειάζεται για να εκτελεστεί.

88

Πολιτικές αναστολής εκτέλεσης διεργασιών — 2

- Η διεργασία με τον μικρότερο αριθμό σελίδων στη μνήμη.
 - Έχει το μικρότερο κόστος στο μελλοντικό φόρτωμά της στη μνήμη αλλά τιμωρούνται οι διεργασίες με έντονη τοπικότητα των αναφορών.
- Η διεργασία με τον μεγαλύτερο αριθμό σελίδων στη μνήμη.
 - Απελευθερώνει πολλά πλαίσια σελίδας και μειώνει την ανάγκη για επιπρόσθετες αναστολές διεργασιών.
- Η διεργασία με τον μεγαλύτερο εναπομείναντα κβάντο εκτέλεσης.
 - Ακολουθεί τη φιλοσοφία του αλγόριθμου χρονοδρομολόγησης SPN.

89

Περιεχόμενα

- Τεχνικές υλοποίησης ιδεατής μνήμης.
- Στρατηγικές διαχείρισης ιδεατής μνήμης.
- Διαχείριση μνήμης στο UNIX και Solaris.
- Διαχείριση μνήμης στο Linux.
- Διαχείριση μνήμης στα Windows.
- Διαχείριση μνήμης στο Android.

90

Unix

- Επειδή το Unix είναι ανεξάρτητο πλατφόρμας, οι μηχανισμοί διαχείρισης μνήμης διαφέρουν κατά περίπτωση στις εκδόσεις του.
- Τα πρώτα Unix χρησιμοποιούσαν μεταβλητά τμήματα χωρίς σελιδοποίηση.
- Τα πιο σύγχρονα Unix (SVR4 και Solaris) χρησιμοποιούν σελιδοποίηση και ιδεατή μνήμη.
- Ο πυρήνας του Unix χρησιμοποιεί μνήμη χωρίς σελιδοποίηση.

91

Δομή σελίδων

Page frame number	Age	Copy on write	Modify	Reference	Valid	Protect
-------------------	-----	---------------	--------	-----------	-------	---------

(a) Page table entry

Swap device number	Device block number	Type of storage
--------------------	---------------------	-----------------

(b) Disk block descriptor

Page state	Reference count	Logical device	Block number	Pfdata pointer
------------	-----------------	----------------	--------------	----------------

(c) Page frame data table entry

Reference count	Page/storage unit number
-----------------	--------------------------

(d) Swap-use table entry

Figure 8.20 UNIX SVR4 Memory Management Formats

92

Στοιχεία του πίνακα σελίδων

Page Table Entry	
Page frame number	Refers to frame in real memory.
Age	Indicates how long the page has been in memory without being referenced. The length and contents of this field are processor dependent.
Copy on write	Set when more than one process shares a page. If one of the processes writes into the page, a separate copy of the page must first be made for all other processes that share the page. This feature allows the copy operation to be deferred until necessary and avoided in cases where it turns out not to be necessary.
Modify	Indicates page has been modified.
Reference	Indicates page has been referenced. This bit is set to zero when the page is first loaded and may be periodically reset by the page replacement algorithm.
Valid	Indicates page is in main memory.
Protect	Indicates whether write operation is allowed.

93

Πεδία περιγραφέα μπλοκ δίσκου

Disk Block Descriptor	
Swap device number	Logical device number of the secondary device that holds the corresponding page. This allows more than one device to be used for swapping.
Device block number	Block location of page on swap device.
Type of storage	Storage may be swap unit or executable file. In the latter case, there is an indication as to whether the virtual memory to be allocated should be cleared first.

94

Δομή πλαισίων σελίδων

Page Frame Data Table Entry	
Page State	Indicates whether this frame is available or has an associated page. In the latter case, the status of the page is specified: on swap device, in executable file, or DMA in progress.
Reference count	Number of processes that reference the page.
Logical device	Logical device that contains a copy of the page.
Block number	Block location of the page copy on the logical device.
Pfdata pointer	Pointer to other pfdata table entries on a list of free pages and on a hash queue of pages.

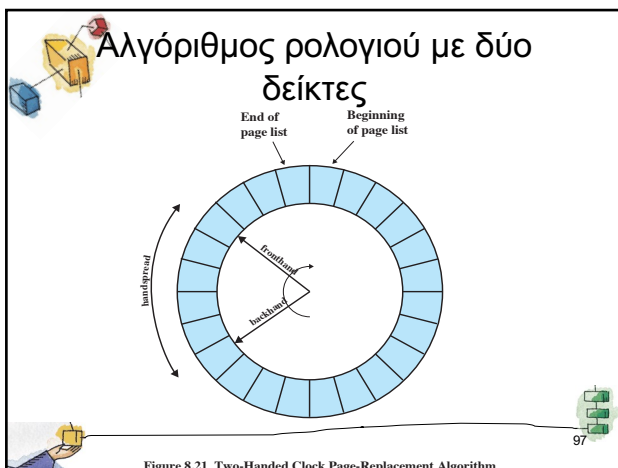
Swap-Use Table Entry	
Reference count	Number of page table entries that point to a page on the swap device.
Page/storage unit number	Page identifier on storage unit.

95

Αντικατάσταση σελίδων

- Γίνεται χρήση του πίνακα πλαισίων σελίδων.
- Γίνεται χρήση δεικτών για τη δημιουργία λίστας ελεύθερων πλαισίων.

96



97

Λειτουργία του αλγόριθμου του ρολογιού με δύο δείκτες

- Όταν μία σελίδα μεταφέρεται στη μνήμη, το bit αναφοράς είναι 0.
- Κάθε φορά που γίνεται αναφορά στη σελίδα για διάβασμα ή γράψιμο, το bit αυτό γίνεται 1.
- Ο πρώτος δείκτης σαρώνει τις σελίδες και θέτει το bit της κάθε σελίδας στη τιμή 0.
- Μετά από κάποιο χρονικό διάστημα, έρχεται η σειρά του δεύτερου δείκτη να σαρώσει τις σελίδες.
 - Αν το bit μίας σελίδας είναι 1, τότε η σελίδα αυτή έχει χρησιμοποιηθεί στο μεσοδιάστημα μεταξύ του σαρώματος των δύο δεικτών και επομένως αγνοείται.
 - Αν το bit μίας σελίδας είναι 0 τότε η σελίδα αυτή δεν έχει χρησιμοποιηθεί στο μεσοδιάστημα μεταξύ του σαρώματος των δύο δεικτών και επομένως γίνεται υποψήφια για απομάκρυνση από τη μνήμη.

98

Παράμετροι του αλγόριθμου του ρολογιού με δύο δείκτες

- Η συχνότητα με την οποία οι δείκτες σαρώνουν τις σελίδες (υπολογίζεται σε αριθμό σελίδων ανά δευτερόλεπτο).
- Η απόσταση μεταξύ των δεικτών (δηλαδή το χρονικό διάστημα που θα μεσολαβήσει μεταξύ του πρώτου και του δεύτερου σαρώματος).
- Οι τιμές των παραμέτρων αυτών καθορίζονται από το Λ.Σ. όταν ξεκινάει η λειτουργία του συστήματος αλλά μεταβάλλονται δυναμικά ανάλογα με το ποσοστό ελεύθερης μνήμης που υπάρχει ανά πάσα χρονική στιγμή στο σύστημα.

99

Διαχείριση μνήμης στον πυρήνα του Unix

- Ο πυρήνας συνεχώς δημιουργεί και καταστρέφει πίνακες και προσωρινούς χώρους μνήμης οι οποίοι είναι πολύ μικρότεροι σε μέγεθος από αυτό μίας τυπικής σελίδας.
- Επομένως η τεχνική της σελιδοποίησης δεν είναι κατάλληλη σε αυτή την περίπτωση.
- Ο πυρήνας για τη δυναμική δέσμευση και αποδέσμευση μνήμης κάνει χρήση μίας παραλλαγής του αλγόριθμου των φίλων (lazy buddy) που αποφεύγει να συγχωνεύσει μέρη μνήμης για όσο μεγαλύτερο χρονικό διάστημα μπορεί και μετά συγχωνεύει όσα περισσότερα μπορεί.
 - Βασίζεται στην παρατήρηση ότι το Unix συχνά χρειάζεται για σχετικά μεγάλο χρονικό διάστημα μπλοκ μνήμης συγκεκριμένου μεγέθους και δεν ωφείλει να συγχωνεύουμε πολύ συχνά τη μνήμη.

100

Παράμετροι του αλγόριθμου του Lazy Buddy

- N_i = τρέχων αριθμός μπλοκ μεγέθους 2^i .
- A_i = τρέχων αριθμός μπλοκ μεγέθους 2^i που είναι δεσμευμένα.
- G_i = τρέχων αριθμός μπλοκ μεγέθους 2^i που είναι ελεύθερα σε σχέση με όλη τη μνήμη.
- L_i = τρέχων αριθμός μπλοκ μεγέθους 2^i που είναι ελεύθερα σε σχέση με μέρος της μνήμης.
- $D_i = A_i - L_i = N_i - 2L_i - G_i$ είναι ο χρόνος που μεσολαβεί μεταξύ δύο συγχωνεύσεων.

101

Περιγραφή του Lazy Buddy

```

Initial value of  $D_i$  is 0
After an operation, the value of  $D_i$  is updated as follows

(I) if the next operation is a block allocate request:
    if there is any free block, select one to allocate
    if the selected block is locally free
        then  $D_i := D_i + 2$ 
        else  $D_i := D_i + 1$ 
    otherwise
        first get two blocks by splitting a larger one into two (recursive operation)
        allocate one and mark the other locally free
         $D_i$  remains unchanged (but  $D$  may change for other block sizes because of the recursive call)

(II) if the next operation is a block free request
    Case  $D_i \geq 2$ 
        mark it locally free and free it locally
         $D_i := D_i - 2$ 
    Case  $D_i = 1$ 
        mark it globally free and free it globally; coalesce if possible
         $D_i := 0$ 
    Case  $D_i = 0$ 
        mark it globally free and free it globally; coalesce if possible
        select one locally free block of size  $2^i$  and free it globally; coalesce if possible
         $D_i := 0$ 
    
```

Figure 8.22 Lazy Buddy System Algorithm

102

Περιεχόμενα

- Τεχνικές υλοποίησης ιδεατής μνήμης.
- Στρατηγικές διαχείρισης ιδεατής μνήμης.
- Διαχείριση μνήμης στο UNIX και Solaris.
- Διαχείριση μνήμης στο Linux.
- Διαχείριση μνήμης στα Windows.
- Διαχείριση μνήμης στο Android.

103

Διαχείριση μνήμης στο Linux

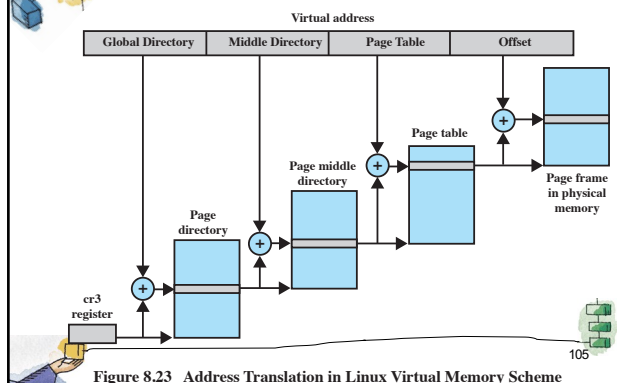
- Έχει πολλά κοινά με αυτήν του Unix αλλά είναι πιο πολύπλοκη.
- Ο πίνακας σελίδων είναι τριών επιπέδων και έχει μέγεθος μίας σελίδας.
- Κάθε διεργασία έχει ένα κατάλογο σελίδων (page directory) που έχει μέγεθος μίας σελίδας και πρέπει να βρίσκεται στη μνήμη.
- Κάθε στοιχείο του καταλόγου αυτού είναι ένας δείκτης σε μία σελίδα στον ενδιάμεσο κατάλογο σελίδων (middle page directory) του οποίου το μέγεθος μπορεί να είναι μεγαλύτερο από αυτό μίας σελίδας.
- Κάθε στοιχείο του ενδιάμεσου καταλόγου σελίδων δείχνει σε μία σελίδα του πίνακα σελίδων του οποίου το μέγεθος επίσης μπορεί να είναι μεγαλύτερο από αυτό μίας σελίδας.
- Κάθε στοιχείο του πίνακα σελίδων δείχνει σε μία ιδεατή σελίδα της διεργασίας.

104

103

104

Αποκωδικοποίηση διευθύνσεων



105

Αντικατάσταση σελίδων

- Μέχρι την έκδοση 2.6.28, χρησιμοποιείτο μία παραλλαγή του αλγόριθμου του ρολογιού.
- Το bit χρήσης μίας σελίδας αντικαθίσταται με μία μεταβλητή 8 bits η οποία αυξάνεται κατά μία μονάδα κάθε φορά που η σελίδα χρησιμοποιείται.
- Ταυτόχρονα, το Linux περιοδικά σαρώνει τις σελίδες και μειώνει την τιμή της μεταβλητής.
- Οι σελίδες εκείνες με τιμή 0 στη μεταβλητή τους αυτή είναι υποψήφιες για απομάκρυνση από τη μνήμη.
- Σε γενικές γραμμές, όσο μεγαλύτερη τιμή έχει αυτή η μεταβλητή τόσο περισσότερη χρήση της αντίστοιχης σελίδας γίνεται και επομένως τόσο λιγότερο υποψήφια είναι αυτή για απομάκρυνση.
- Επομένως, ο αλγόριθμος του Linux ήταν μία μορφή της πολιτικής της λιγότερο χρησιμοποιούμενης σελίδας.

106

105

106

Ο αλγόριθμος του διαιρεμένου LRU

- 1

- Το πρόβλημα με τον προηγούμενο αλγόριθμο είναι ότι η περιοδική σάρωση των σελίδων για τη μείωση της μεταβλητής καταναλώνει πολύ χρόνο χρήσης της ΚΜΕ για μεγάλες ποσότητες μνήμης.
- Από την έκδοση 2.6.28 το Linux χρησιμοποιεί ένα καινούργιο αλγόριθμο, του διαιρεμένου LRU (split LRU).
- Στον νέο αλγόριθμο, για κάθε σελίδα υπάρχουν δύο bits, ένα για ενεργοποίηση και ένα για αναφορά. Η κάθε σελίδα βρίσκεται είτε στην ομάδα των ενεργοποιημένων σελίδων είτε στην ομάδα των μη ενεργοποιημένων σελίδων.
- Η φυσική μνήμη χωρίζεται σε «ζώνες» με βάση τη διεύθυνσή τους.
- Σε κάθε ζώνη υπάρχει μία συνδεδεμένη λίστα με τις σελίδες που βρίσκονται σε κάθε μία από τις δύο προαναφερθείσες καταστάσεις.
- Μία διεργασία χαμηλού επιπέδου (δαίμων) σαρώνει περιοδικά την κάθε ζώνη και τροποποιεί τα δύο bits με βάση την ακόλουθη λογική

107

107

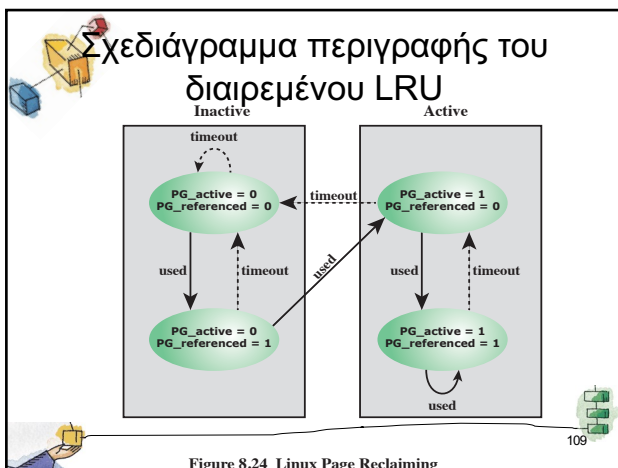
Ο αλγόριθμος του διαιρεμένου LRU

- 2

1. Την πρώτη φορά που μία σελίδα βρίσκεται σε μη ενεργοποιημένη κατάσταση αναφέρεται, το bit αναφοράς γίνεται 1.
2. Την επόμενη φορά που θα γίνει αναφορά σε αυτήν τη σελίδα, η σελίδα μεταφέρεται στη λίστα των ενεργοποιημένων σελίδων. Δηλαδή, χρειάζονται δύο συνεχόμενες αναφορές στη σελίδα για να θεωρηθεί ενεργοποιημένη.
3. Αν η δεύτερη αναφορά στη σελίδα δεν γίνει σε συγκεκριμένο χρονικό διάστημα, τότε το bit αναφοράς γίνεται 0.
4. Παρομοίως, για μία σελίδα που είναι ενεργοποιημένη, χρειάζονται δύο περίοδοι μη αναφοράς σε αυτήν (timeouts) για να μεταφερθεί η σελίδα στην ομάδα των μη ενεργοποιημένων σελίδων.
- Οι σελίδες που είναι στην ομάδα των μη ενεργοποιημένων σελίδων είναι υποψήφιες για αντικατάσταση με βάση τον αλγόριθμο LRU.

108

108



109

Κατανομή του χώρου του πυρήνα

- Στο επίπεδο του πυρήνα, το Linux κατανέμει τα φυσικά πλαίσια σελίδων, ανάλογα με τις ανάγκες που παρουσιάζονται:
 - Σε διεργασίες των χρηστών.
 - Για δεδομένα και κώδικα του πυρήνα.
 - Για τις ανάγκες κρυφής μνήμης (page cache).
- Γίνεται χρήση του αλγόριθμου των φίλων για την κατανομή της μνήμης στο μέγεθος μίας σελίδας.
- Όμως οι ανάγκες του πυρήνα σε μνήμη είναι συνήθως μικρότερη από το μέγεθος μίας σελίδας.
- Γι' αυτό χρησιμοποιείται η τεχνική της «κατανομής πλάκας» (slab allocation), που κατανέμει μέρος μίας σελίδας.

110

Περιεχόμενα

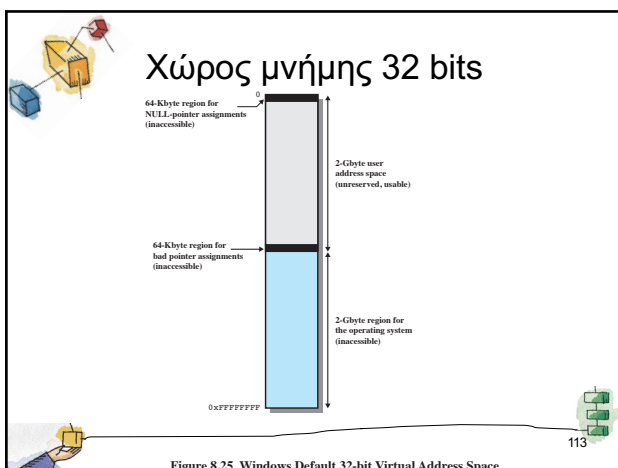
- Τεχνικές υλοποίησης ιδεατής μνήμης.
- Στρατηγικές διαχείρισης ιδεατής μνήμης.
- Διαχείριση μνήμης στο UNIX και Solaris.
- Διαχείριση μνήμης στο Linux.
- ➔ Διαχείριση μνήμης στα Windows.
- Διαχείριση μνήμης στο Android.

111

Διαχείριση μνήμης στα Windows

- Ο διαχειριστής μνήμης έχει σχεδιασθεί να λειτουργεί σε διαφορετικές πλατφόρμες και το μέγεθος της σελίδας κυμαίνεται μεταξύ 4K-64K (π.χ. στις πλατφόρμες Intel και AMD64 το μέγεθος είναι 4K και στην Intel Itanium είναι 8K).
- Σε πλατφόρμες των 32 bits, κάθε διεργασία έχει χώρο μνήμης των 32 bits, δηλαδή 4 GB.
- Ο μισός χώρος χρησιμοποιείται από τα ίδια τα Windows, αφήνοντας 2 GB για τη διεργασία.
- Υπάρχει πάντως η πρόβλεψη για αύξηση του χώρου μνήμης μίας διεργασίας στα 3 GB, αν υπάρχει ανάγκη.

112




113

Διαχείριση σελίδων



- Μόλις δημιουργηθεί, μία διεργασία μπορεί να χρησιμοποιήσει όλο το χώρο της των 2 GB.
- Ο χώρος αυτός είναι χωρισμένος σε σταθερά τμήματα των 64 KB, τα οποία με τη σειρά τους είναι χωρισμένα σε σελίδες σταθερού μεγέθους.
- Το κάθε τμήμα μπορεί να είναι σε μία από 3 καταστάσεις:
 - Διαθέσιμο, αν οι διευθύνσεις μνήμης του δεν χρησιμοποιούνται από τη διεργασία.
 - Κρατημένο, αν ο διαχειριστής μνήμης το χρησιμοποιεί για κάποιο σκοπό (π.χ. για αποθήκευση μίας ουράς).
 - Δεσμευμένο, αν οι διευθύνσεις του χρησιμοποιούνται από την ίδια τη διεργασία (οι σελίδες αυτού του τμήματος μπορεί να είναι στη μνήμη ή στο δίσκο).

114




Διαχείριση χώρου σελίδων

- Τα Windows χρησιμοποιούν την πολιτική της «μεταβλητής κατανομής, τοπικής ανακατάστασης».
- Μόλις ενεργοποιηθεί, μία διεργασία εφοδιάζεται με σχετικές δομές δεδομένων για να διαχειρισθεί τις σελίδες της, καθώς αυτές φορτώνονται στη μνήμη.
- Ο χώρος μνήμης που έχει δοθεί σε κάθε διεργασία αναπροσαρμόζεται περιοδικά με βάση πόση ελεύθερη μνήμη υπάρχει στο σύστημα και ανάλογα με την περίπτωση ο αριθμός πλαισίων που έχουν δοθεί σε κάθε διεργασία αυξάνεται ή μειώνεται (στη δεύτερη περίπτωση αφαιρούνται οι λιγότερο χρησιμοποιούμενες σελίδες της διεργασίας).



115

115




Περιεχόμενα

- Τεχνικές υλοποίησης ιδεατής μνήμης.
- Στρατηγικές διαχείρισης ιδεατής μνήμης.
- Διαχείριση μνήμης στο UNIX και Solaris.
- Διαχείριση μνήμης στο Linux.
- Διαχείριση μνήμης στα Windows.
- Διαχείριση μνήμης στο Android.

116



116



Διαχείριση χώρου μνήμης

- Το Android επεκτείνει το μοντέλο διαχείρισης μνήμης του πυρήνα του Linux με ένα αριθμό από επιπλέον χαρακτηριστικά, όπως τα ακόλουθα:
- ASHMem.
 - Παρέχει ανώνυμη διαμοιραζόμενη μνήμη, όπου η μνήμη παρουσιάζεται ως περιγραφέας αρχείου ο οποίος μπορεί να δοθεί από μία διεργασία σε άλλη για να διαμοιραστούν τη μνήμη.
- Pmem.
 - Παρέχει ιδεατή μνήμη σε φυσικά συνεχόμενο χώρο και είναι χρήσιμο για υλικό που δεν υποστηρίζει τεχνικές ιδεατής μνήμης.
- Low Memory Killer.


Ενημερώνει διεργασίες ότι πρέπει να απελευθερώσουν μνήμη και αν δεν συμμορφωθούν τερματίζει την εκτέλεσή τους.

117

117



ΕΠΛ222: Λειτουργικά Συστήματα
(μετάφραση στα ελληνικά των διαφανειών του βιβλίου Operating Systems: Internals and Design Principles, 9E, William Stallings)



Τέλος Ενότητας 8

Οι διαφάνειες αυτές έχουν συμπληρωματικό και επεξηγηματικό χαρακτήρα και σε καμία περίπτωση δεν υποκαθιστούν το βιβλίο

Γιώργος Α. Παπαδόπουλος
Τμήμα Πληροφορικής
Πανεπιστήμιο Κύπρου

Operating Systems
Internals and Design Principles
9th Edition
© 2013

118

118