



Λειτουργικά Συστήματα

Ενότητα 8: Διαχείριση Μνήμης. Φυσικές και
Λογικές Διευθύνσεις

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών
<http://arch.ict.e.uowm.gr/mdasyg>

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Σκοπός Ενότητας

- Η σημασία της MMU.
- Η κατανόηση της αντιστοίχισης ιδεατών και πραγματικών διευθύνσεων.



Η μνήμη RAM ποτέ δεν είναι αρκετή

- Η κύρια μνήμη είναι, μετά από το χρόνο χρήσης της CPU, ο **δεύτερος πιο σημαντικός πόρος** σε ένα υπολογιστικό σύστημα.
- Ακόμη και με σχετικά μεγάλο μέγεθος η ποσότητα της διαθέσιμης κύριας μνήμης, **συχνά δεν είναι ικανοποιητική**.
- Η λήψη πληροφοριών από τον σκληρό δίσκο αντί της κύριας μνήμης καθυστερεί υπέρμετρα το σύστημα.
 - 60ns χρόνος προσπέλασης της κύριας μνήμης.
 - 10ms (=10,000,000ns) μέσος χρόνος προσπέλασης των σκληρών δίσκων.
- Πολλές διεργασίες πρέπει να **συνυπάρχουν** στη μνήμη.

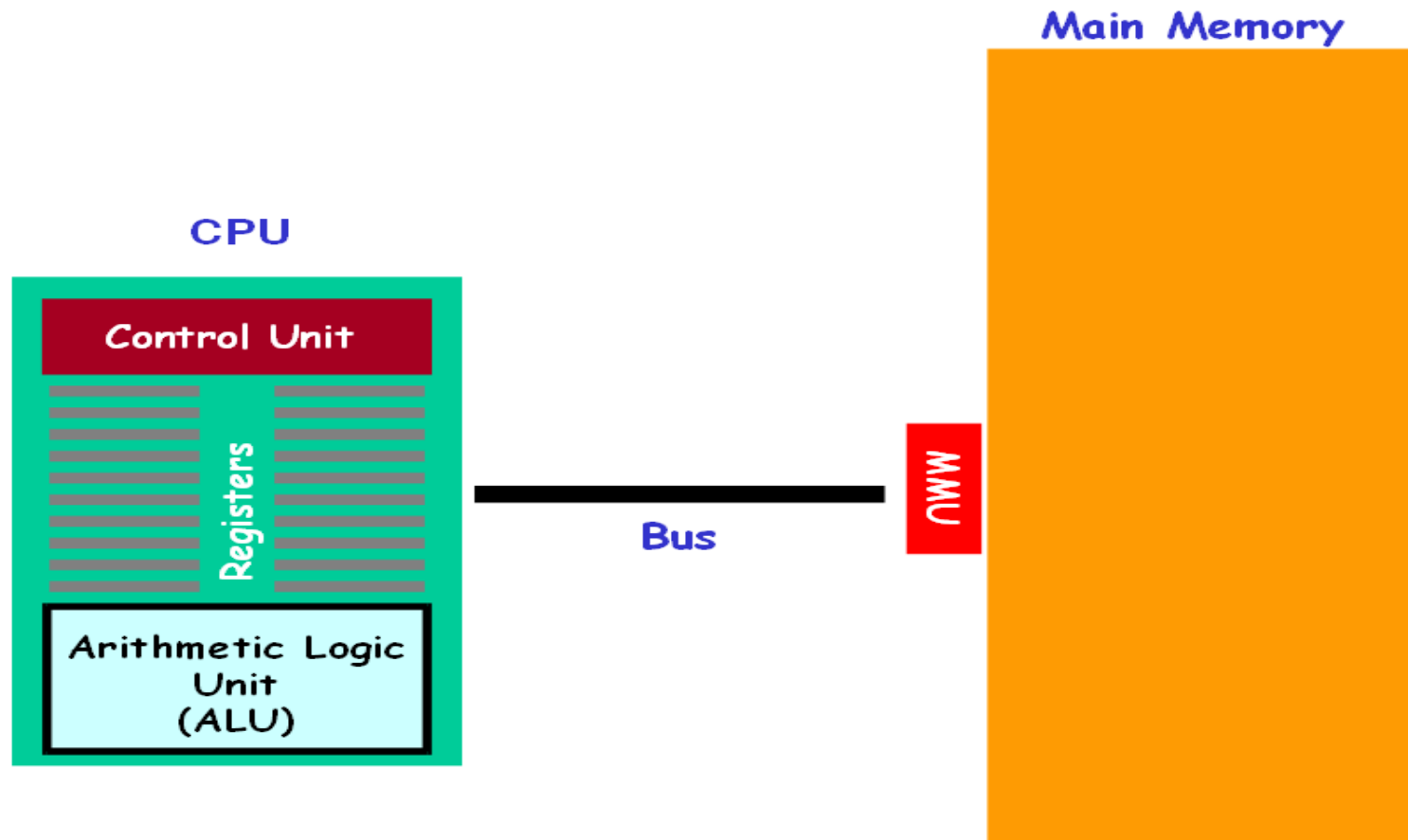


Διαχείριση Μνήμης: Συνεργασία hardware & Operating System

- Η διαχείριση μνήμης επιτυγχάνεται μέσω μιας πολύπλοκης σχέσης μεταξύ του υλικού μέρους του επεξεργαστή και του λογισμικού του Λ.Σ.
- Οι βασικές τεχνικές διαχείρισης μνήμης ανταγωνίζονται για τη δέσμευση περιορισμένου χώρου στην κύρια μνήμη.
- Η λύση της μεγαλύτερης κύριας μνήμης είναι συνήθως απαγορευτικά δαπανηρή.
- Η δεύτερη λύση είναι η δημιουργία της **ψευδαίσθησης ότι υπάρχει περισσότερη μνήμη** από όση είναι εγκατεστημένη και αποτελεί τη βασική ιδέα της ιδεατής μνήμης (virtual memory).



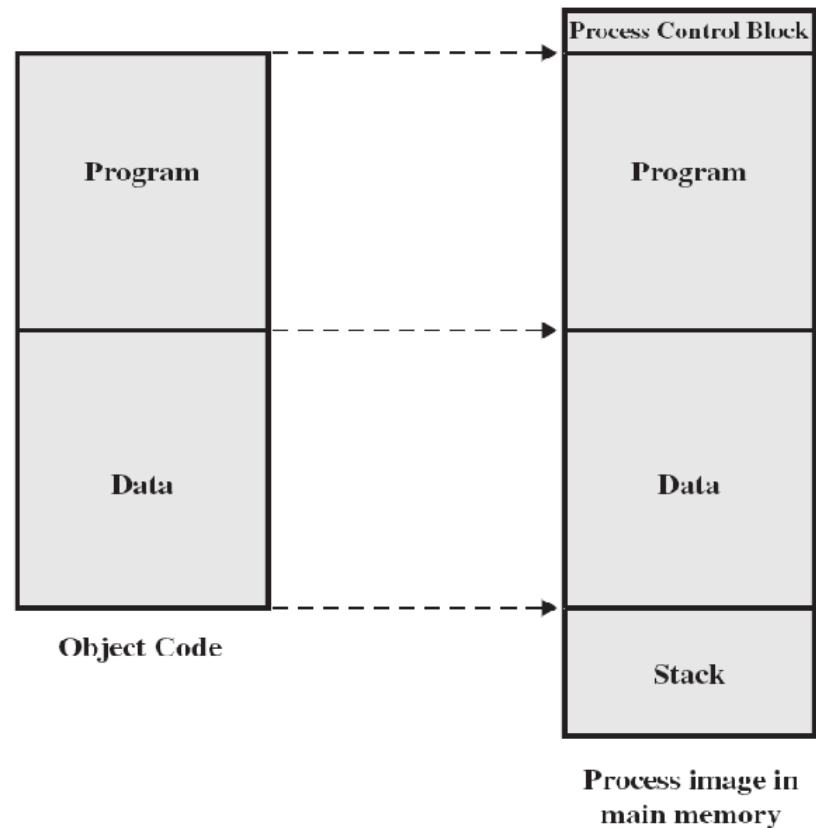
Μνήμη και CPU



Σύνδεση και Φόρτωση

Πρόγραμμα στο δίσκο

Το πρώτο βήμα κατά τη δημιουργία μιας ενεργής διεργασίας είναι η φόρτωση ενός προγράμματος στην κύρια μνήμη και η δημιουργία της εικόνας διεργασίας.



Η εφαρμογή αποτελείται από ενότητες (1/2)

- Η εφαρμογή αποτελείται από έναν αριθμό μεταφρασμένων και μεταγλωττισμένων ενοτήτων (**modules**) με τη μορφή κώδικα αντικειμένου (object code).
- Αυτές οι ενότητες συνδέονται για την επίλυση οποιασδήποτε αναφοράς μεταξύ τους, ενώ συγχρόνως πραγματοποιούνται αναφορές και σε ρουτίνες βιβλιοθήκης.
- Οι ρουτίνες βιβλιοθήκης μπορούν να ενσωματωθούν στο πρόγραμμα ή να αναφέρονται ως διαμοιραζόμενος κώδικας που πρέπει να παρέχεται από το Λ.Σ. κατά τη στιγμή της εκτέλεσης.

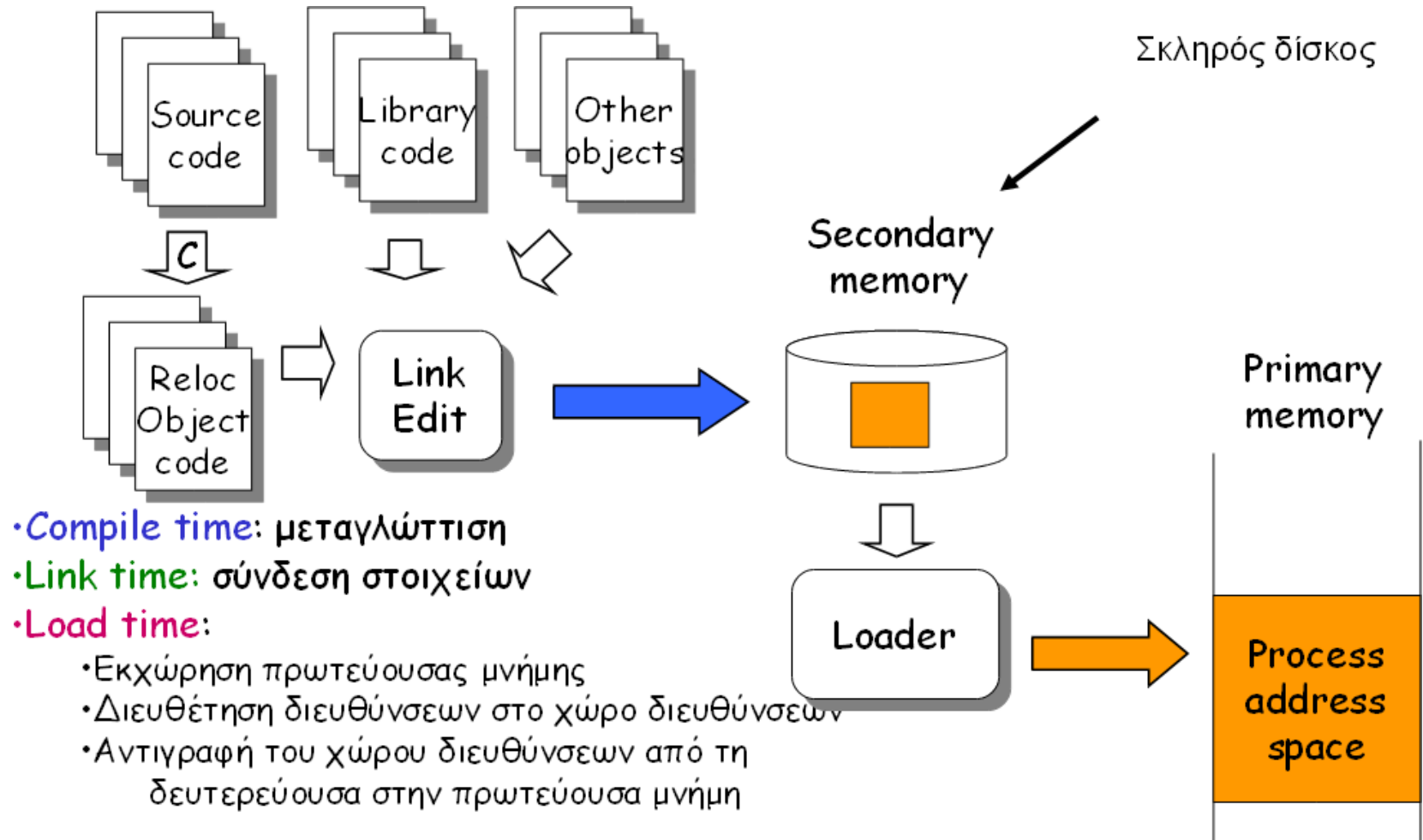


Η εφαρμογή αποτελείται από ενότητες (2/2)

- Η μονάδα σύνδεσης (**linker**) παίρνει ως είσοδο μια συλλογή από modules αντικειμένων και παράγει ένα module φόρτωσης που αποτελείται από ένα ολοκληρωμένο σύνολο από προγράμματα και δεδομένα που θα περάσουν στον φορτωτή. Σε κάθε module αντικειμένου μπορούν να υπάρχουν αναφορές διεύθυνσης σε θέσεις άλλων modules.
- Η μονάδα φόρτωσης τοποθετεί το module που πρόκειται να φορτωθεί στην κύρια μνήμη.



Δόμηση του χώρου διευθύνσεων



Δυο είναι οι τρόποι φόρτωσης

- **Απόλυτη φόρτωση.**

- Το υπό φόρτωση module φορτώνεται πάντοτε στην ίδια θέση στην κύρια μνήμη. Οι αναφορές διεύθυνσης θα πρέπει να είναι απόλυτες διευθύνσεις της κύριας μνήμης.

- **Φόρτωση με μετατόπιση (relocation).**

- Το υπό φόρτωση module μπορεί να τοποθετηθεί οπουδήποτε στην κύρια μνήμη. Ο interpreter ή ο compiler παράγουν διευθύνσεις σχετικές με ένα γνωστό σημείο που είναι συνήθως η αρχή του προγράμματος.



Στρατηγικές σύνδεσης

- **Στατική σύνδεση (static linking):**
 - Τα object files και οι βιβλιοθήκες συνδυάζονται σε ένα μοναδικό εκτελέσιμο αρχείο κατά τη διάρκεια της σύνδεσης.
- **Δυναμική σύνδεση (dynamic linking):**
 - Τα object files παραμένουν αποσυνδεδεμένα.
 - Ο κώδικας αντιστοιχείται και συνδέεται κατά τον χρόνο εκτέλεσης.
 - Παράδειγμα αποτελούν οι βιβλιοθήκες συστήματος (system libraries).



Στατική Σύνδεση & Δυναμική Σύνδεση (παράδειγμα Linux)

```
$ldd /rescue/ls
ldd: /rescue/ls: not a dynamic executable
```

```
$ldd /bin/ls
/bin/ls:
  libutil.so.5 => /lib/libutil.so.5 (0x2807f000)
  libncurses.so.6 => /lib/libncurses.so.6 (0x2808b000)
  libc.so.6 => /lib/libc.so.6 (0x280cc000)
```

\$ Το /bin/ls έχει δυναμική σύνδεση
ενώ το /rescue/ls έχει στατική
σύνδεση

```
$ls -la /bin/ls /rescue/ls
-r-xr-xr-x  1 root  wheel   23564 May 16  2008 /bin/ls
-r-xr-xr-x 131 root  wheel 3370176 May 16  2008 /rescue/ls
$
```



Η στατική σύνδεση απαιτεί πολύ περισσότερο χώρο, αλλά δε χρειάζεται εξωτερικά αρχεία.



Έξοδος «lsdf» (=list open files) (παράδειγμα Linux)

```
COMMAND  PID    USER  FD  TYPE  DEVICE  SIZE/OFF  NODE  NAME
init     1      root  cwd  VDIR  0,69    1024      2     /
init     1      root  rtd  VDIR  0,69    1024      2     /
init     1      root  txt  VREG  0,69    517240    1051  /sbin/init
devd     480    root  cwd  VDIR  0,69    1024      2     /
devd     480    root  rtd  VDIR  0,69    1024      2     /
devd     480    root  txt  VREG  0,69    282900    1110  /sbin/devd
devd     480    root  0u   VCHR  0,9     0t0      9     /dev/null
devd     480    root  1u   VCHR  0,9     0t0      9     /dev/null
devd     480    root  2u   VCHR  0,9     0t0      9     /dev/null
devd     480    root  3r   VCHR  0,4     0t4658   4     /dev/devctl
devd     480    root  4u   unix  0xc3167000 0t0      /var/run/devd.pipe
natd     492    root  cwd  VDIR  0,69    1024      2     /
natd     492    root  rtd  VDIR  0,69    1024      2     /
natd     492    root  txt  VREG  0,69    23672    1052  /sbin/natd
natd     492    root  txt  VREG  0,69    161596   3910  /libexec/ld-elf.so.1
natd     492    root  txt  VREG  0,69    48296    50232 /lib/libalias.so.5
natd     492    root  txt  VREG  0,69    930792   50224 /lib/libc.so.6
natd     492    root  0u   VCHR  0,9     0t0      9     /dev/null
natd     492    root  1u   VCHR  0,9     0t0      9     /dev/null
natd     492    root  2u   VCHR  0,9     0t0      9     /dev/null
natd     492    root  3u   IPv4  0xc329d000 0t0      DIVERT *:8668
natd     492    root  4u   rte   0xc2fcd460 0t0      ICMP  *:8668
natd     492    root  5u   IPv4  0xc329f000 0t0      ICMP  *:8668
natd     492    root  6u   unix  0xc336e858 0t0      ->(none)
rpcbind  825    root  cwd  VDIR  0,69    1024      2     /
rpcbind  825    root  rtd  VDIR  0,69    1024      2     /
rpcbind  825    root  txt  VREG  0,74    39596    1254758 /usr/sbin/rpcbind
rpcbind  825    root  txt  VREG  0,69    161596   3910  /libexec/ld-elf.so.1
rpcbind  825    root  txt  VREG  0,74    27812    377523 /usr/lib/libwrap.so.4
rpcbind  825    root  txt  VREG  0,69    46168    50231  /lib/libutil.so.5
```



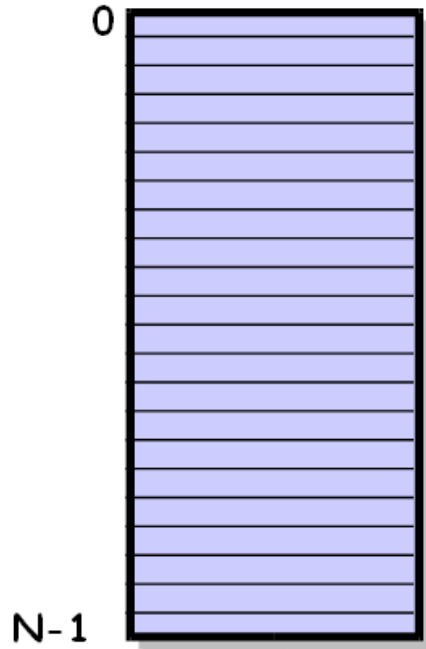
Ιδεατές & πραγματικές διευθύνσεις

- Τα συστήματα ιδεατής μνήμης καλύπτουν τις ανάγκες των διεργασιών μέσω της ψευδαίσθησης ότι έχουν στη διάθεσή τους περισσότερη κύρια μνήμη από όση διαθέτει το υπολογιστικό σύστημα.
- Έτσι υπάρχουν **δύο τύποι διευθύνσεων** στα συστήματα ιδεατής μνήμης :
 - Αυτές στις οποίες αναφέρονται οι διεργασίες (**ιδεατές ή λογικές ή εικονικές διευθύνσεις** – virtual addresses).
 - Αυτές που είναι διαθέσιμες στην κύρια μνήμη (**φυσικές ή πραγματικές διευθύνσεις** – real addresses).

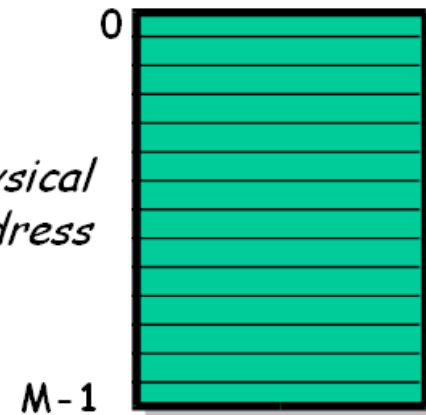



Εικονική και φυσική μνήμη

*virtual memory
(data names)*



*physical memory
(data locations)*



virtual address  *physical address*

translation



Λογικές, σχετικές και φυσικές διευθύνσεις

- Μια **λογική διεύθυνση** είναι μια αναφορά σε μια θέση μνήμης ανεξάρτητα από τη φυσική δομή και οργάνωση της μνήμης, καθώς και από την τρέχουσα εκχώρηση των δεδομένων στη μνήμη. Οι compilers παράγουν κώδικα στον οποίο όλες οι αναφορές μνήμης είναι λογικές διευθύνσεις. Οι λογικές διευθύνσεις παράγονται από τον επεξεργαστή. Πριν πραγματοποιηθεί η πρόσβαση στην κύρια μνήμη γίνεται μετάφραση στη φυσική διεύθυνση.
- Μια **σχετική διεύθυνση** είναι ένα παράδειγμα λογικής διεύθυνσης, στο οποίο η διεύθυνση εκφράζεται ως μια θέση σχετική με ένα γνωστό σημείο του προγράμματος (π.χ. η αρχή).
- Μια **φυσική ή απόλυτη διεύθυνση** είναι μια φυσική θέση στην κύρια μνήμη.



Μετάφραση των λογικών σε φυσικών διευθύνσεων μνήμης (1/2)

- Οι σχετικές διευθύνσεις είναι ο πλέον συνηθισμένος τύπος λογικών διευθύνσεων που χρησιμοποιούνται στα εκτελέσιμα αρχεία (modules προγραμματισμού). Αυτά φορτώνονται στην κύρια μνήμη με όλες τις αναφορές προς τη μνήμη σε σχετική μορφή.
- Οι φυσικές διευθύνσεις «υπολογίζονται» καθώς εκτελούνται οι εντολές.
- Για να υπάρχει επαρκής απόδοση η μετάφραση από σχετικές σε φυσικές διευθύνσεις **γίνεται από το υλικό.**

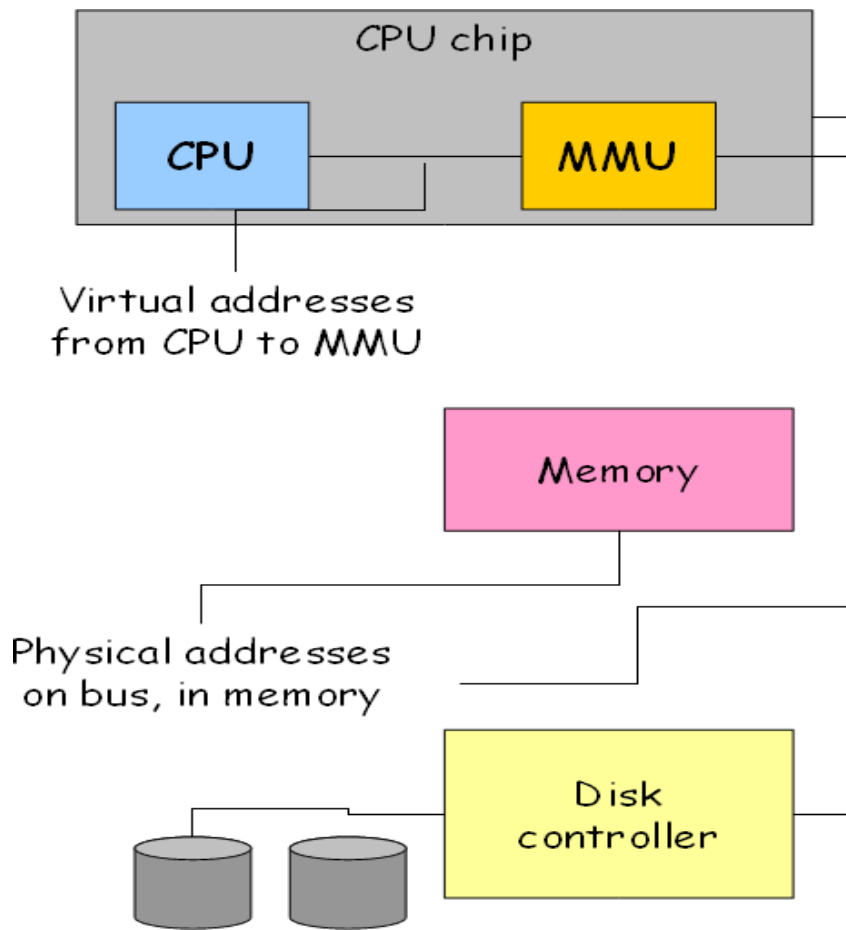


Μετάφραση των λογικών σε φυσικών διευθύνσεων μνήμης (2/2)

- Τα προγράμματα των χρηστών διαχειρίζονται μόνον λογικές διευθύνσεις και **δεν «βλέπουν» ποτέ τις πραγματικές φυσικές διευθύνσεις.**
- Η μονάδα διαχείρισης μνήμης (Memory Management Unit - MMU) είναι μια συσκευή που αντιστοιχεί τις εικονικές σε φυσικές διευθύνσεις.



Η θέση και η λειτουργία της MMU



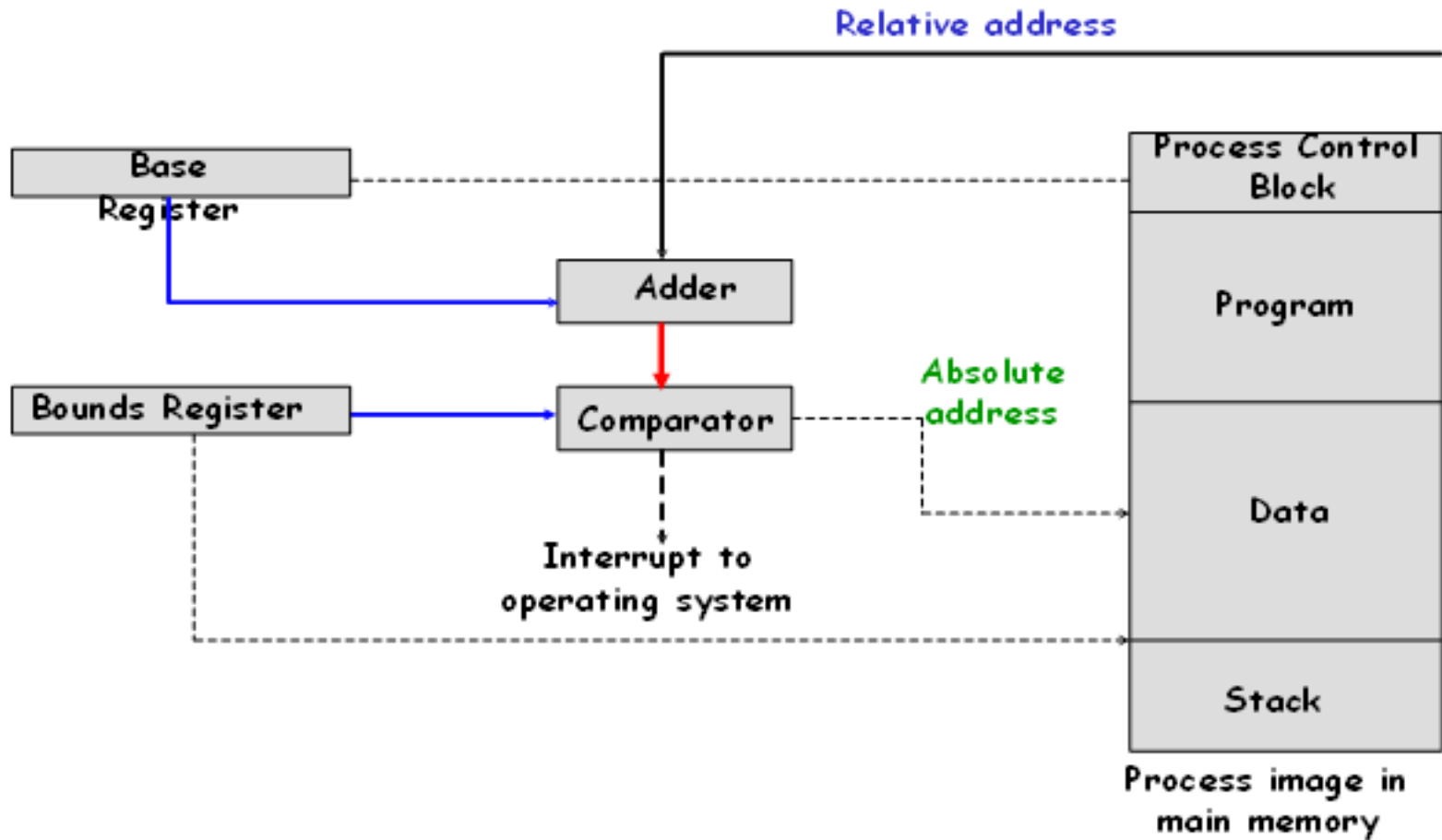
Οι καταχωρητές βάσης και ορίου χρησιμοποιούνται για τη μετάφραση διευθύνσεων και την προστασία

- Όταν μια διεργασία περιέρχεται σε κατάσταση εκτέλεσης ένας καταχωρητής βάσης της CPU (**base register**) φορτώνεται με τη φυσική διεύθυνση εκκίνησης της διεργασίας.
- Ένας καταχωρητής ορίου (**bound or limit register**) φορτώνεται με τη φυσική διεύθυνση τερματισμού της διεργασίας.
- Όταν προκύψει μια σχετική διεύθυνση, αυτή προστίθεται στο περιεχόμενο του καταχωρητή βάσης για να αποκτηθεί η φυσική διεύθυνση που συγκρίνεται με το περιεχόμενο του καταχωρητή ορίου.
- Αν η διεύθυνση είναι εκτός ορίων, δημιουργείται μια διακοπή (interrupt) προς το Λ.Σ.
- Αυτό παρέχει προστασία υλικού: κάθε διεργασία μπορεί να έχει πρόσβαση μόνον στη δική της εικόνα (process image).

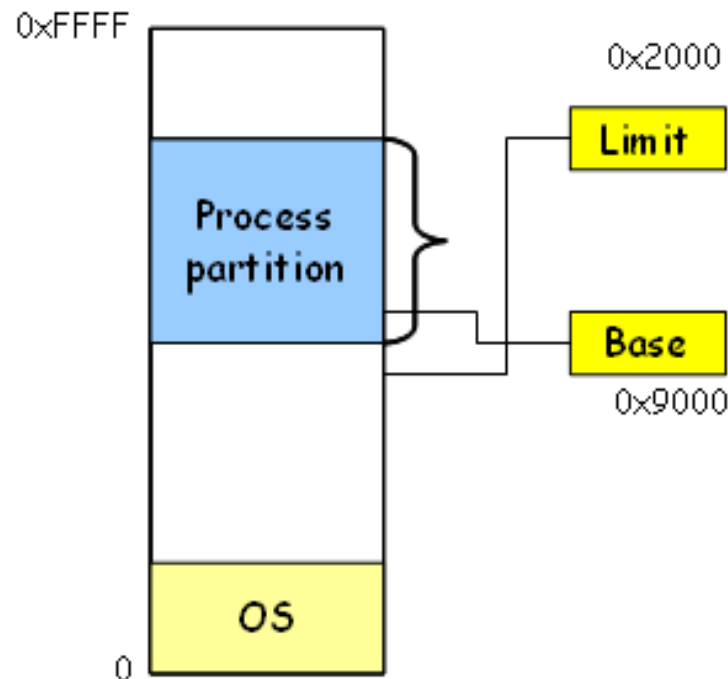
Συμπέρασμα : Το υλικό μέρος του επεξεργαστή και το λογισμικό του Λ.Σ. μεταφράζουν τις αναφορές στη μνήμη που περιέχει ο κώδικας του προγράμματος σε πραγματικές διευθύνσεις της φυσικής μνήμης.



Υποστήριξη υλικού μέρους για τη μετατόπιση



Καταχωρητές βάσης και ορίου



Λογική διεύθυνση: 0x1204
φυσική διεύθυνση:
 $0x1204 + 0x9000 = 0xa204 < 0x2000 \rightarrow$ δεν υπάρχει σφάλμα



Λογική οργάνωση

- Η κύρια μνήμη σε ένα υπολογιστικό σύστημα **οργανώνεται ως ένας γραμμικός, μονοδιάστατος χώρος διευθύνσεων.**
- Η δευτερεύουσα μνήμη, σε φυσικό επίπεδο, οργανώνεται με παρόμοιο τρόπο.
- Τα προγράμματα οργανώνονται και γράφονται σε ενότητες (modules).
- Οι ενότητες αυτές γράφονται και μεταφράζονται ανεξάρτητα.
- Στις ενότητες δίνονται διαφορετικοί βαθμοί προστασίας (read-only, execute-only).
- Οι ενότητες μπορούν να διαμοιράζονται μεταξύ των διεργασιών.



Προστασία και κοινή χρήση της μνήμης

- Το ΛΣ πρέπει να διατηρεί ξεχωριστά τη μνήμη για κάθε διεργασία.
 - **Προστασία** μιας διεργασίας από άλλες που θέλουν να διαβάσουν ή να γράψουν στη δική της περιοχή μνήμης
 - Προστασία μιας διεργασίας από την τροποποίηση της δικής της μνήμης με ανεπιθύμητο τρόπο (πχ γράφοντας στο τμήμα κώδικα).
- Το Λ.Σ. πρέπει να επιτρέπει σε **πολλές διεργασίες να έχουν πρόσβαση** στην ίδια περιοχή της μνήμης.
 - Είναι προτιμότερο να επιτρέπεται η πρόσβαση σε μια διεργασία (σε ένα άτομο) στο ίδιο αντίγραφο του προγράμματος από το να υπάρχει ένα αντίγραφο για κάθε μια διεργασία.
- Για να ικανοποιηθούν αυτές τις ανάγκες χρησιμοποιείται ως εργαλείο ή τμηματοποίηση της ιδεατής μνήμης.



Τμηματοποίηση ιδεατής μνήμης

- Η ιδεατή μνήμη χρησιμοποιεί **τρεις** βασικές τεχνικές:
 - **Σελιδοποίηση** (paging).
 - **Κατάτμηση** (segmentation) (τείνει να εγκαταλειφθεί!!!).
 - **Κατάτμηση με σελιδοποίηση** (segmentation with paging).

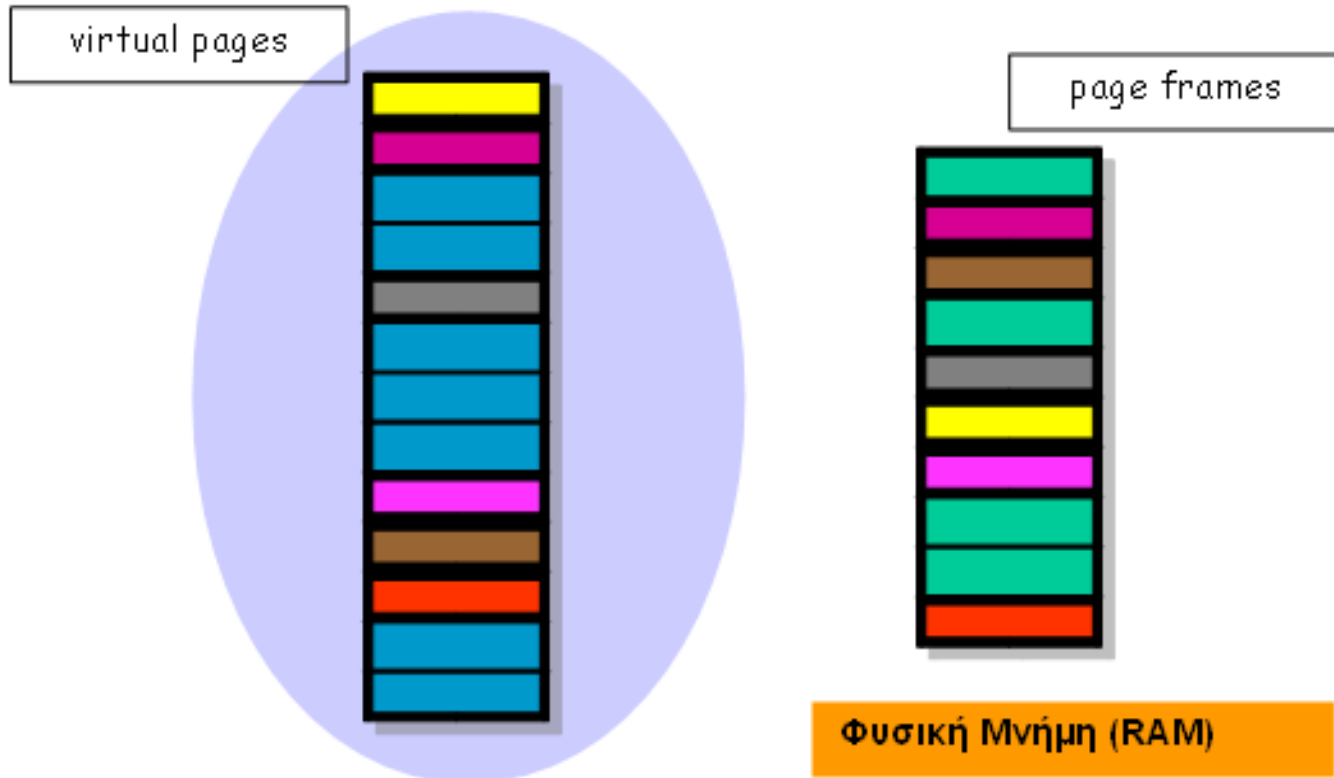


Σελιδοποίηση της μνήμης

- Η κατάτμηση της μνήμης σε μικρά ίσου μεγέθους τμήματα (blocks, chunks) και η διαίρεση κάθε διεργασίας σε τμήματα του ίδιου μεγέθους.
- Τα τμήματα μιας διεργασίας λέγονται σελίδες(pages) και τα τμήματα της μνήμης πλαίσια (frames).
- Ο εικονικός χώρος διευθύνσεων διαμοιράζεται σε σελίδες σταθερού μεγέθους, ενώ η φυσική μνήμη διαμοιράζεται σε πλαίσια σελίδας (page frames) (μεγέθους ίδιου με τη σελίδα).
- Μια σελίδα μπορεί να τοποθετηθεί σε οποιοδήποτε πλαίσιο σελίδας.

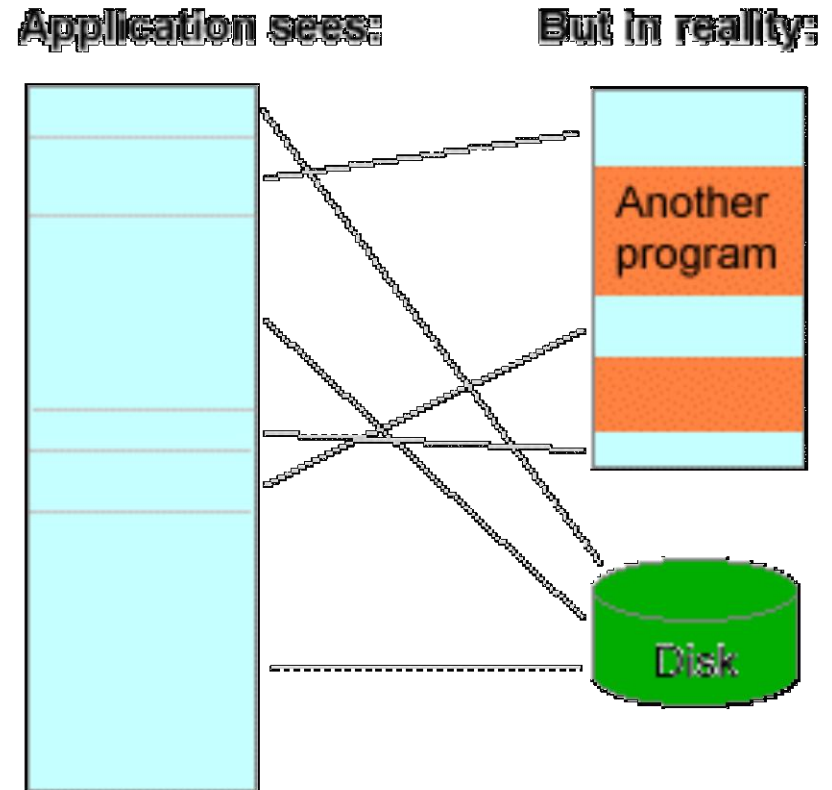


Οι σελίδες βρίσκονται στην εικονική μνήμη ενώ τα πλαίσια στη φυσική μνήμη



Virtual Memory VS swapping

- Η ιδεατή μνήμη δεν είναι μόνο η χρήση χώρου στο δίσκο για να επεκταθεί το φυσικό μέγεθος της μνήμης (αυτό έρχεται ως αποτέλεσμα).
- Η ιδεατή μνήμη «ξεγελάει» τα προγράμματα ώστε να θεωρούν ότι χρησιμοποιούν ένα μεγάλο συνεχόμενο εύρος διευθύνσεων (ενώ δε συμβαίνει αυτό).



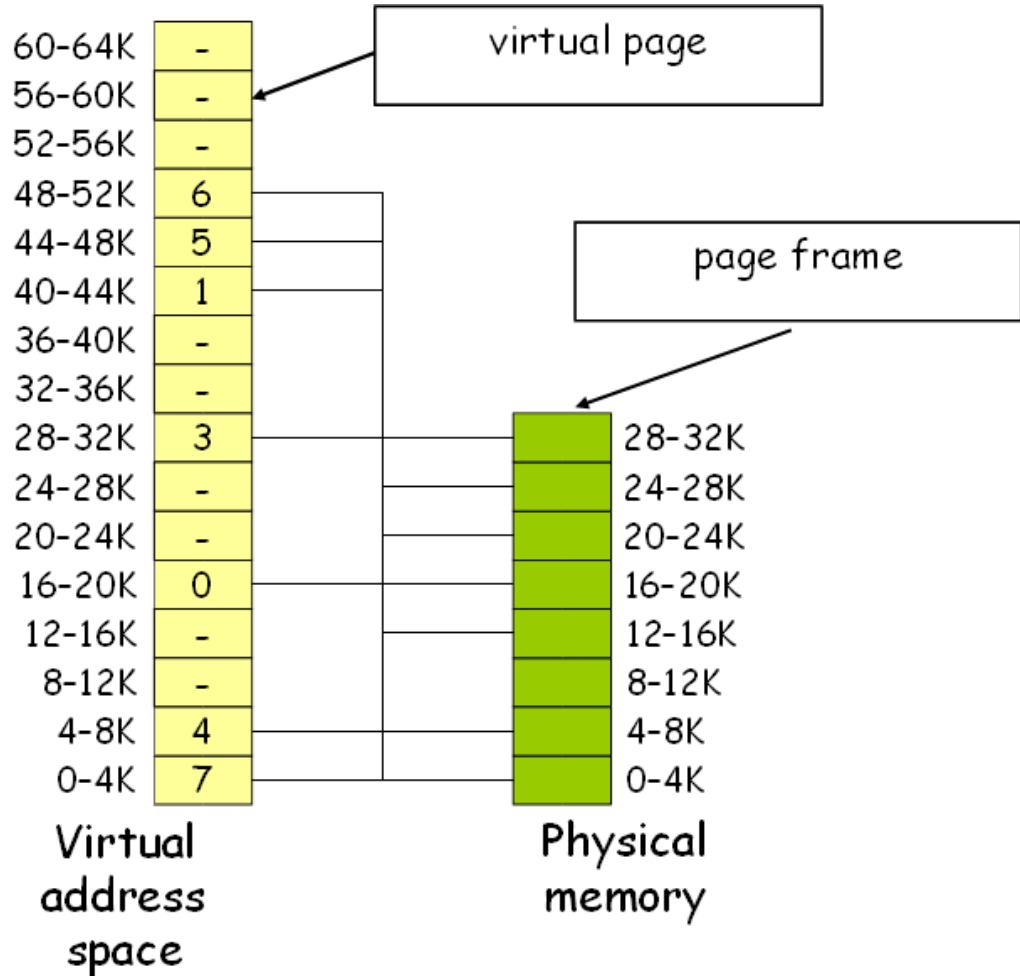
Αντιστοίχιση πλαισίων σε σελίδες

**Windows XP
Professional**

Virtual address space: 2GB
+ 2GB(OS)

FreeBSD

Virtual address space: 3GB
+ 1GB (OS)

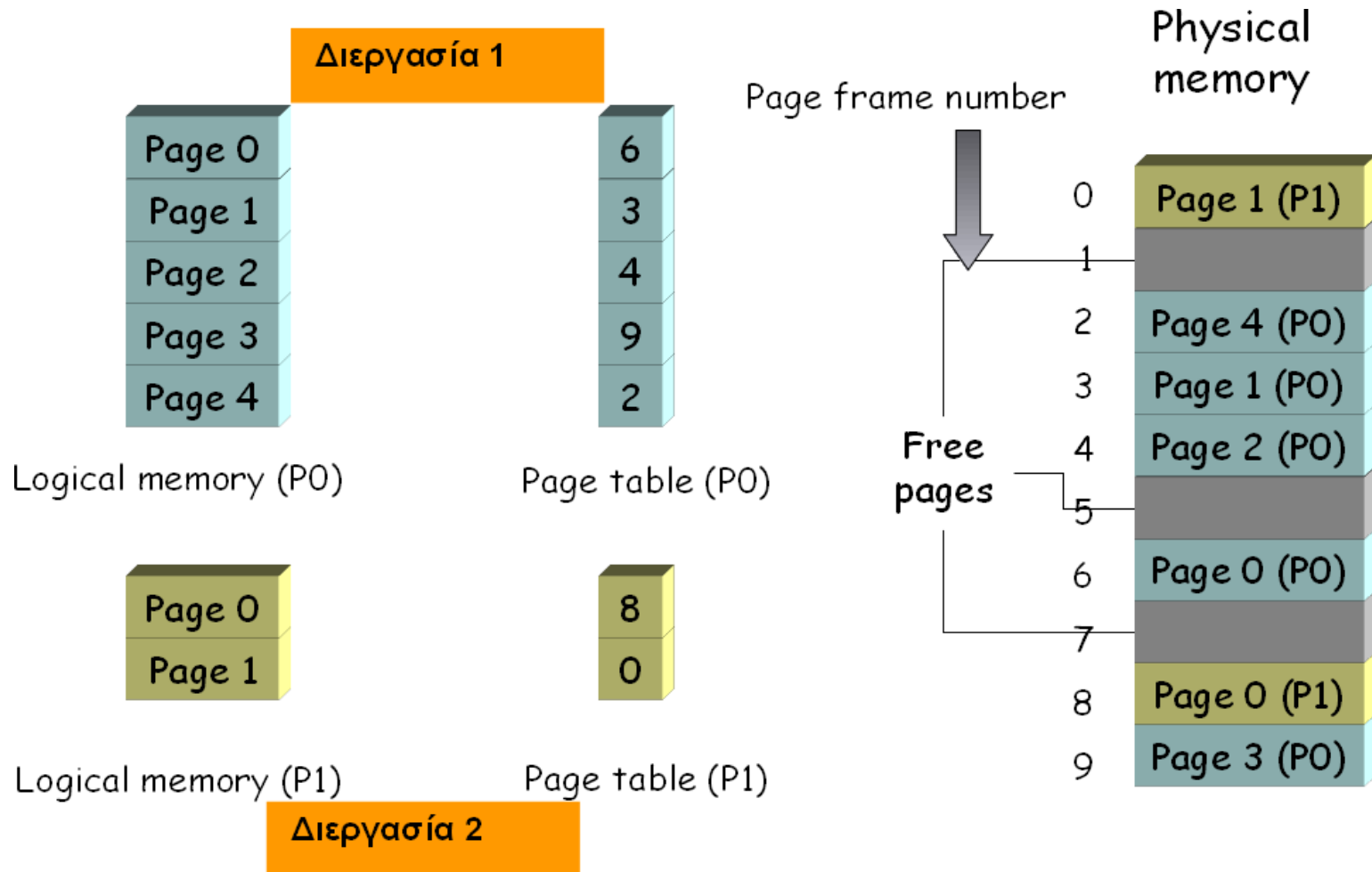


Ο πίνακας σελίδων της διεργασίας

- Το Λ.Σ. διατηρεί:
 - ένα πίνακα σελίδων (**page table**) για κάθε διεργασία που περιέχει τη θέση πλαισίου για κάθε σελίδα της διεργασίας.
 - μια λίστα των ελεύθερων πλαισίων, με όλα τα πλαίσια στην κύρια μνήμη που είναι ελεύθερα και διαθέσιμα για τις σελίδες.
- Η διεύθυνση μνήμης μέσα στο πρόγραμμα (λογική διεύθυνση) αποτελείται από έναν **αριθμό σελίδας** και μια **μετατόπιση** (offset) εντός της σελίδας
- Ο επεξεργαστής χρησιμοποιεί τον πίνακα σελίδων για να παράγει τη **φυσική διεύθυνση** (**αριθμός πλαισίου, μετατόπιση**) που αντιστοιχεί σε μια **λογική διεύθυνση** (**αριθμός σελίδας, μετατόπιση**).



Αντιστοίχιση σελίδων διεργασιών με ελεύθερα πλαίσια



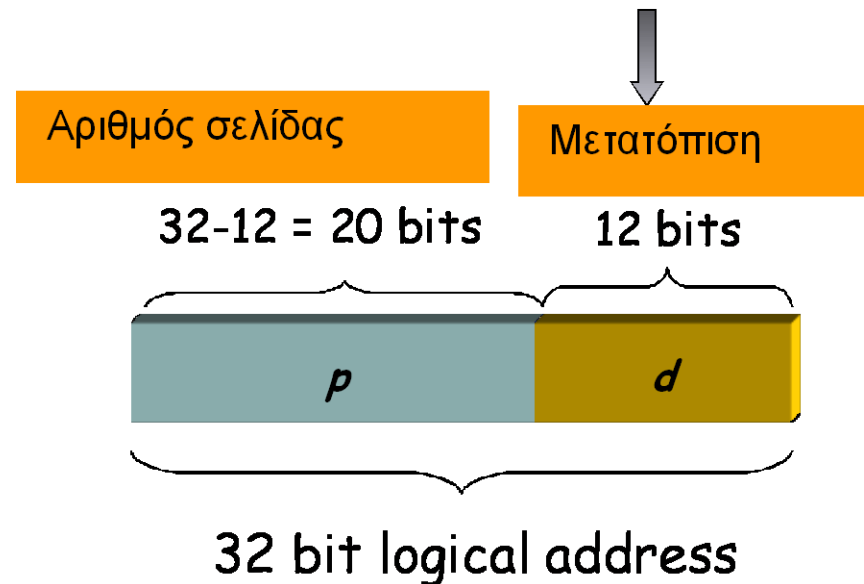
Αντιστοίχιση λογικών σε φυσικές διευθύνσεις

- Διαίρεση της διεύθυνσης από τη CPU σε δύο τμήματα:
 - Αριθμός σελίδας (p).
 - Μετατόπιση στη σελίδα (d).
- Αριθμός σελίδας:
 - Δείκτης στον πίνακα σελίδων.
 - Ο πίνακας σελίδων περιέχει τη διεύθυνση βάσης της σελίδας στη φυσική μνήμη.
- Μετατόπιση στη σελίδα:
 - Προστίθεται στη διεύθυνση βάσης για να βρεθεί η πραγματική διεύθυνση στη φυσική μνήμη.
- Μέγεθος σελίδας = 2^d bytes.

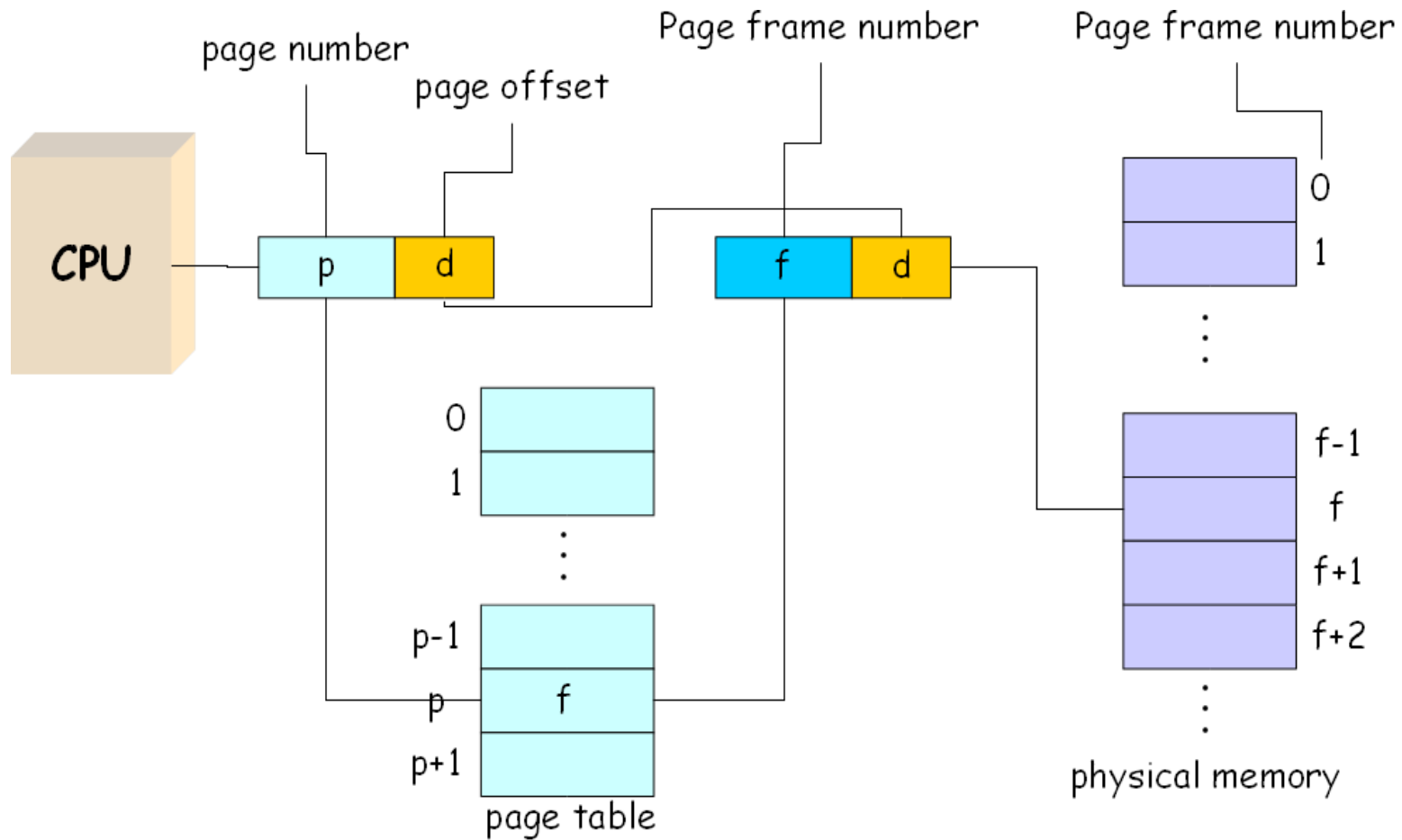
παράδειγμα:

- 4 KB (=4096 byte) pages
- 32 bit logical addresses

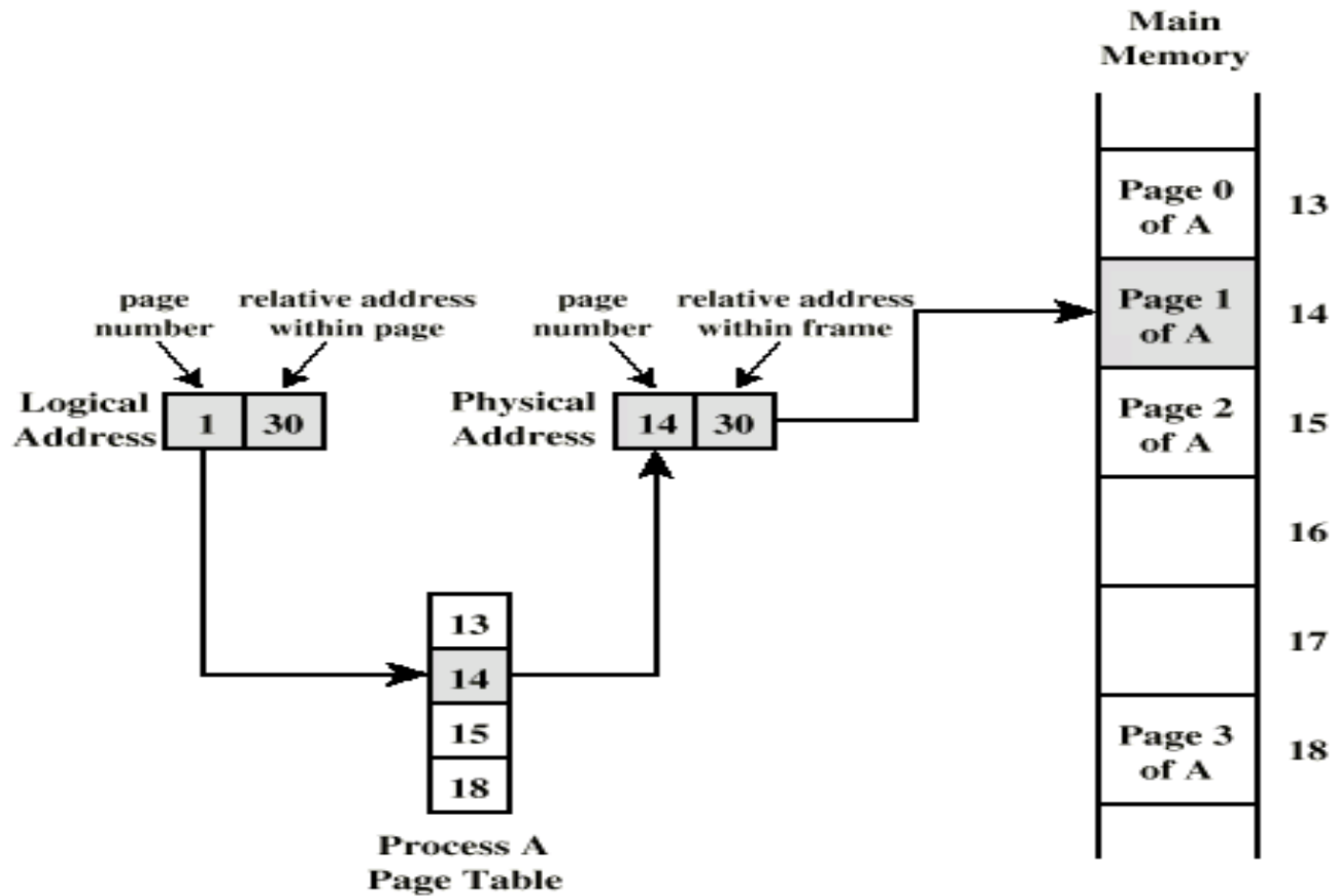
$$2^d = 4096 \longrightarrow d = 12$$



Αρχιτεκτονική μετάφρασης της διεύθυνσης



Παράδειγμα - 1



Παράδειγμα - 2

Page table

	present bit
000	0
000	0
000	0
000	0
111	1
000	0
101	1
000	0
000	0
000	0
011	1
100	1
000	1
110	1
001	1
010	1

4-bit index
into page table
virtual page = 0010 = 2

Out going physical address



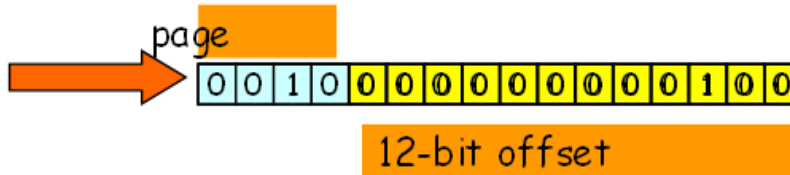
(0x6004, 24580)



Παράδειγμα:

- 4 KB pages (12-bit offsets)
- 16 bit virtual address space → 16 pages (4-bit index)
- 8 physical pages (3-bit index) [=32 MB 8x4096]

Incoming virtual address
(0x2004, 8196)



Τμηματοποίηση σταθερού μεγέθους

Η σελιδοποίηση είναι ανάλογη με την τμηματοποίηση σταθερού μεγέθους, με τις εξής διαφορές:

- 1. Τα τμήματα δεν χρειάζεται να είναι συνεχόμενα.**
- 2. Τα τμήματα είναι αρκετά μικρά.**
- 3. Ένα πρόγραμμα μπορεί να απασχολεί περισσότερα από ένα τμήματα.**

Η σπατάλη μνήμης οφείλεται στον εσωτερικό κατακερματισμό που είναι κλάσμα της τελευταίας σελίδας της διεργασίας.

Εξωτερικός κατακερματισμός δεν υπάρχει. (Γιατί;)



Χρησιμοποιούνται δυνάμεις του 2

- Το μέγεθος σελίδας και πλαισίου είναι δύναμη του 2 (συνήθως μεταξύ 512 bytes και 8192 bytes).
- Αποδεικνύεται ότι η σχετική διεύθυνση που ορίζεται σε σχέση με την αρχή του προγράμματος και η λογική διεύθυνση που εκφράζεται ως ένας αριθμός σελίδας και μετατόπιση, είναι ΙΔΙΕΣ.
- Οφέλη της χρήσης μεγεθών σελίδας που είναι δυνάμεις του 2:
 - Το σχήμα της λογικής διευθυνσιοδότησης δεν είναι εμφανές στον προγραμματιστή.
 - Είναι εύκολη η εφαρμογή μιας συνάρτησης σε επίπεδο υλικού μέρους που θα πραγματοποιεί δυναμική μετάφραση των διευθύνσεων κατά τη στιγμή της εκτέλεσης.



Απαιτείται συμβιβασμός στο μέγεθος σελίδας

- Μικρό μέγεθος σελίδας.

Πλεονεκτήματα:

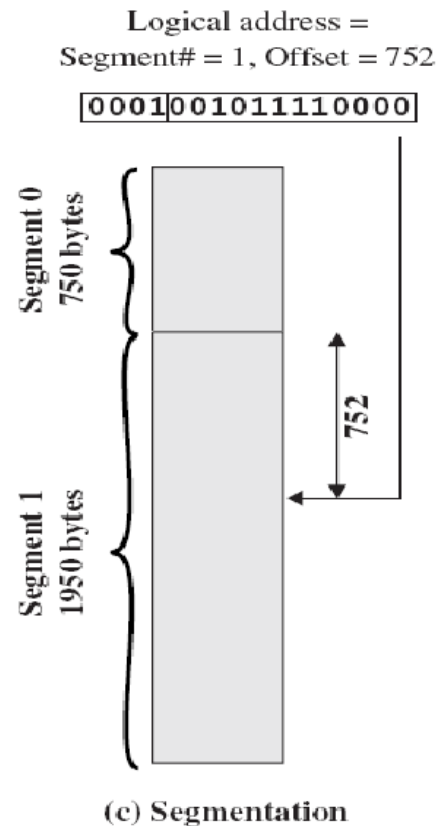
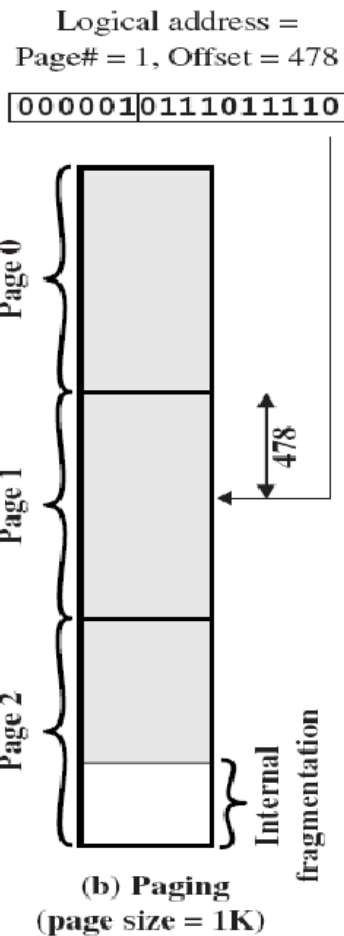
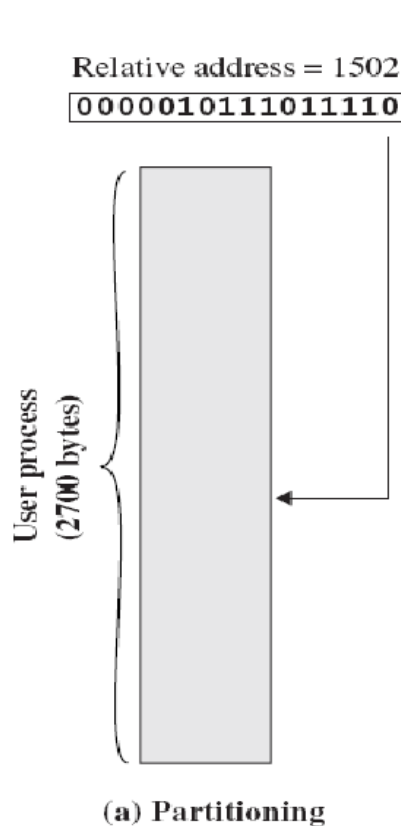
- Λιγότερος εσωτερικός κατακερματισμός.
- Καλύτερο ταίριασμα για διάφορες δομές δεδομένων και τμήματα κώδικα.
- Λιγότερο μη χρησιμοποιούμενο πρόγραμμα στη μνήμη.

Μειονεκτήματα:

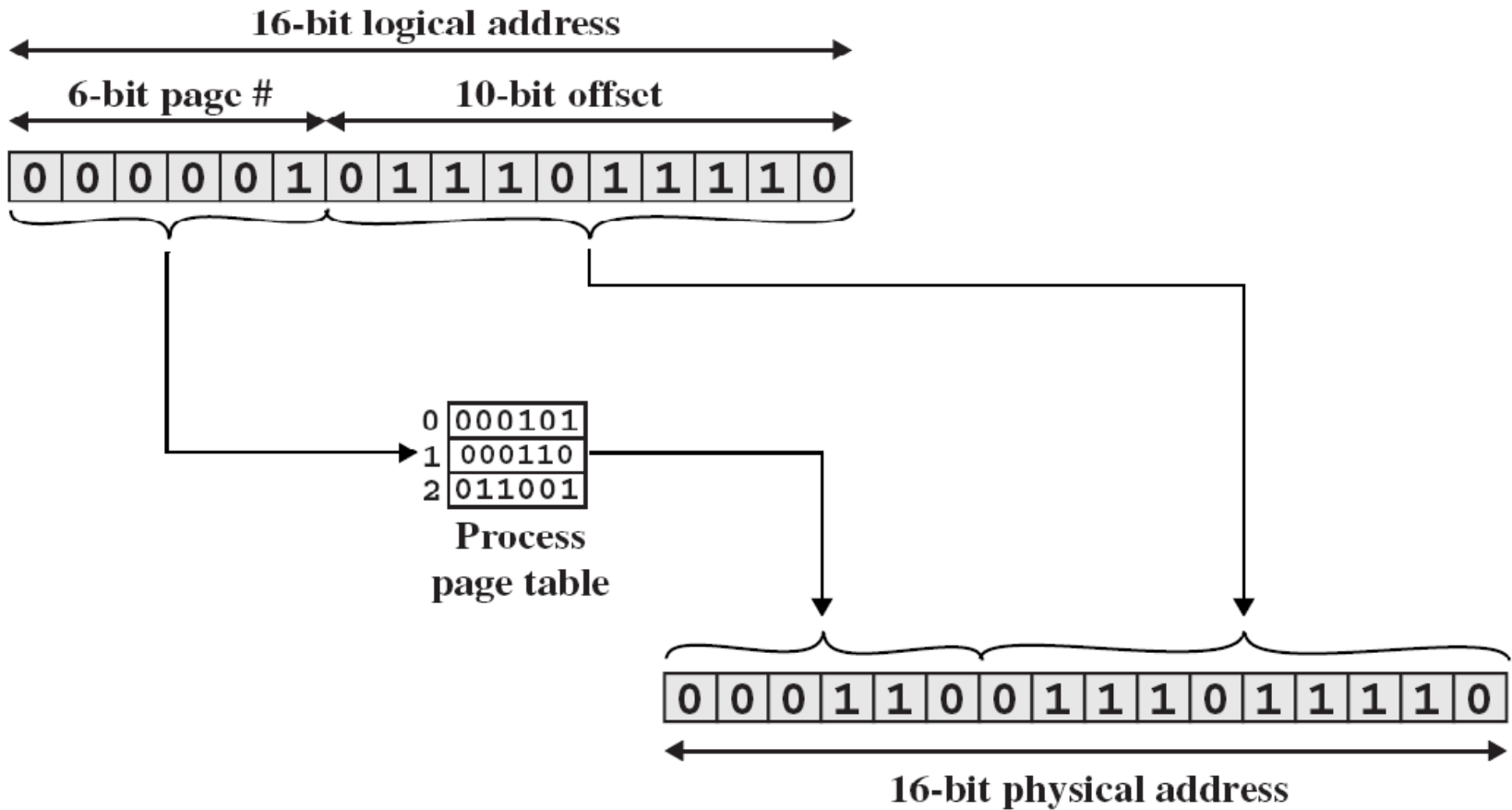
- Τα προγράμματα χρειάζονται πολλές σελίδες και μεγαλύτερους πίνακες σελίδων.
- Απαιτούνται περισσότερες καταχωρήσεις στον πίνακα αντιστοίχισης.



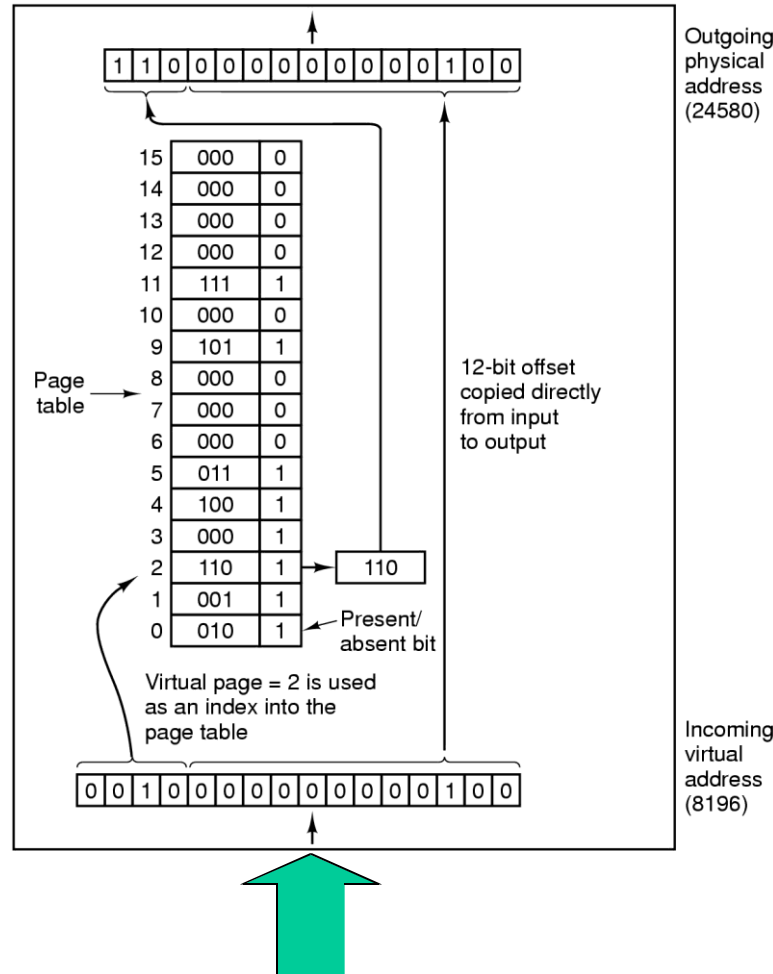
Λογικές διευθύνσεις - παραδείγματα



Σελιδοποίηση



Εσωτερική λειτουργία της MMU με 16 σελίδες των 4 KB



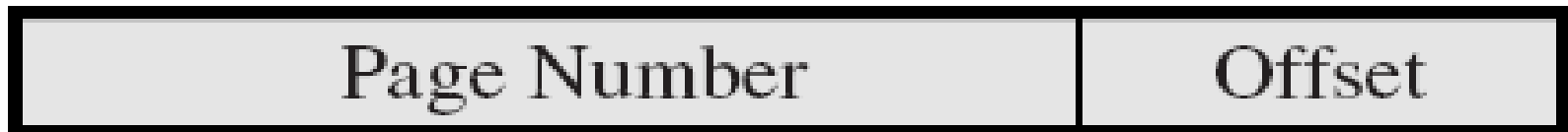
Κάθε διεργασία έχει το δικό της εικονικό χώρο διευθύνσεων

- Κάθε διεργασία έχει τον δικό της εικονικό χώρο διευθύνσεων, άρα και τον δικό της πίνακα σελίδων που αποθηκεύεται στην κύρια μνήμη.
- **Κάθε καταχώρηση (θέση) του πίνακα σελίδων περιέχει τον αριθμό πλαισίου της αντίστοιχης σελίδας στην κεντρική μνήμη.**
- Επιπλέον απαιτούνται δυο bit:
 - ένα bit (**P**resent bit) που θα δείχνει αν μια σελίδα βρίσκεται στην κεντρική μνήμη ή όχι.
 - ένα bit (**M**odify bit) για να δείχνει αν η σελίδα έχει αλλάξει από την τελευταία φορά που φορτώθηκε στην κεντρική μνήμη. Αν δεν έχει γίνει αλλαγή, τότε η σελίδα δεν πρέπει να γραφεί στο δίσκο όταν χρειάζεται να γίνει εναλλαγή.

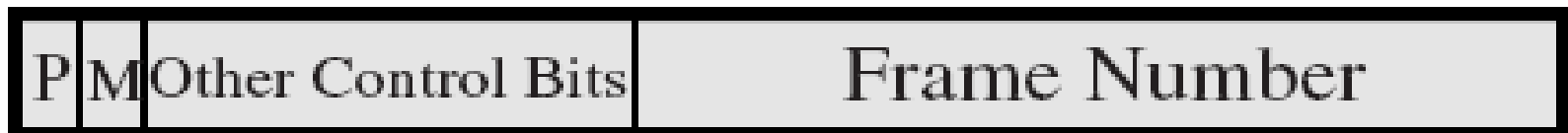


Είσοδοι πίνακα σελίδων

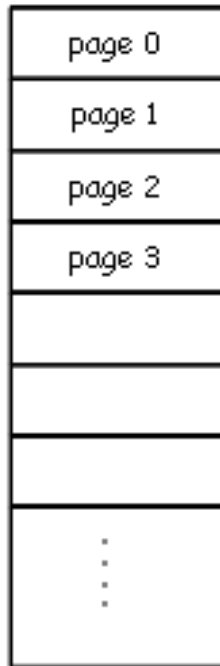
Virtual Address



Page Table Entry



Παράδειγμα

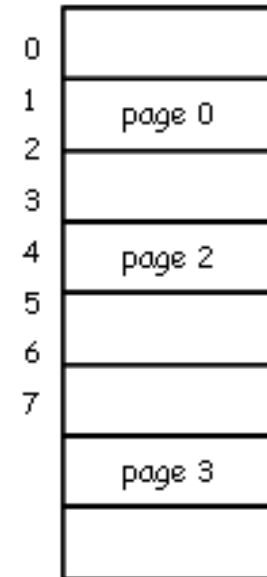


virtual memory

ρ

1	1
0	
1	3
1	6
	⋮

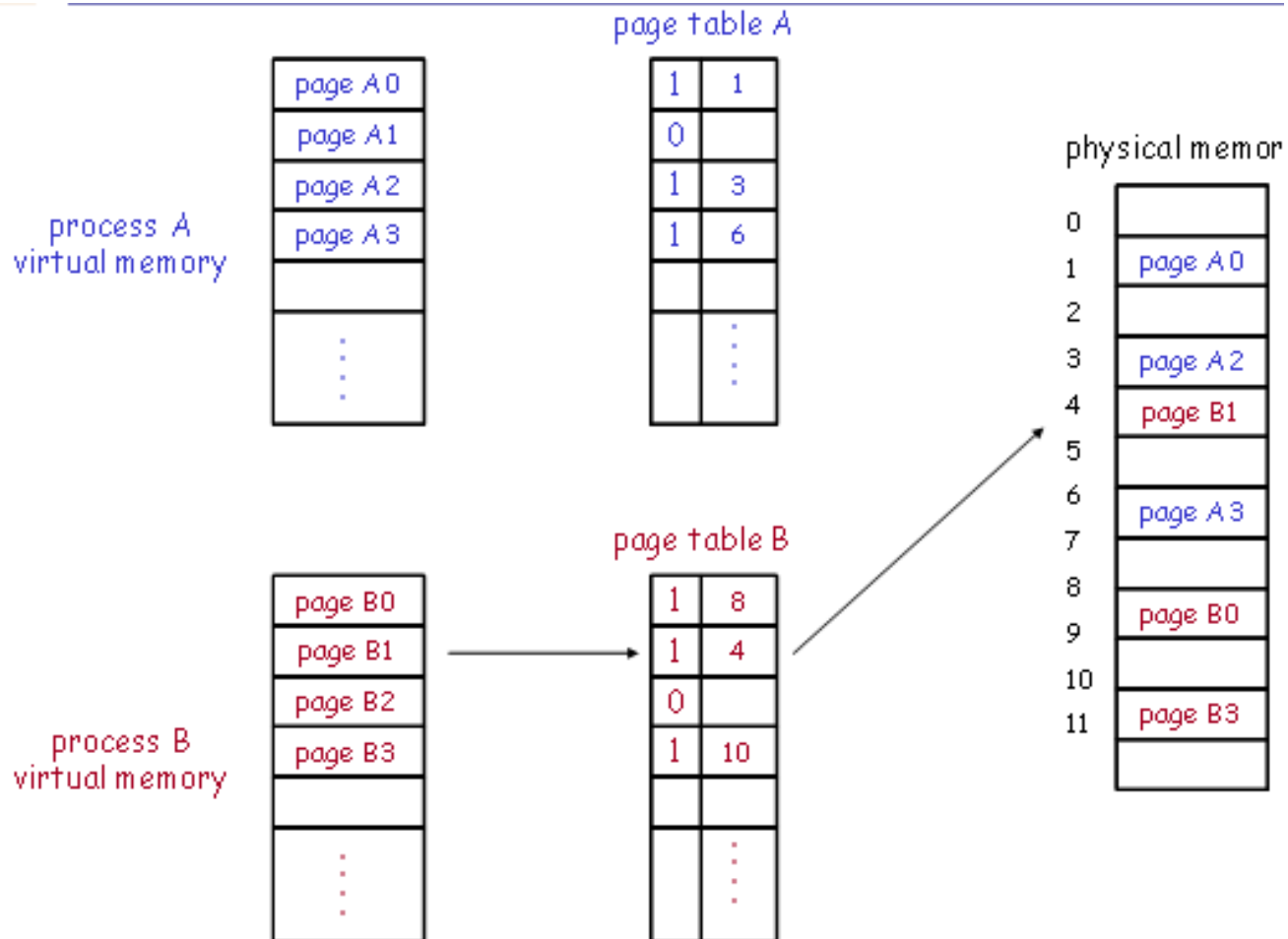
Page Table



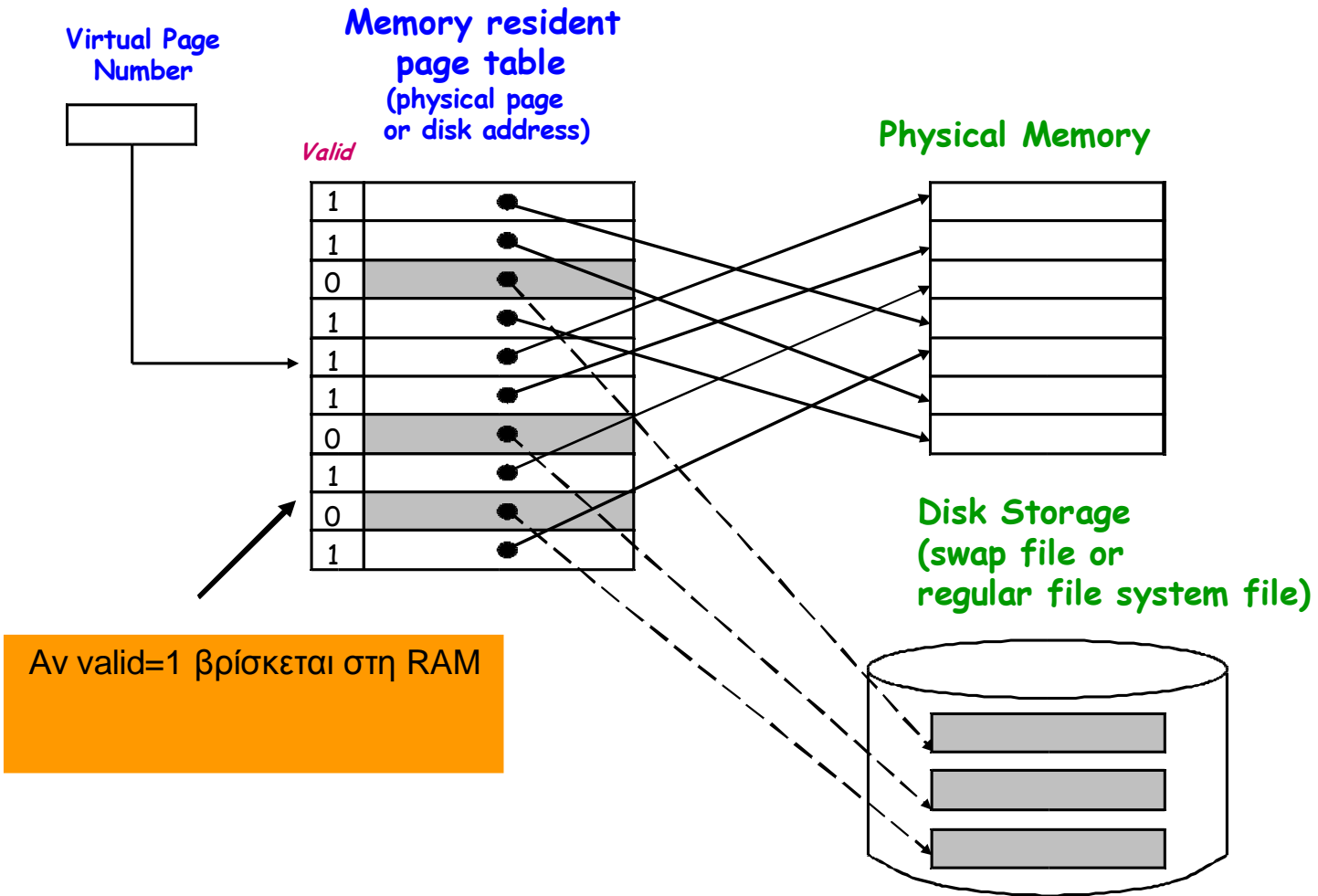
physical memory



Πολλαπλές διεργασίες στη φυσική μνήμη

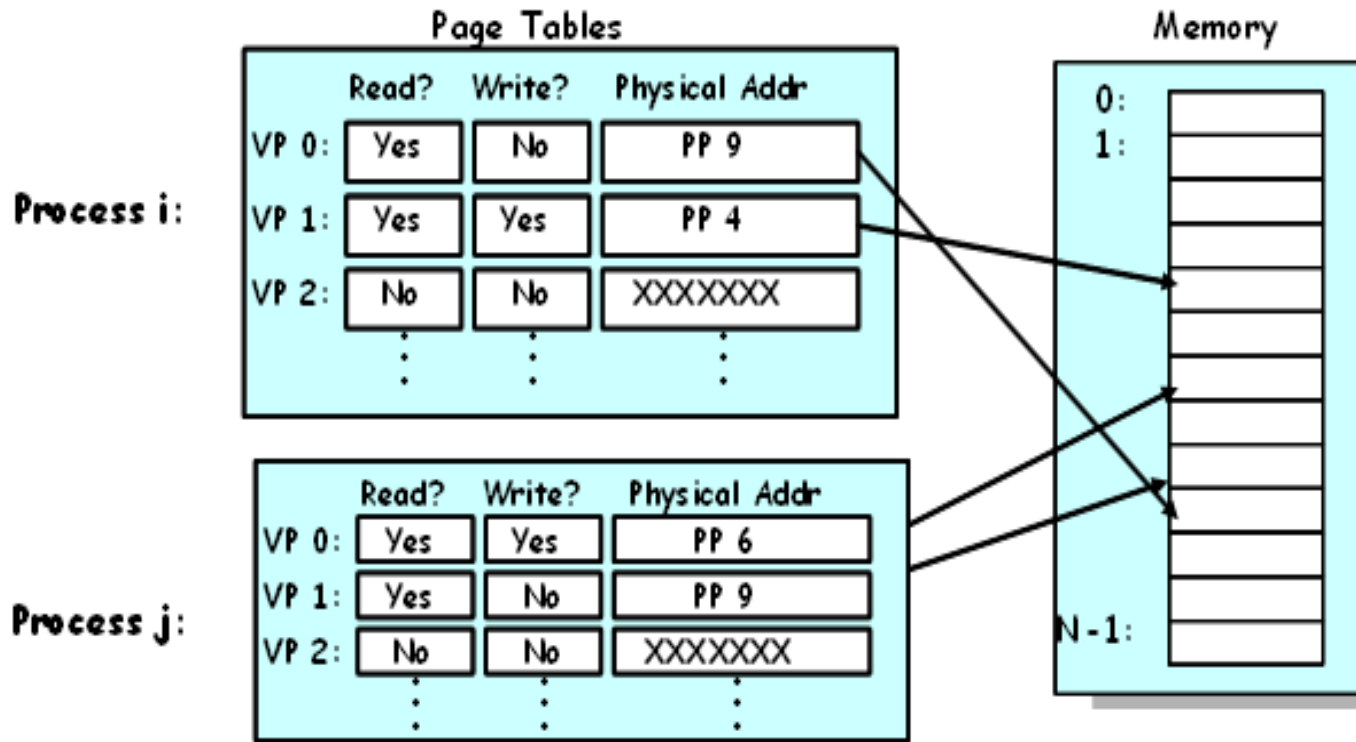


Πίνακες σελίδων – valid bit



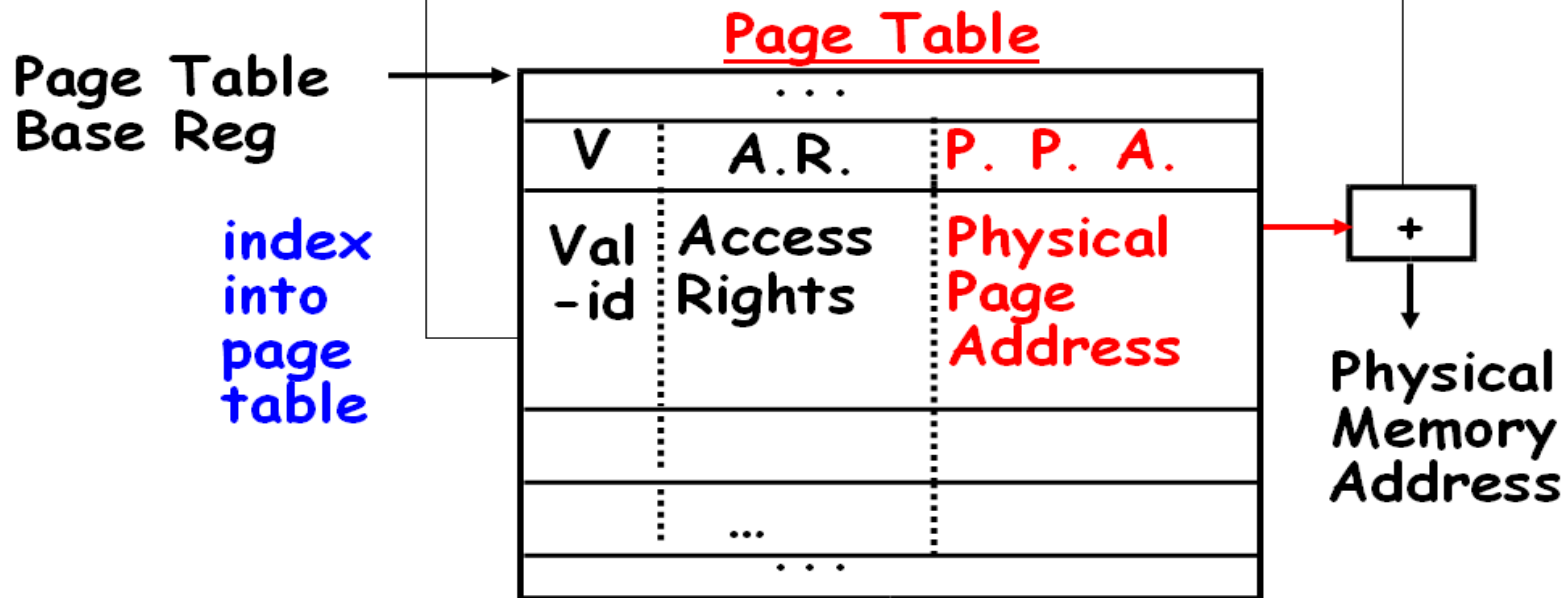
Προστασία

Κάθε είσοδος στον πίνακα σελίδων περιέχει πληροφορίες για τα δικαιώματα πρόσβασης (access rights).



Αντιστοίχιση διευθύνσεων : Πίνακας σελίδων

Virtual Address:



Ο πίνακας σελίδων βρίσκεται στη φυσική μνήμη



Πίνακες σελίδων

- Ο πίνακας σελίδων έχει μέγιστο πλήθος εισόδων, όσο και το πλήθος των εικονικών σελίδων της διεργασίας.
- Ο πίνακας σελίδων είναι μεταβλητού μήκους (το μήκος του εξαρτάται από το μέγεθος της διεργασίας) και πρέπει να βρίσκεται στην κύρια μνήμη για να είναι προσπελάσιμος.
- Ολόκληρος ο πίνακας σελίδων είναι πιθανόν να καταλαμβάνει πολύ μεγάλο μέρος της κεντρικής μνήμης.
- Το μέγεθος της μνήμης που αφιερώνεται στους πίνακες σελίδων μπορεί να γίνει απαράδεκτα μεγάλο.
- Μεγαλύτερο μέγεθος σελίδας είναι μια ενδεδειγμένη λύση;
- Τα περισσότερα συστήματα ιδεατής μνήμης αποθηκεύουν τους πίνακες των σελίδων στην ιδεατή μνήμη, αντί για τη φυσική. Όταν μια διεργασία εκτελείται, ένα τμήμα του πίνακα σελίδων της βρίσκεται στην κεντρική μνήμη.

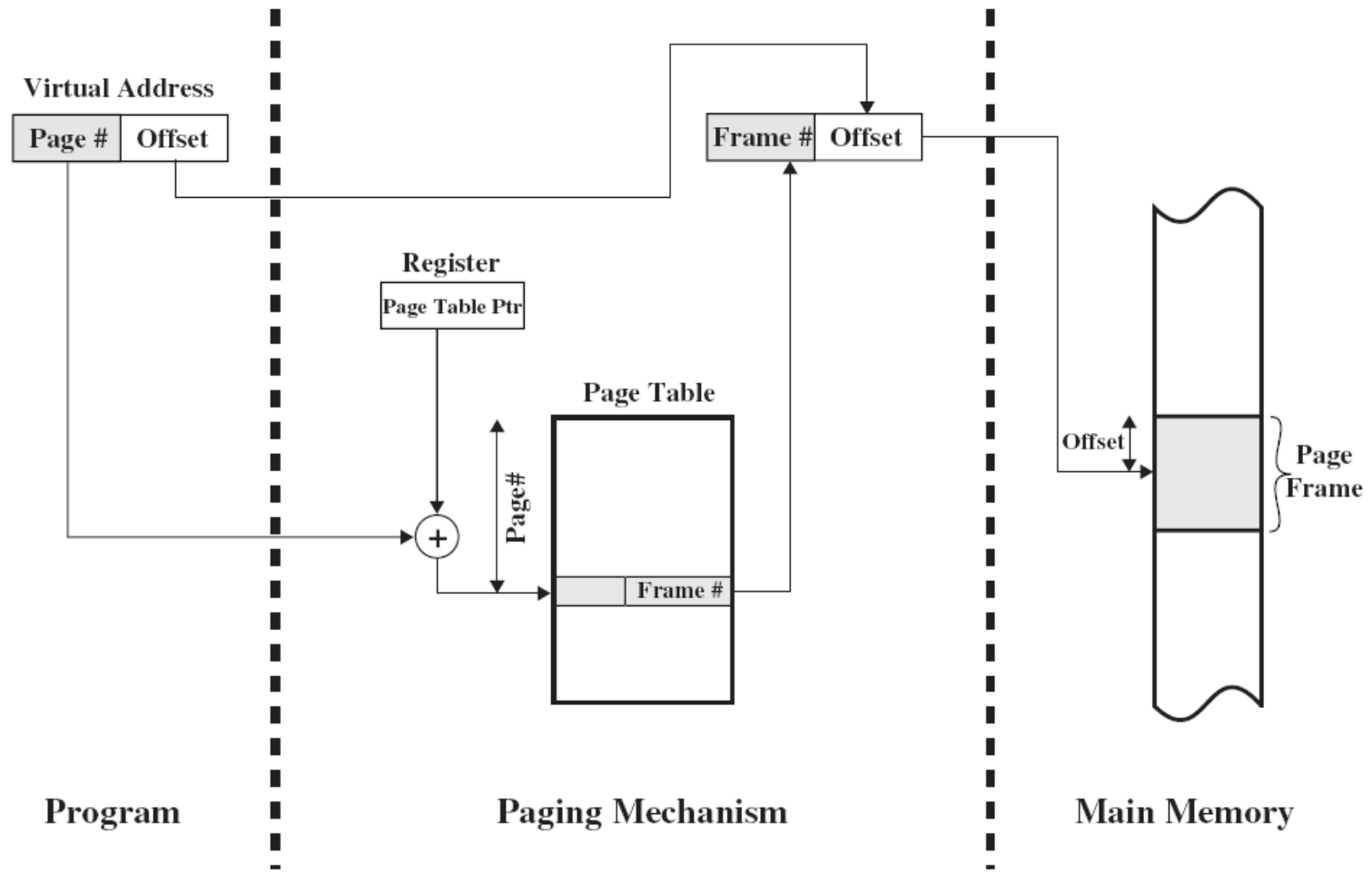


Καταχωρητές που χρησιμοποιούνται

- Κάθε πίνακας σελίδων χρησιμοποιεί τους καταχωρητές:
 - **Page Table Base Register (PTBR)**: κρατά την αρχική φυσική διεύθυνση του πίνακα σελίδων της εκτελούμενης διεργασίας.
 - **Page-table length register (PRLR)**: το υλικό ελέγχει αν ο αριθμός σελίδας βρίσκεται εντός του ορίου και το μέγεθος του πίνακα σελίδων περιορίζεται.
- Ορισμένοι επεξεργαστές (Pentium) χρησιμοποιούν πίνακες σελίδων πολλαπλών επιπέδων για να οργανώσουν καλύτερα μεγάλους πίνακες σελίδων.



Page tables Πολλαπλών Επιπέδων

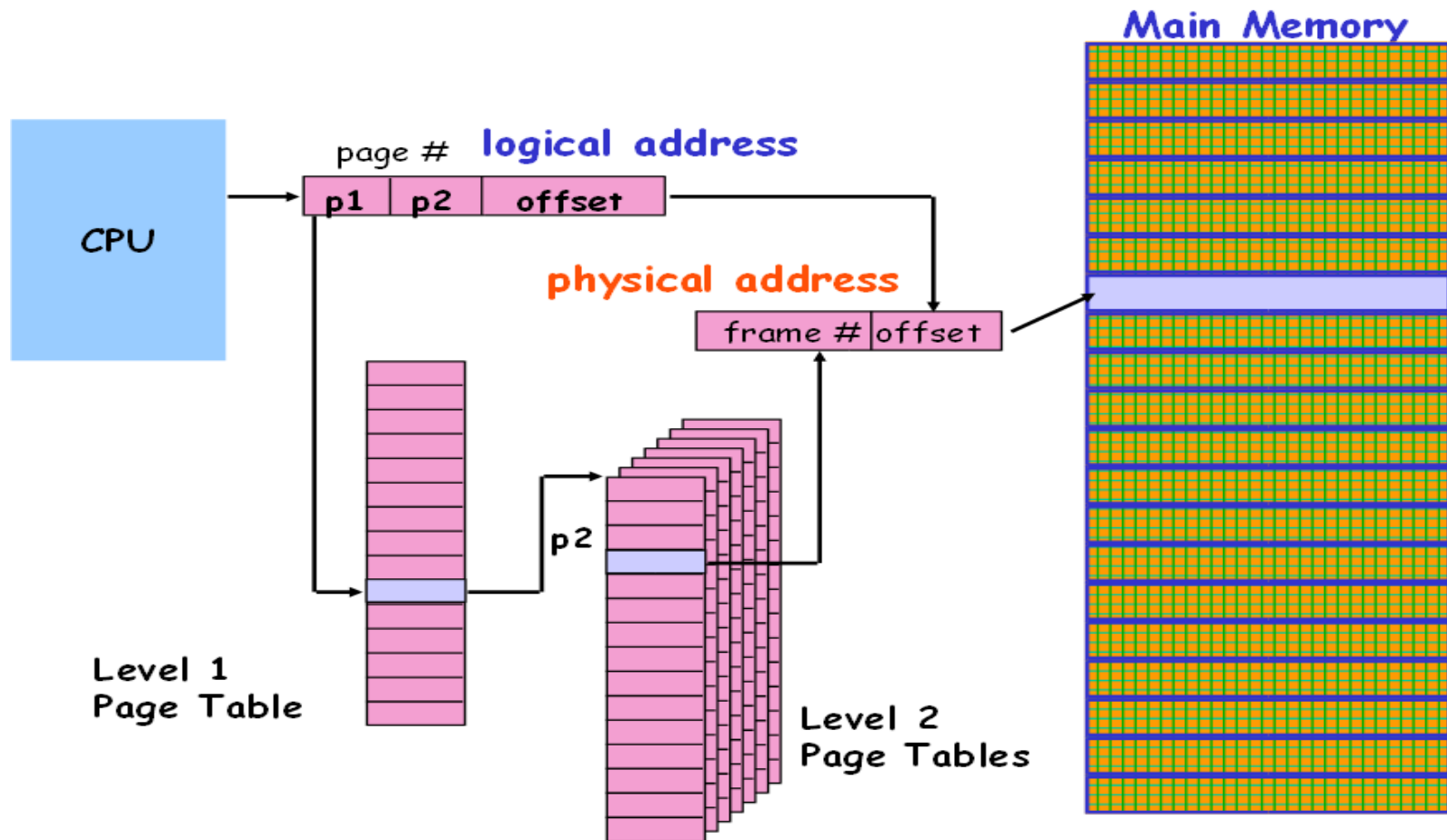


Παράδειγμα Πίνακα σελίδων

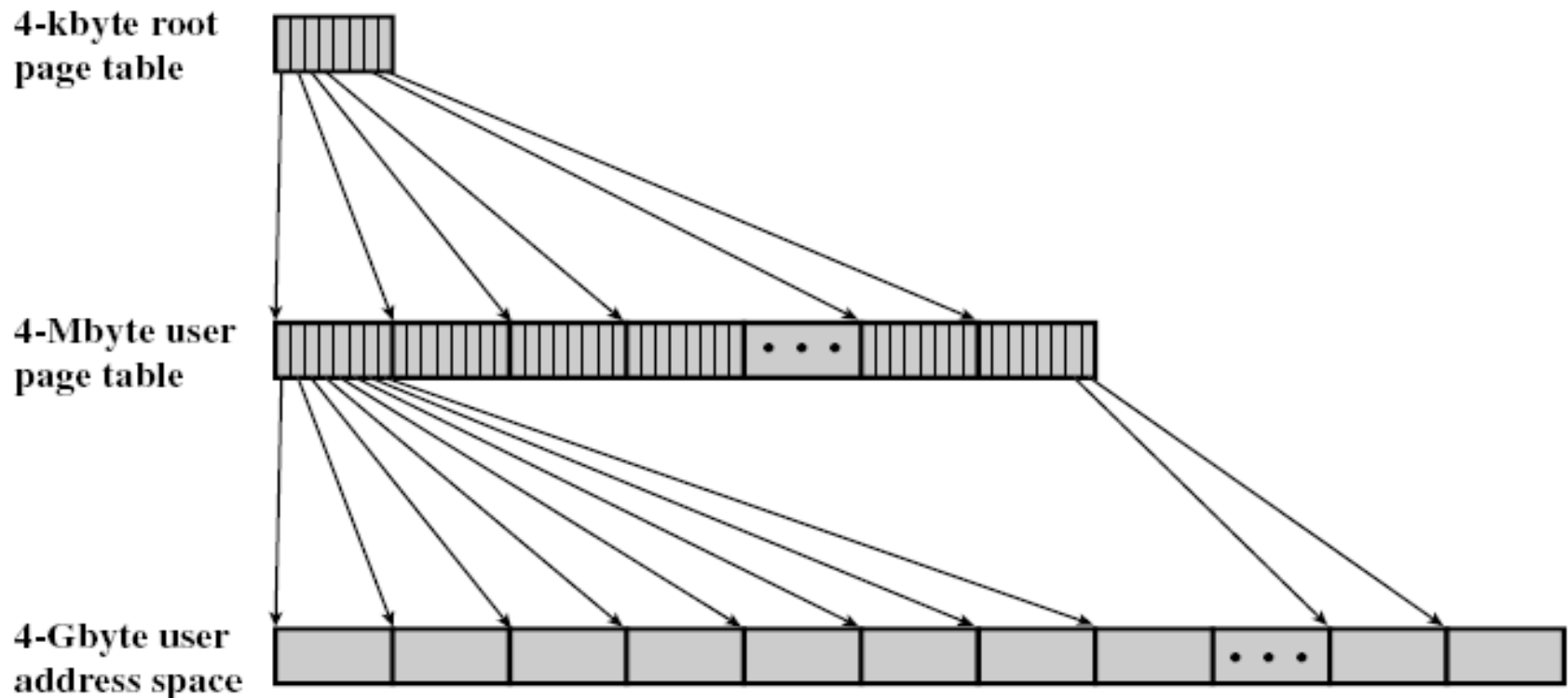
- Αν χρησιμοποιούνται 32 bits για εικονικό χώρο διευθύνσεων με μέγεθος σελίδων 4KB ($=2^{12}$), τότε ένας πίνακας σελίδων μπορεί να έχει 2^{20} εισόδους (θέσεις).
- Αν κάθε θέση του καταλαμβάνει 4 bytes τότε δεσμεύεται χώρος κύριας μνήμης :
 - $2^{20} \times 4 \text{ bytes} = 2^{22} \text{ bytes} = 4 \text{ MB}$ (ανά πίνακα διεργασίας).
- Αν στο σύστημα «τρέχουν» 25 διεργασίες τότε απαιτείται χώρος για τους πίνακες σελίδων : $25 \times 4 \text{ MB} = 100 \text{ MB}!!!!$
- Μέγεθος φυσικής μνήμης 4Gb είναι συνηθισμένο και πιθανό ;
- Ήδη χρησιμοποιείται διευθυνσιοδότηση 64-bits για την αύξηση του εικονικού χώρου διευθύνσεων:
 - Intel Itanium, AMD Clawhammer, DEC Alpha.



Πίνακες σελίδων 2 επιπέδων



Σχήμα 2 επιπέδων για 32-bit διεύθυνση



Επιτάχυνση της διαδικασίας μετάφρασης με hardware TLB (1/2)

- Κάθε αναφορά στην ιδεατή μνήμη προκαλεί **δύο προσβάσεις** στη φυσική μνήμη.
 - Μία για την προσκόμιση (fetch) του πίνακα σελίδων, ώστε τελικά να δημιουργηθεί η φυσική διεύθυνση.
 - Μία για την προσκόμιση των δεδομένων που υπάρχουν στη διεύθυνση.
- Ένα άμεσο σχήμα ιδεατής μνήμης θα οδηγούσε σε διπλασιασμό του χρόνου προσπέλασης της μνήμης.



Επιτάχυνση της διαδικασίας μετάφρασης με hardware TLB (2/2)

- Η βελτίωση της ταχύτητας πρόσβασης επιτυγχάνεται με τη χρήση ειδικής cache μνήμης υψηλής ταχύτητας που ονομάζεται **Translation Look-aside Buffer (TLB)**– ενδιάμεσος χώρος για τη μετάφραση σελίδων μνήμης που :
 - αποθηκεύει τις καταχωρήσεις του πίνακα σελίδων,
 - περιέχει τις καταχωρήσεις του πίνακα σελίδων που έχουν χρησιμοποιηθεί πρόσφατα και,
 - λειτουργεί με τρόπο παρόμοιο με αυτόν της μνήμης cache.

Το TLB αναφέρεται συχνά και με τον όρο associative registers .

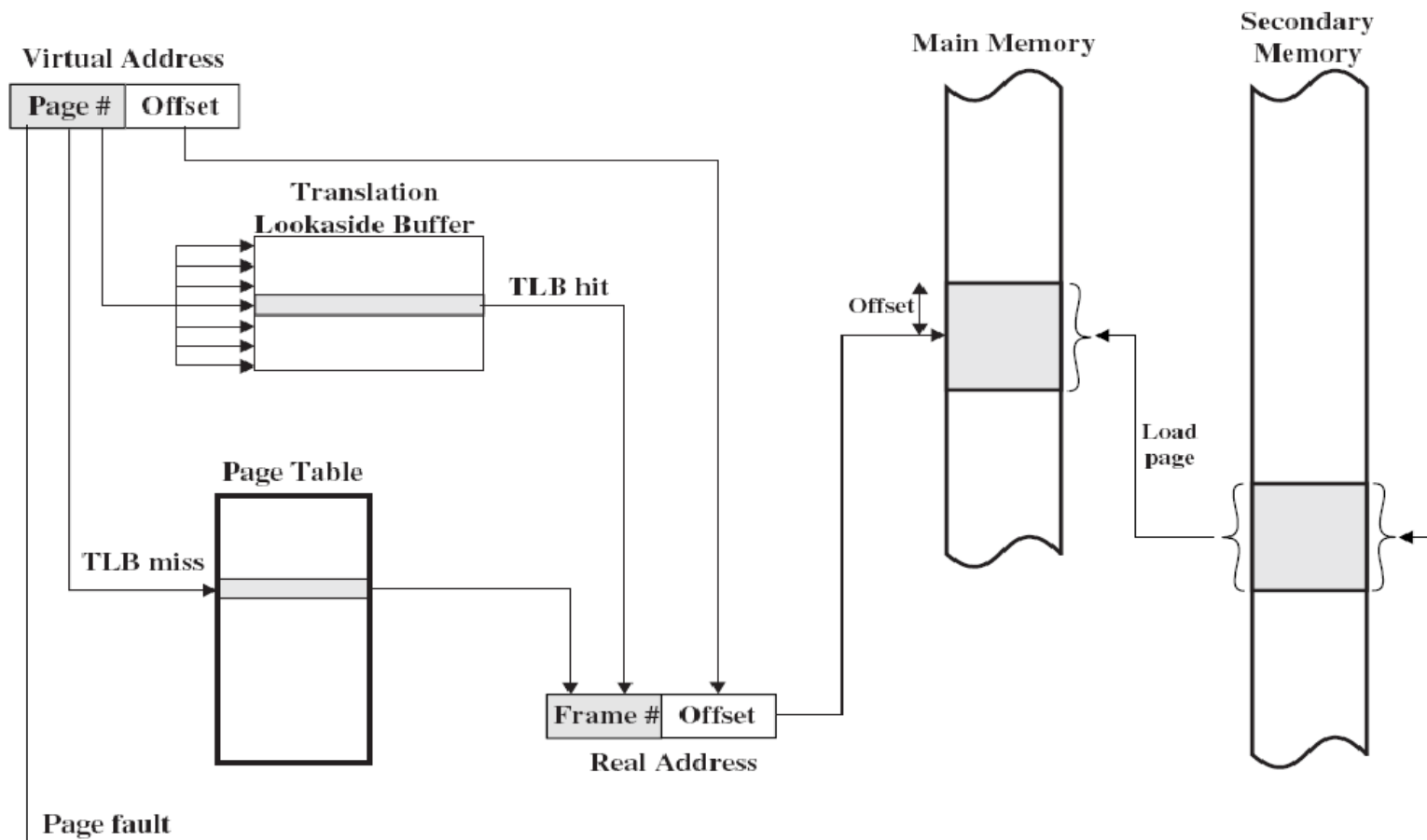


Η χρήση του TLB (1/2)

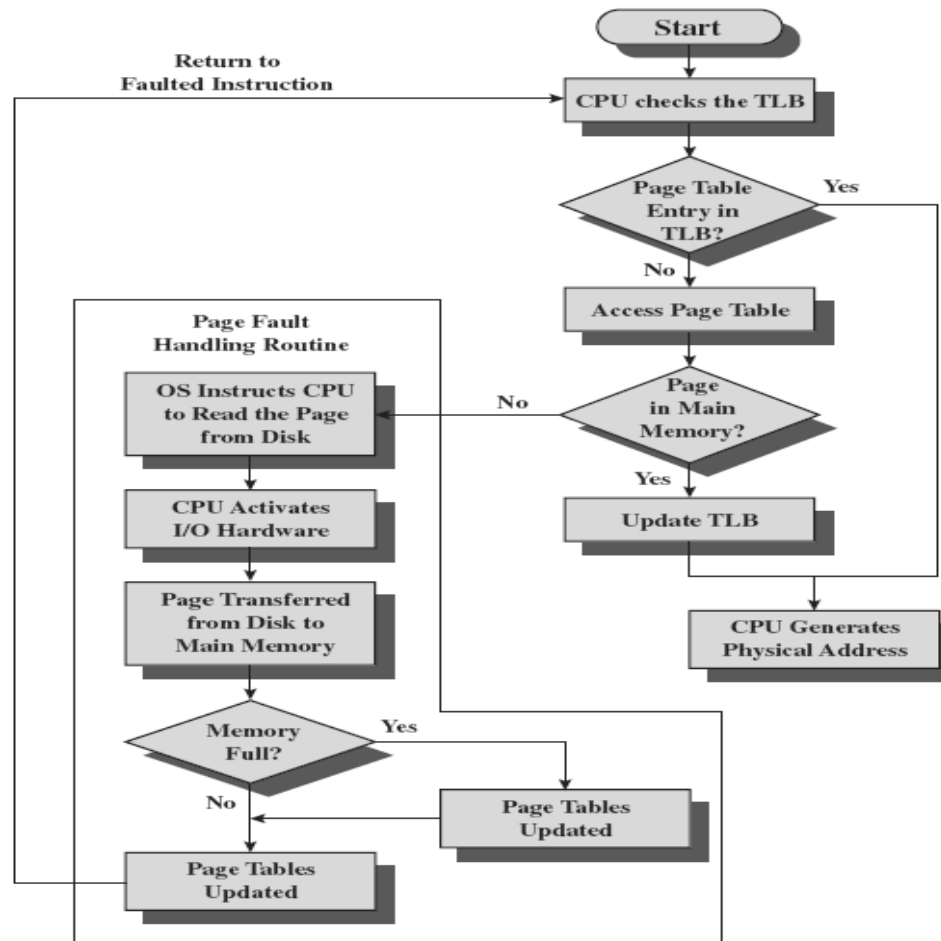
- Δεδομένης μιας εικονικής διεύθυνσης ο επεξεργαστής εξετάζει το TLB.
- Αν η καταχώρηση στον πίνακα σελίδων υπάρχει (**ευστοχία** - hit) γίνεται ανάκτηση του αριθμού πλαισίου και δημιουργείται η πραγματική διεύθυνση.
- Αν η καταχώρηση στον πίνακα σελίδων δε βρεθεί (**αστοχία** - miss), ο αριθμός σελίδας χρησιμοποιείται ως δείκτης στον πίνακα σελίδων της διεργασίας. Αν η ζητούμενη σελίδα υπάρχει εκεί ανακτάται και το TLB ενημερώνεται ώστε να συμπεριλάβει τη νέα είσοδο σελίδας. Αν η σελίδα δεν υπάρχει στον πίνακα σελίδων (άρα και στην κύρια μνήμη) απορρέει ένα σφάλμα σελίδας (page fault).
- Ως αναλογία επιτυχίας (hit ratio) ορίζεται το ποσοστό των επιτυχών αναζητήσεων μιας σελίδας στο TLB.



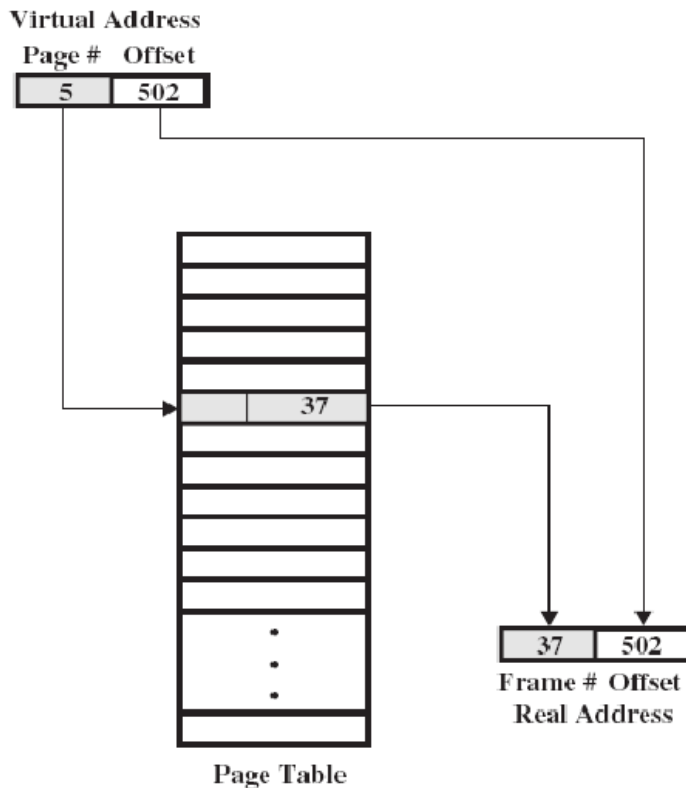
Η χρήση του TLB (2/2)



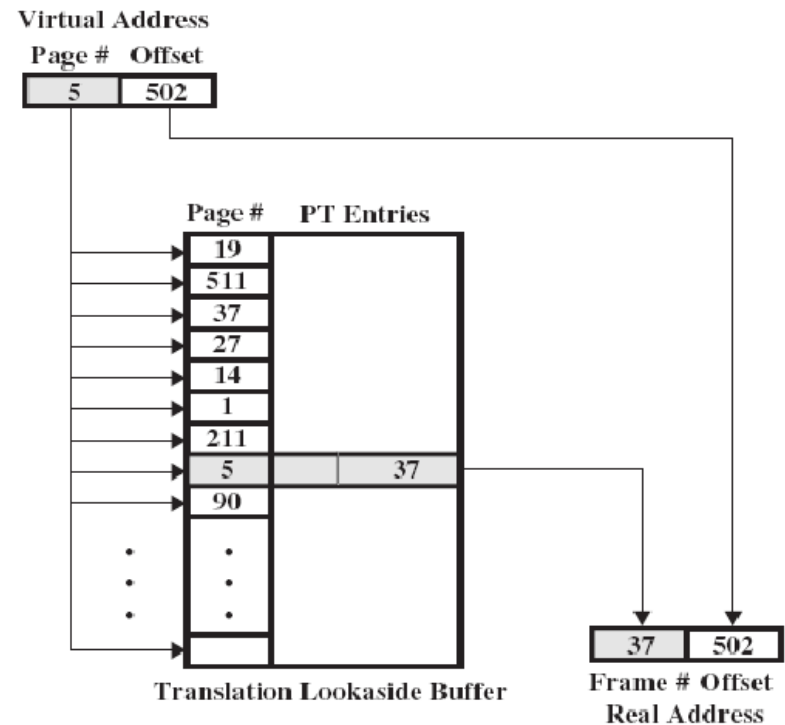
Διάγραμμα ροής



Σύγκριση άμεσης και συσχετιζόμενης αναζήτησης στον πίνακα σελίδων



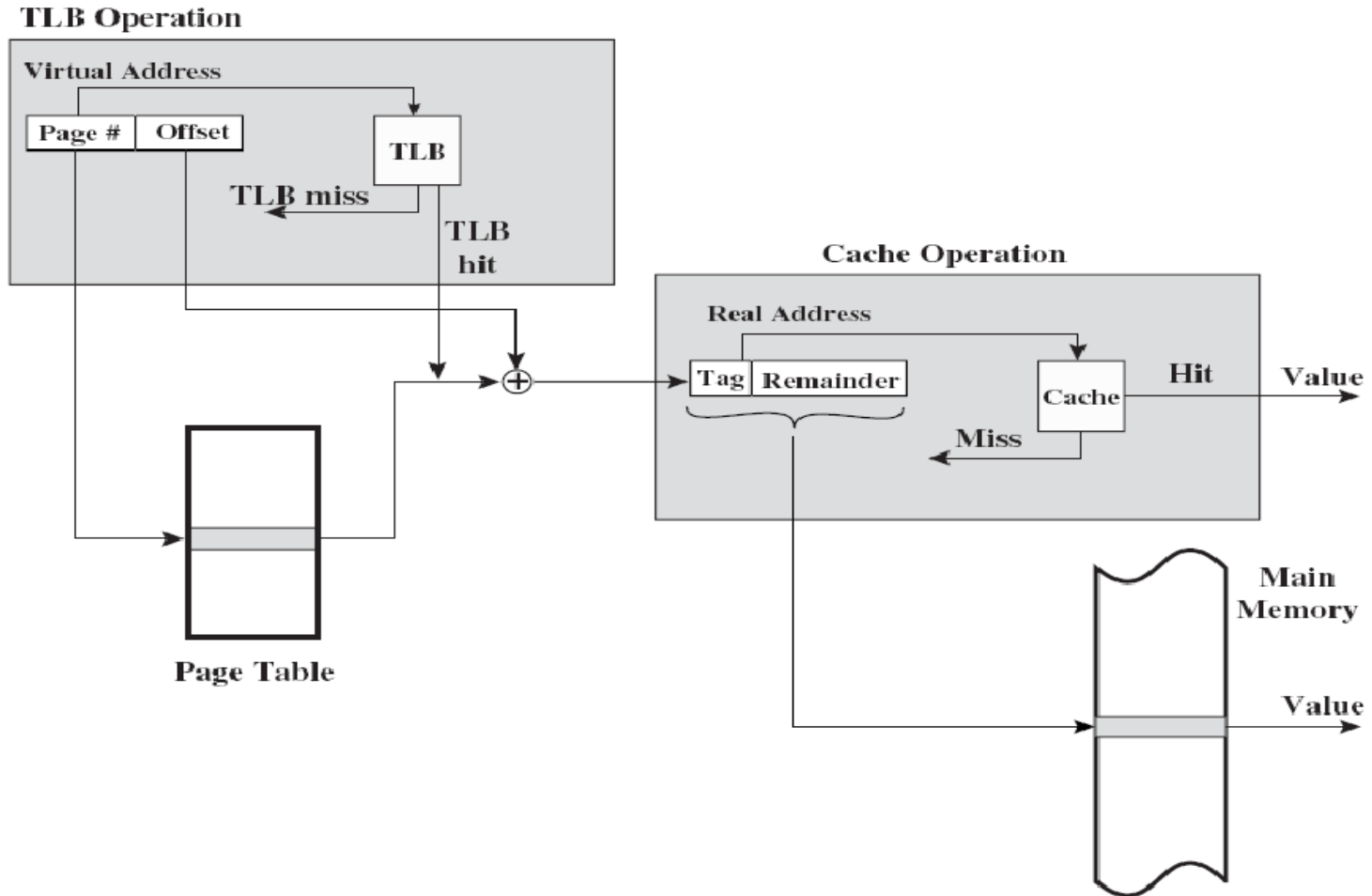
(a) Direct mapping



(b) Associative mapping



Λειτουργία του TLB



Ο πραγματικός χρόνος προσπέλασης

- Ο μέσος χρόνος προσπέλασης της κύριας μνήμης εξαρτάται από το πόσο συχνά ο ζητούμενος αριθμός σελίδας βρίσκεται στο TLB.
- Αυτό είναι γνωστό και ως hit ratio.
- Ο πραγματικός χρόνος προσπέλασης (EAT) υπολογίζεται λαμβάνοντας υπόψη την περίπτωση να βρεθεί ή να μη βρεθεί η ζητούμενη σελίδα σύμφωνα με την πιθανότητα που δίνει το hit ratio.

$$\text{EAT} = \text{hit ratio} * \text{entry found} + (1 - \text{hit ratio}) * \text{entry not found}$$



Παράδειγμα EAT

- Βασικές παραδοχές:

memory access time: 60 ns; TLB access time: 6 ns.

- EAT χωρίς σελιδοποίηση: 60 ns.
- EAT με χρήση του πίνακα σελίδων: 60 ns + 60 ns = 120 ns.
- EAT TLB/πίνακας σελίδων.

hit ratio 90%

$$\text{EAT} = 0.9 * (6 + 60) + 0.1 * (6 + 60 + 60) = 59.4 + 12.6 = 72$$

20% απώλεια απόδοσης σε σχέση με τη μη σελιδοποίηση

hit ratio 99%

$$\text{EAT} = 0.99 * (6 + 60) + 0.01 * (6 + 60 + 60) = 65.34 + 1.26 = 66.60$$

11% απώλεια απόδοσης σε σχέση με τη μη σελιδοποίηση

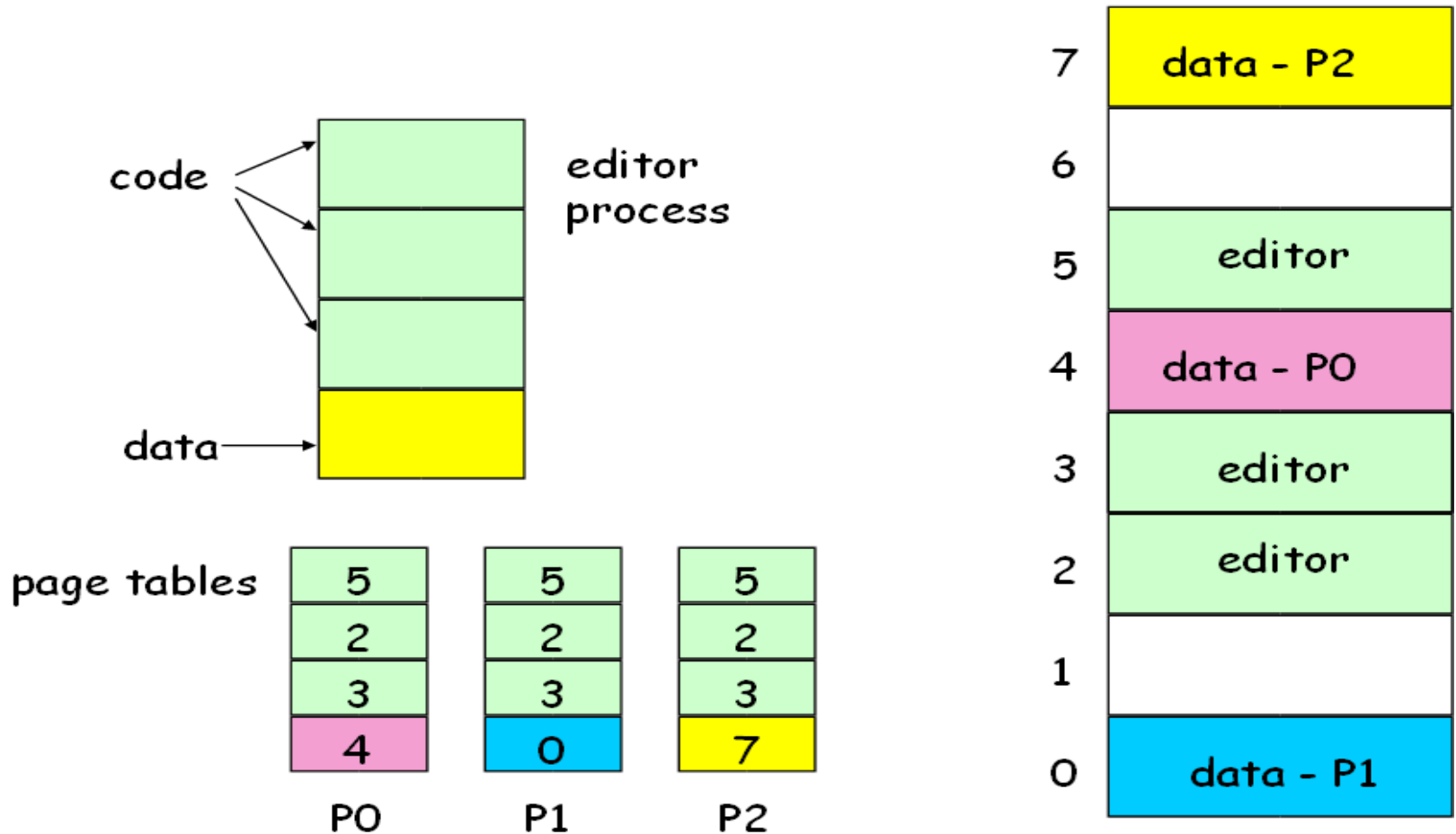


Πίνακας σελίδων σε πραγματικά ΛΣ

- Τα σύγχρονα λειτουργικά συστήματα έχουν σελίδες μεγέθους 4K (=4096 bytes)-(Sun SPARC 4Kb pages, Intel x86: 4Kb pages).
- Για τη διευθυνσιοδότηση χρησιμοποιούνται 32 bits, δηλαδή υποστηρίζεται μέγιστο μέγεθος κεντρικής μνήμης 4GB.
- Το μέγεθος της εικονικής μνήμης είναι πολύ μεγαλύτερο από τη διαθέσιμη φυσική μνήμη του συστήματος.
- Ο πίνακας σελίδων έχει 2^{20} θέσεις (εισόδους) και κάθε είσοδος έχει μέγεθος 4 Bytes. Άρα δεσμεύει 4MB κεντρικής μνήμης.



Διαμοιραζόμενες σελίδες



Ο κώδικας της διεργασίας ανήκει σε δύο τύπους

- **Διαμοιραζόμενος κώδικας:**
 - Μόνο για ανάγνωση επανεισαγόμενος (reentrant) διαμοιραζόμενος κώδικας μεταξύ των διεργασιών.
 - Ο διαμοιραζόμενος κώδικας εμφανίζεται στην ίδια θέση στο φυσικό χώρο διευθύνσεων.
- **Ιδιωτικός κώδικας και δεδομένων:**
 - Κάθε διεργασία διατηρεί ένα ξεχωριστό αντίγραφο του κώδικα και των δεδομένων.
 - Οι ιδιωτικές σελίδες μπορούν να εμφανίζονται οπουδήποτε στο φυσικό χώρο διευθύνσεων.



Κατάτμηση (segmentation) (1/2)

- Ένα τμήμα (segment) είναι ένα μεταβλητού μεγέθους σύνολο συνεχόμενων διευθύνσεων μνήμης στον ιδεατό χώρο διευθύνσεων μιας διεργασίας που οργανώνεται και διαχειρίζεται από το Λ.Σ. ως μια ενιαία μονάδα.
- Κατάτμηση είναι ο τρόπος οργάνωσης της ιδεατής μνήμης σε τμήματα.
- Μια διεύθυνση αποτελείται από δύο μέρη—έναν αριθμό τμήματος και μια μετατόπιση (offset).



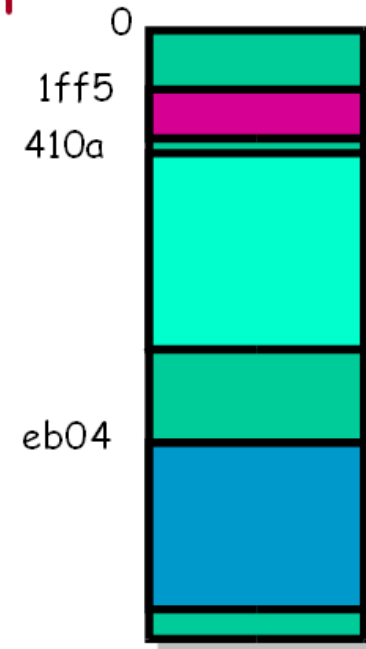
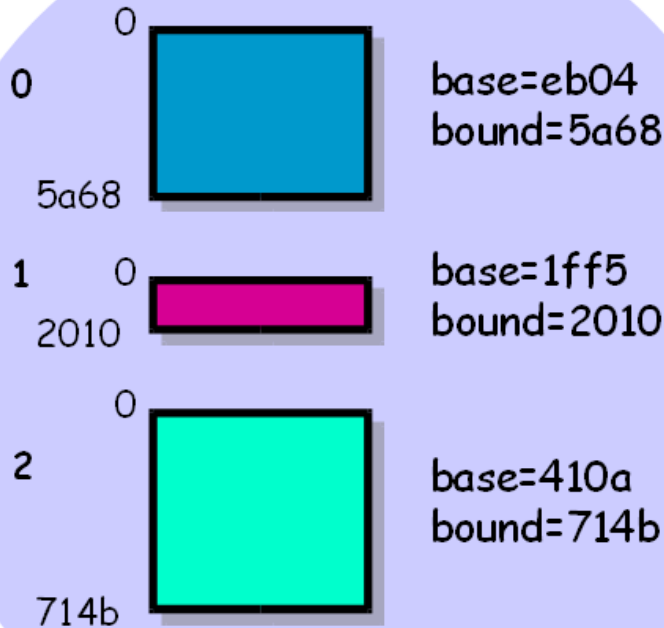
Τα τμήματα έχουν δυναμικό μέγεθος

- Τα τμήματα δεν είναι ίσα και η κατάτμηση είναι παρόμοια με τη δυναμική τμηματοποίηση. Μειώνεται ο εσωτερικός κατακερματισμός.
- Τα τμήματα μπορούν να έχουν δυναμικό μέγεθος ώστε να απλοποιείται η διαχείριση δυναμικών δομών δεδομένων.
- Η κατάτμηση:
 - επιτρέπει στα προγράμματα να τροποποιούνται και να μεταφράζονται εκ νέου ανεξάρτητα.
 - είναι κατάλληλη για διαμοίραση και προστασία δεδομένων.



Κατάτμηση (Segmentation) (2/2)

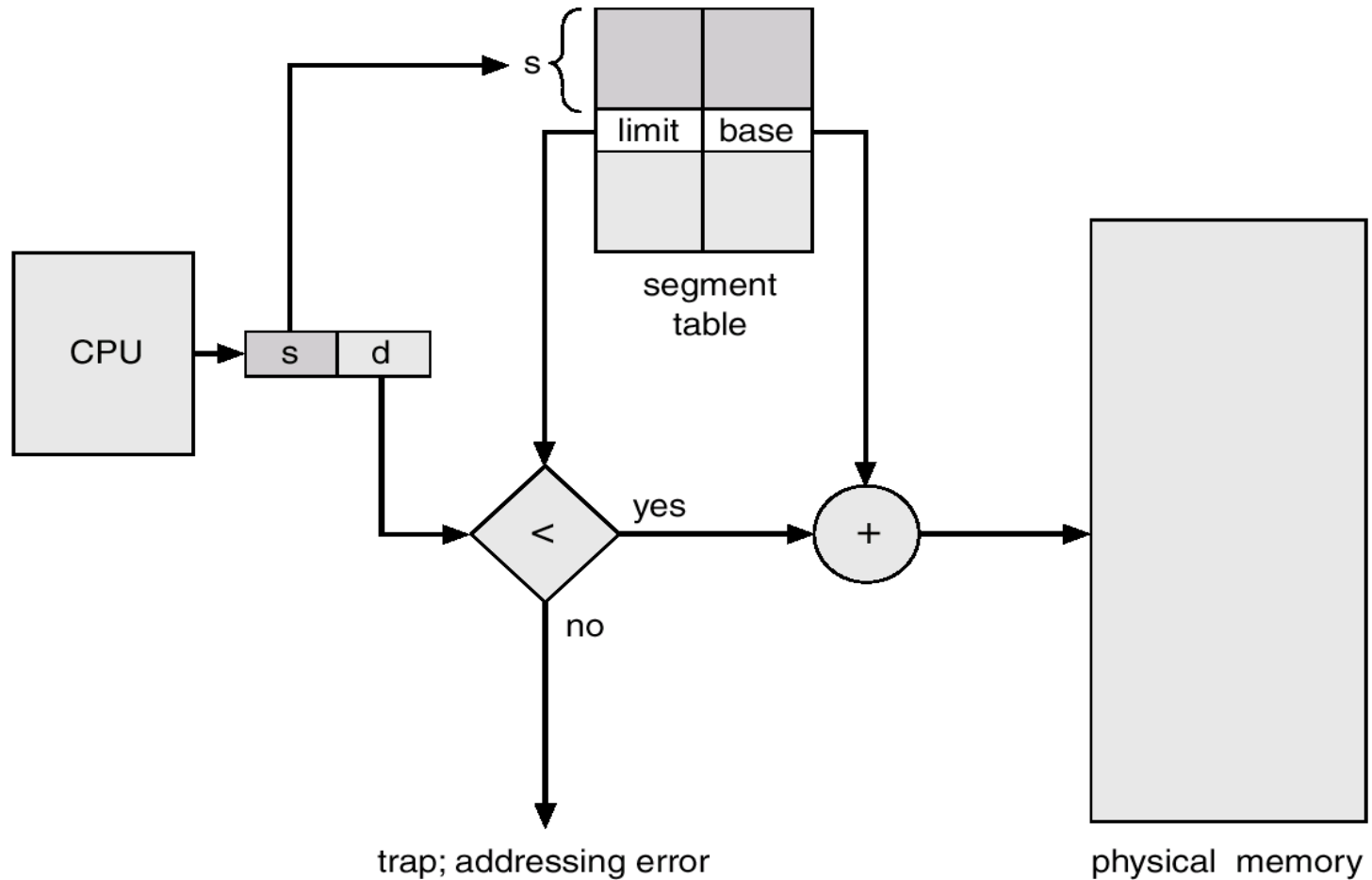
Μια εικονική διεύθυνση είναι ένας αριθμός τμήματος (segment number) και μια μετατόπιση offset.



Κάθε τμήμα τοποθετείται σε μια συνεχόμενη περιοχή της μνήμης.



Μετατροπή διεύθυνσης σε σύστημα με κατάτμηση (1/2)



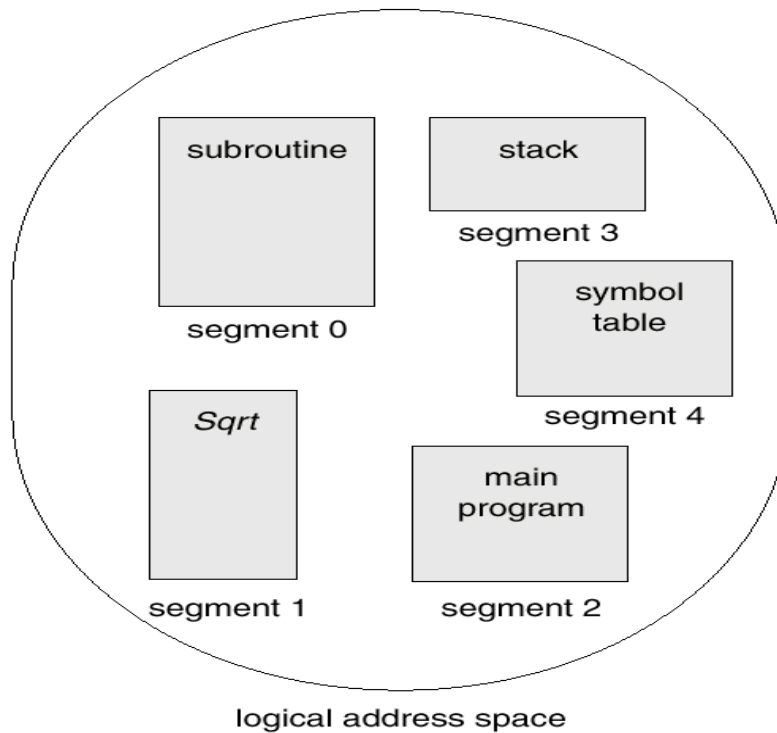
Κατάτμηση και σελιδοποίηση ως προς τον προγραμματιστή

- Η κατάτμηση είναι φανερή στον προγραμματιστή σε αντίθεση με τη σελιδοποίηση και παρέχεται ως διευκόλυνση για την οργάνωση προγραμμάτων και δεδομένων (αρκεί ο προγραμματιστής να γνωρίζει τα όρια τμημάτων που αναγνωρίζει η γλώσσα προγραμματισμού, καθώς και αν όλα μπορούν να έχουν δυναμικό μέγεθος!!!).
- Ο προγραμματιστής βλέπει το πρόγραμμα σαν συλλογή από segments π.χ. main program, function, objects, global variables, stack...

Δεν υπάρχει μια απλή συσχέτιση μεταξύ των λογικών και των φυσικών διευθύνσεων.

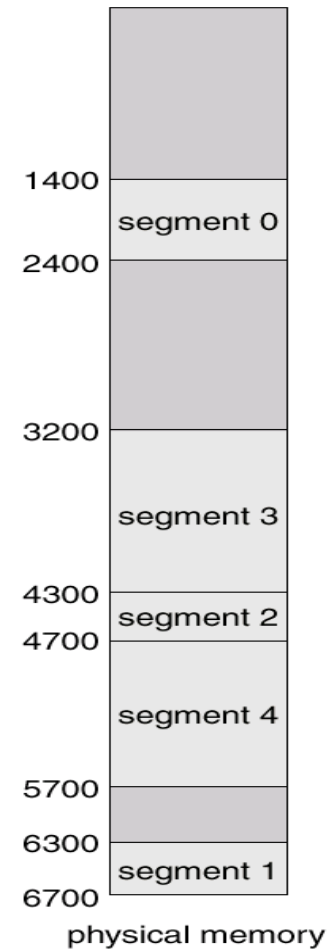


Παράδειγμα συστήματος με τμήματα



	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

segment table

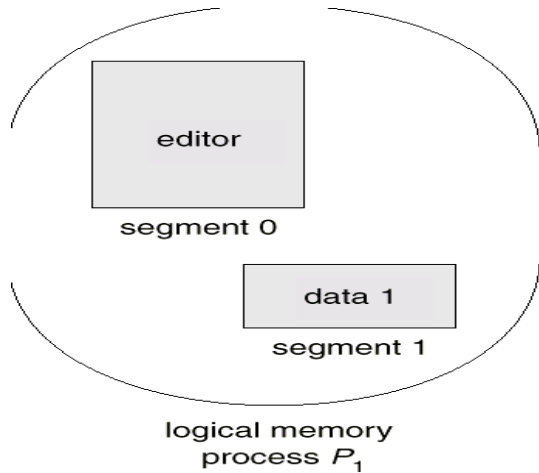


Πλεονεκτήματα της κατάτμησης

- Η εικόνα της μνήμης είναι η εικόνα που έχει ο προγραμματιστής (ή ο έμπειρος χρήστης).
- Τα τμήματα προστατεύονται μεταξύ τους.
- Κάθε τμήμα περιέχει ένα τύπο πληροφορίας (πχ, instructions, stack, ...).
- Η διαμοίραση τμημάτων είναι λογική και εύκολη.
- Αν όλες οι εντολές είναι σε ένα τμήμα και όλα τα δεδομένα σε άλλο, το τμήμα εντολών μπορεί να διαμοιραστεί ελεύθερα σε διαφορετικές διεργασίες (κάθε μια με τα δικά της δεδομένα).

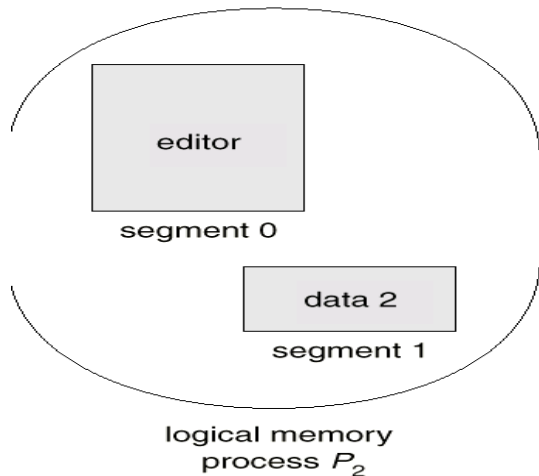


Διαμοίραση τμημάτων



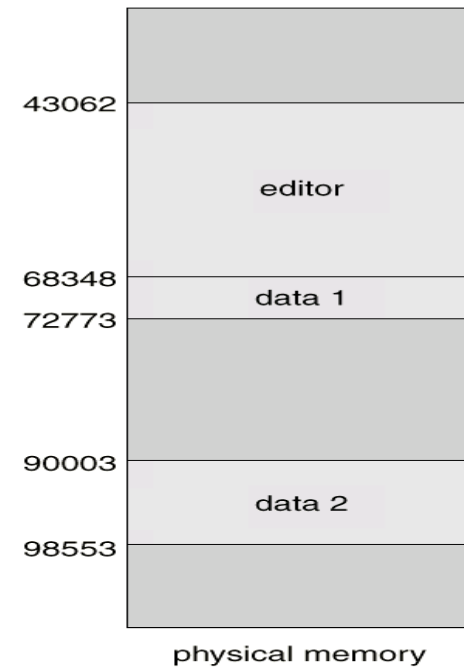
	limit	base
0	25286	43062
1	4425	68348

segment table
process P_1

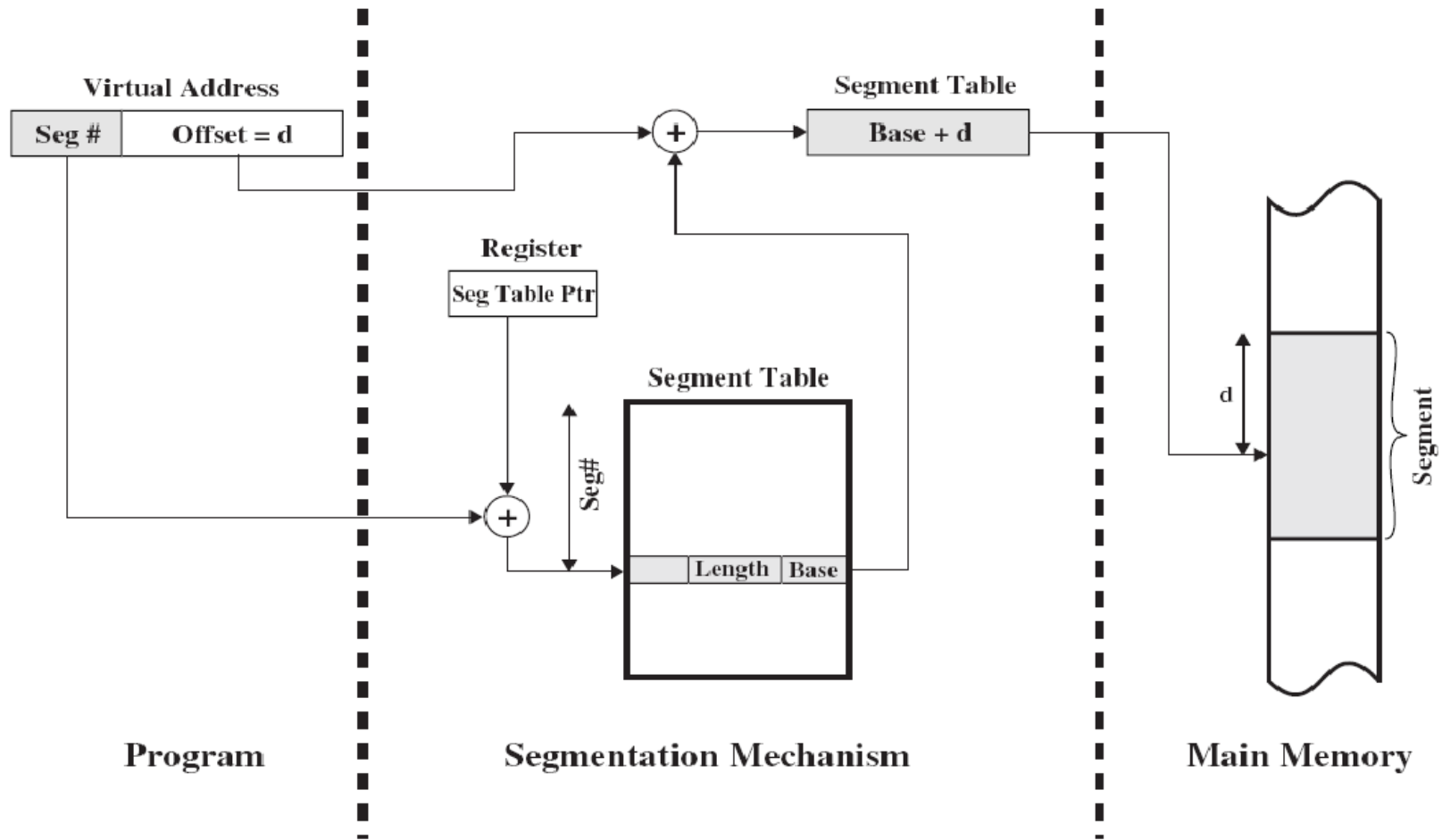


	limit	base
0	25286	43062
1	8850	90003

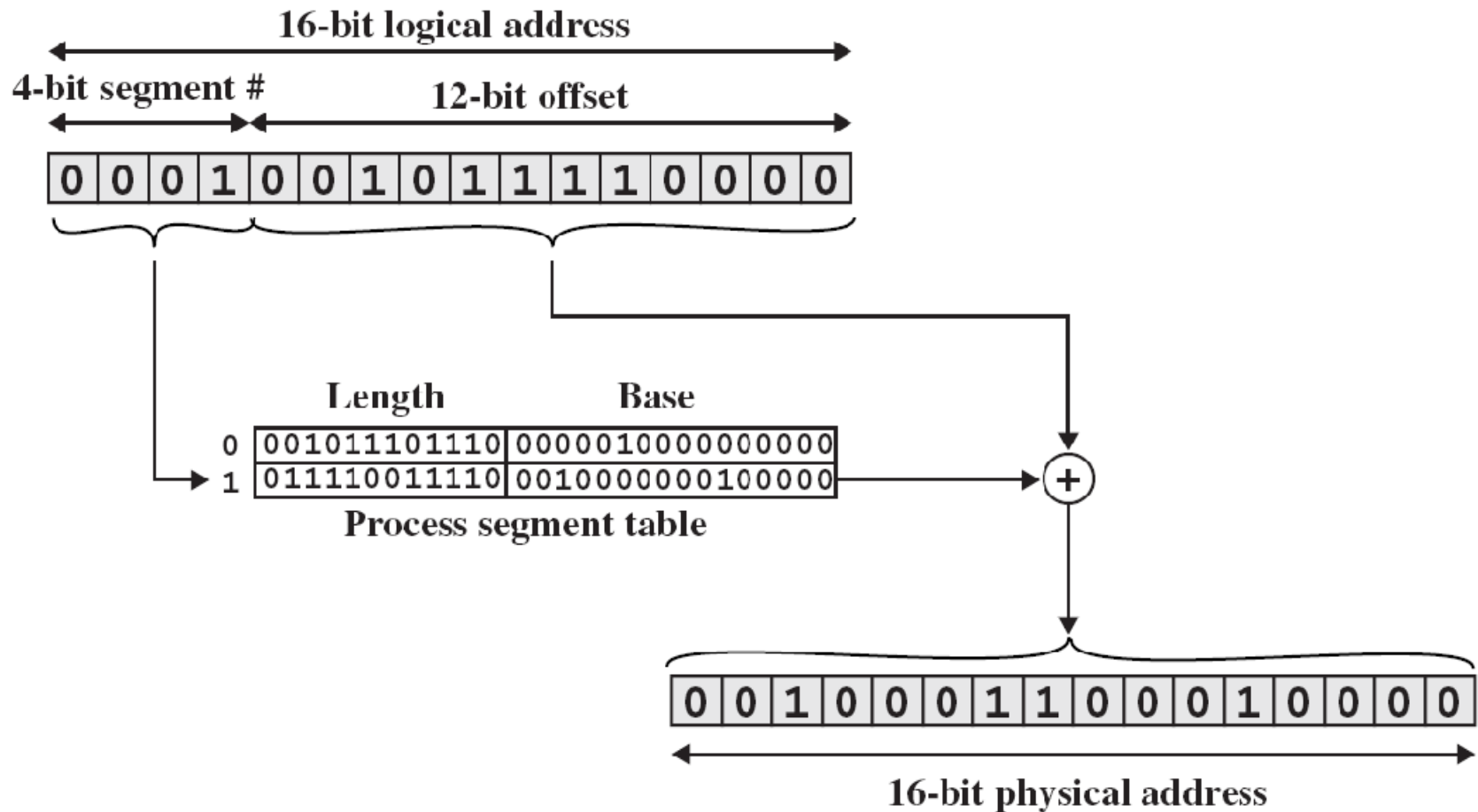
segment table
process P_2



Μετατροπή διεύθυνσης σε σύστημα με κατάτμηση (2/2)



Κατάτμηση: μετατροπή λογικών σε φυσικές διευθύνσεις (1/2)



Πίνακες Τμημάτων (Segment Tables)

- Αντιστοιχίζουν τα segments στην κύρια μνήμη.
- Κάθε είσοδος περιέχει τον αριθμό και το μήκος του segment.
- Απαιτούνται :
 - ένα bit για να αποφασιστεί αν το segment είναι ήδη στην κύρια μνήμη,
 - ένα επιπλέον bit για να αποφασιστεί αν το segment έχει μεταβληθεί από τότε που φορτώθηκε στην κύρια μνήμη.
- Ο πίνακας τμημάτων έχει μεταβλητό μέγεθος, βρίσκεται στην κύρια μνήμη και ένας καταχωρητής κρατά την αρχική διεύθυνση του πίνακα τμημάτων για τη διεργασία που εκτελείται.



Είσοδοι Πίνακα Τμημάτων

Virtual Address



Segment Table Entry



Η αρχιτεκτονική της κατάτμησης

- Ο compiler δημιουργεί segments για: **main(), functions, globals, ...**
- Κάθε τμήμα αποθηκεύεται ξεχωριστά στη μνήμη.
- Πρέπει να υπάρχει υποστήριξη υλικού για την αντιστοίχιση λογικών διευθύνσεων (segment number + offset) σε φυσικές διευθύνσεις.
- Καταχωρητές που χρησιμοποιούνται:
 - **segment table base register (STBR)**: δείχνει στη διεύθυνση μνήμης του πίνακα τμημάτων στη μνήμη.
 - **segment table length (limit) register (STLR)**: δείχνει τον αριθμό των τμημάτων που χρησιμοποιεί το πρόγραμμα.



Σύγκριση απλής κατάτμησης και σελιδοποίησης

- Η κατάτμηση απαιτεί περισσότερο σύνθετο hardware για τον μετασχηματισμό των διευθύνσεων.
- Η κατάτμηση έχει το μειονέκτημα του εξωτερικού κατακερματισμού.
- Η σελιδοποίηση δημιουργεί πολύ μικρό εσωτερικό κατακερματισμό.
- Η κατάτμηση είναι ορατή από τον προγραμματιστή ενώ η σελιδοποίηση είναι αδιαφανής .
- Η κατάτμηση θεωρείται ως πλεονέκτημα που προσφέρεται στον προγραμματιστή για να οργανώσει λογικά ένα πρόγραμμα σε segments και να χρησιμοποιήσει διαφορετικά είδη προστασίας (π.χ. execute-only, read-write) .
- Για το σκοπό αυτό στους πίνακες τμημάτων πρέπει να χρησιμοποιούνται bits προστασίας.



Σύγκριση σελιδοποίησης και κατάτμησης

	σελιδοποίηση	κατάτμηση
Είναι απαραίτητο να γνωρίζει ο προγραμματιστής ότι χρησιμοποιείται αυτή η τεχνική; Πόσοι χώροι γραμμικών διευθύνσεων υπάρχουν;	OXI 1	ΝΑΙ ΠΟΛΛΟΙ
Ο συνολικός χώρος διευθύνσεων υπερβαίνει το μέγεθος της φυσικής μνήμης;	ΝΑΙ	ΝΑΙ
Μπορούν οι διαδικασίες και τα δεδομένα να διαχωριστούν και να προστατευθούν ξεχωριστά; Μπορούν πίνακες με αυξομειούμενο μέγεθος να εξυπηρετηθούν εύκολα;	OXI OXI	ΝΑΙ ΝΑΙ
Διευκολύνεται η διαμοίραση των διαδικασιών μεταξύ των χρηστών;	OXI	ΝΑΙ
Ποιος είναι ο σκοπός αυτής της τεχνικής;	Η απόκτηση ενός μεγάλου γραμμικού χώρου διευθύνσεων χωρίς να είναι αναγκαία η αγορά επιπλέον φυσικής μνήμης	Να δοθεί η δυνατότητα σε προγράμματα και δεδομένα να διασπαστούν σε ανεξάρτητες λογικές ενότητες που διαμοιράζονται και προστατεύονται εύκολα.



Κατάτμηση με σελιδοποίηση

- Οι σύγχρονοι υπολογιστές χρησιμοποιούν συνδυασμό κατάτμησης και σελιδοποίησης.
- Ο χώρος διευθύνσεων του χρήστη χωρίζεται σε ένα πλήθος τμημάτων και κάθε τμήμα διασπάται σε ένα πλήθος σελίδων σταθερού μεγέθους που είναι το ίδιο με το μέγεθος πλαισίου της κύριας μνήμης.
- Κάθε διεργασία συνδέεται με έναν πίνακα τμήματος και έναν αριθμό από πίνακες σελίδων, έναν για κάθε τμήμα της διεργασίας.
- Επιλύεται το πρόβλημα του εξωτερικού κατακερματισμού μέσω της σελιδοποίησης των τμημάτων.

Κάθε είσοδος στον πίνακα τμημάτων δεν περιέχει τη διεύθυνση βάσης του τμήματος, αλλά τη διεύθυνση βάσης του πίνακα σελίδων που αντιστοιχεί στο τμήμα.



Είσοδοι πινάκων τμημάτων και σελίδων

Virtual Address



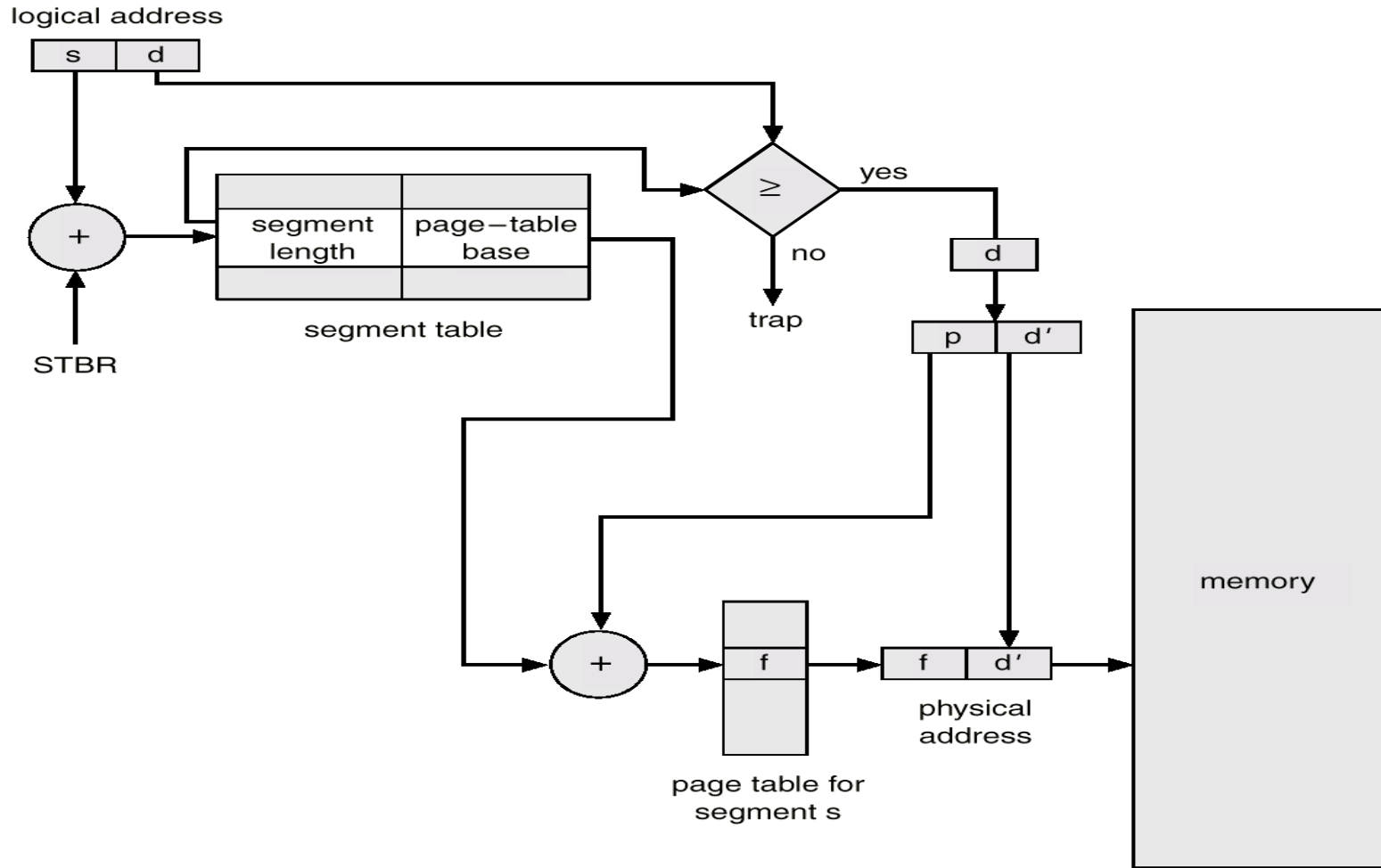
Segment Table Entry



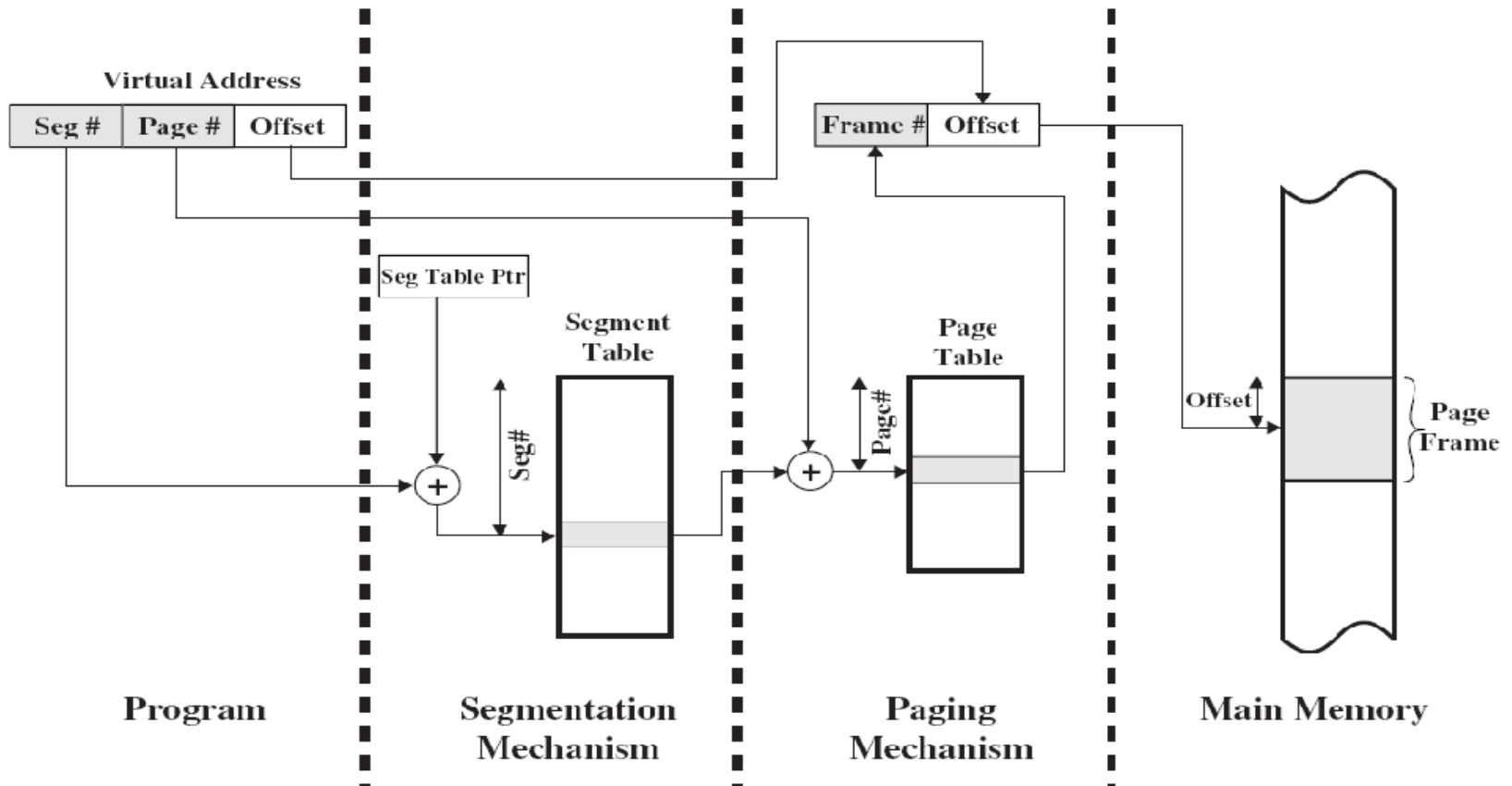
Page Table Entry



Μετατροπή διεύθυνσης (1/2)



Μετατροπή διεύθυνσης (2/2)



Ασκήσεις



Μετάφραση σε φυσική διεύθυνση (1/2)

- Αριθμός τμήματος 0001.
- Ο αριθμός αυτός χρησιμοποιείται στον segment table της διεργασίας για να βρεθεί η φυσική διεύθυνση της αρχής του τμήματος.
- Συγκρίνεται η μετατόπιση με το μήκος του τμήματος. Αν ΜΕΤΑΤΟΠΙΣΗ > ΜΗΚΟΣ ΤΜΗΜΑΤΟΣ προκύπτει ΜΗ ΕΓΚΥΡΗ ΔΙΕΥΘΥΝΣΗ.
- **ΦΥΣΙΚΗ ΔΙΕΥΘΥΝΣΗ=ΑΘΡΟΙΣΜΑ: ΦΥΣΙΚΗΣ ΔΙΕΥΘΥΝΣΗΣ ΑΡΧΗΣ ΤΜΗΜΑΤΟΣ + ΜΕΤΑΤΟΠΙΣΗ.**



Μετάφραση σε φυσική διεύθυνση (2/2)

- Από τον process segment table προκύπτει ότι το **segment 0** έχει μήκος : $512 + 128 + 64 + 32 + 8 + 4 + 2 = 750$

2048	1024	512	256	128	64	32	16	8	4	2	1
0	0	1	0	1	1	1	0	1	1	1	0

• Άρα η λογική διεύθυνση πράγματι βρίσκεται στο segment 1 (επειδή $752 > 750$)

• Για να βρεθεί η φυσική διεύθυνση θα προσθέσω τη μετατόπιση στη βάση του segment 1 :

base	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
offset	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0
address	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0



Ανάλυση του παραδείγματος

- Λογική διεύθυνση: **0001001011110000**.
- Αριθμός τμήματος: 4 bits και είναι ο αριθμός **0001**.
- Μετατόπιση τμήματος 12 bits.
- Μέγιστο μέγεθος τμήματος $2^{12} = 4096$ bytes.
- Η μετατόπιση της λογικής διεύθυνσης αντιστοιχεί στον αριθμό: $512+128+64+32+16 = (752)_{10}$.

2048	1024	512	256	128	64	32	16	8	4	2	1
0	0	1	0	1	1	1	1	0	0	0	0



1^η Άσκηση

- Ποιο είναι το μεγαλύτερο μέγεθος προγράμματος που χωρά σε ένα πίνακα σελίδων σε ένα σύστημα που χρησιμοποιεί φυσικές διευθύνσεις 32-bit και μέγεθος σελίδας 1K;

ΛΥΣΗ

ΦΥΣ. ΔΙΕΥΘΥΝΣΗ 32 bits \rightarrow χώρος διευθύνσεων 2^{32} διαφορετικές διευθύνσεις RAM.

Μέγεθος σελίδας 1K = 2^{10} διαφορετικά offset μέσα στη σελίδα.

ΑΡΑ : αφαιρούμε από τα 32 bit τα bit που είναι για το offset και βρίσκουμε ότι μέγιστο πλήθος σελίδων = $2^{32}/2^{10}=2^{22}$.

Κάθε διεύθυνση των 32 bits (=4 bytes) πρέπει να χωρά σε κάθε μια από τις 2^{22} θέσεις του πίνακα σελίδων άρα μέγεθος του πίνακα σελίδων: $2^{22} \times 4\text{bytes} = 16\text{Mb}$.

Μεγαλύτερο μέγεθος προγράμματος: **16Mb**.



2^η Άσκηση (1/3)

- Δίνεται ο πίνακας σελίδων μιας διεργασίας:
- Ποιο είναι το μικρότερο δυνατό μέγεθος σελίδας (ώστε να μπορούν να διευθυνσιοδοτηθούν οι διευθύνσεις 1234, 3333);
- Δώστε σε δυαδική μορφή τη φυσική διεύθυνση της εικονικής διεύθυνσης 1234.
- Δώστε σε δεκαδική μορφή τη φυσική διεύθυνση της εικονικής διεύθυνσης 3333.

Page No

Frame No

0

5

1

3

2

4

3

0

4

1

5

2

5 σελίδες



2^η Λύση Άσκησης (2/3)

Γιατί το μέγεθος σελίδας είναι 1 K;

- Έστω μέγεθος σελίδας 512 bytes ($=2^9$ bytes).
- Η εικονική διεύθυνση 3333 απαιτεί **6 γεμάτες σελίδες** ($=512 \times 6=3072$) και στην 7η σελίδα μετατόπιση (offset) $=3333-3072=261$.
- Άρα απαιτούνται αριθμοί σελίδων **0,1,2,3,4,5,6,7**.



2^η Λύση Άσκησης (3/3)

- Ο πίνακας όμως, δεν έχει αριθμό σελίδας μεγαλύτερο του 5. Στην περίπτωση που θεωρηθεί ως μέγεθος σελίδας 512 bytes, αυτομάτως η διεύθυνση 3333 είναι εκτός μνήμης. **ΑΔΥΝΑΤΟ ΔΙΟΤΙ** μας δίνεται ότι ο πίνακας σελίδων της διεργασίας υφίσταται.

ΕΠΟΜΕΝΩΣ Ο ΕΠΟΜΕΝΟΣ ΑΡΙΘΜΟΣ ΠΟΥ ΥΠΑΡΧΕΙ ΩΣ ΔΥΝΑΜΗ ΤΟΥ 2 ΕΙΝΑΙ $2^{10} = 1024 \text{ bytes} = 1\text{KB}$.



3^η Άσκηση

Θεωρείστε ένα λογικό χώρο διευθύνσεων που αποτελείται από 1024 σελίδες με μέγεθος κάθε σελίδας 1024 ($=2^{10}$) bytes, που αντιστοιχείται σε μια φυσική μνήμη που αποτελείται από 32 frames.

Πόσα bits υπάρχουν στη λογική διεύθυνση;

Πόσα bits υπάρχουν στη φυσική διεύθυνση;



4^η Άσκηση

- Σε ένα σύστημα που χρησιμοποιεί σελιδοποίηση μια διεργασία δεν μπορεί να έχει πρόσβαση σε μνήμη που δεν της ανήκει. Γιατί;

Μια διεύθυνση σε ένα σύστημα με σελιδοποίηση αποτελείται από έναν αριθμό σελίδας και μια μετατόπιση. Η φυσική σελίδα (πλαίσιο) υπολογίζεται μέσω της αναζήτησης στον πίνακα σελίδων της διεργασίας. Επειδή το Λ.Σ. ελέγχει τα περιεχόμενα αυτού του πίνακα μπορεί να περιορίσει μια διεργασία ώστε να έχει πρόσβαση μόνον σε εκείνες τις φυσικές σελίδες (πλαίσια) που της έχουν εκχωρηθεί. Δεν υπάρχει τρόπος για μια διεργασία να αναφερθεί σε σελίδα που δεν της ανήκει επειδή ο αριθμός αυτός δεν βρίσκεται στον πίνακα σελίδων της.



Διαδιεργασιακή επικοινωνία

- Για να επιτραπεί μια τέτοια πρόσβαση το Λ.Σ. απλά χρειάζεται να επιτρέψει σε τμήματα της μνήμης που δεν ανήκουν σε μια διεργασία να προστεθούν στον πίνακα σελίδων της διεργασίας. Αυτό είναι χρήσιμο όταν δύο ή περισσότερες διεργασίες χρειάζεται να ανταλλάσσουν δεδομένα που μόλις διάβασαν και να τα γράψουν στην ίδια φυσική διεύθυνση (που είναι δυνατόν να αντιστοιχεί σε διαφορετικές λογικές διευθύνσεις).
- Η διαδικασία αυτή καθιστά πολύ αποτελεσματική τη **διαδιεργασιακή (Interprocess)** επικοινωνία.



5^η Άσκηση (1/2)

- Θεωρείστε ένα σύστημα που χρησιμοποιεί:
 - απλή σελιδοποίηση και
 - τεχνική TLB.
- Αν μια αναφορά στη μνήμη απαιτεί 400 ns, μια αναφορά στο TLB απαιτεί 50 ns και το ποσοστό επιτυχίας (hit - rate) στο TLB είναι 80% ποιος είναι ο πραγματικός χρόνος αναφοράς στη μνήμη; Πόση είναι η βελτίωση στην ταχύτητα (speed-up) λόγω χρήσης της τεχνικής TLB;



5^η Άσκηση (2/2)

- Απλή σελιδοποίηση: Κάθε αναφορά στη μνήμη απαιτεί 2 προσπελάσεις άρα συνολικός χρόνος: $400 \text{ ns} + 400 \text{ ns} = 800 \text{ ns}$.
- Απλή σελιδοποίηση και TLB:
 - Επιτυχία (hit) στο TLB: $50 \text{ ns} + 400 \text{ ns} = 450 \text{ ns}$.
 - Αποτυχία (miss) στο TLB: $50 \text{ ns} + 400 \text{ ns} + 400 \text{ ns} = 850 \text{ ns}$.
 - Πραγματικός χρόνος προσπέλασης με απλή σελιδοποίηση και TLB:
 - $450 * 80\% + 850 * 20\% = 530 \text{ ns}$.
 - $\text{Speed-up} = 800 / 530 = 1.51$.



6^η Άσκηση (1/2)

- Ένα σύστημα χρησιμοποιεί διευθύνσεις των 32 bits και έχει κεντρική μνήμη 4 MB. Το μέγεθος σελίδας είναι 1 K.

Ποιο είναι το μέγεθος του πίνακα σελίδων;



6^η Άσκηση (2/2)

- Ο πίνακας σελίδων περιέχει το πεδίο του αριθμού σελίδας της εικονικής διεύθυνσης. Έτσι το πλήθος των γραμμών στον πίνακα σελίδων είναι ίσο με τον αριθμό των εικονικών σελίδων:

Εικονικές σελίδες = $2^{32}/2^{10}=2^{22}= 4\text{M}$ σελίδες.

- Το πλάτος κάθε γραμμής στον πίνακα σελίδων είναι ίσο με το πλάτος του αριθμού σελίδας στην εικονική διεύθυνση δηλ. 22 bits. Άρα το μέγεθος του πίνακα σελίδων είναι:
 - Μέγεθος πίνακα σελίδων = πλήθος γραμμών * μέγεθος γραμμής.
 - Μέγεθος πίνακα σελίδων = $2^{22} * 22 \text{ bits} = 11.5 \text{ Mbytes}!!$



7^η Άσκηση (1/5)

Θεωρείστε μια αρχιτεκτονική λογικής μνήμης με τις ακόλουθες παραμέτρους:

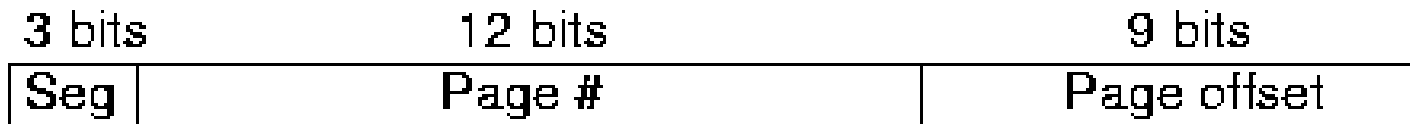
- Μνήμη που διευθυνσιοδοτείται ανά byte.
- 24-bit λογικές διευθύνσεις.
- Μέγεθος segment 2 MB.
- Μέγεθος σελίδας 512 bytes.

Κάθε πίνακας σελίδων πρέπει να χωρά σε μια σελίδα. Υποθέστε ότι κάθε καταχώρηση στον πίνακα σελίδων περιέχει ένα read και ένα write bit προστασίας και ότι το μέγιστο ποσό φυσικής μνήμης που υποστηρίζεται από την αρχιτεκτονική είναι 8MB. Σχεδιάστε τη λογική διεύθυνση, δείχνοντας τον αριθμό των bits που χρησιμοποιούνται ως δείκτης σε κάθε επίπεδο των πινάκων σελίδων. (Υπόδειξη: Πρέπει να υπολογίσετε το μέγεθος κάθε καταχώρησης (PTE) στον πίνακα σελίδων. Ως ενδιάμεσο βήμα, ίσως θεωρήσετε χρήσιμο να υπολογίσετε τον αριθμό που χωρούν στη φυσική μνήμη. Μετά, πόσα bits σε κάθε PTE απαιτούνται για να σχεδιάσετε μία από αυτές τις σελίδες; Προσθέστε τα 2 bits προστασίας για να υπολογίσετε το μέγεθος κάθε PTE.)



7^η Λύση Άσκησης (2/5)

- Από τις αρχικές παραμέτρους προκύπτει :
- Μέγεθος segment $2\text{MB}=2*1\text{MB}=2*2^{20}=2^{21}$ bits.
- Εφόσον η λογική διεύθυνση έχει 24 bits απομένουν $24-21=3$ bits για τον αριθμό segments.
- Επειδή μέγεθος σελίδας = 512 bytes = 2^9 bits =offset και απομένουν $21-9=12$ bits για τον αριθμό σελίδας (page number).



7^η Λύση Άσκησης (3/5)

- Με μέγεθος φυσικής μνήμης $8\text{MB} = 2^{23}$ bits.
για τον αριθμό πλαισίων απομένουν $23-9=14$ bits (ο αριθμός πλαισίων αναφέρεται και ως πλήθος των σελίδων φυσικής μνήμης (**Physical Page Number**)).
- (με τον γνωστό τρόπο το πλήθος των σελίδων φυσικής μνήμης είναι $2^{23}/2^9=2^{14}$ δηλαδή απαιτούνται 14 bits για το πλήθος των σελίδων φυσικής μνήμης).
- Άρα κάθε καταχώρηση (entry) στον πίνακα σελίδων απαιτεί $14 + 2$ (read-write) = 16 bits = 2 bytes.



7^η Λύση Άσκησης (4/5)

- Για να χωρά κάθε πίνακας σελίδων σε μια σελίδα ($=2^9$ bits) ο μέγιστος αριθμός PTE (Page Table Entries) θα είναι $2^9 / 2 = 2^8$.
- Άρα το πολύ 8 bits μπορούν να χρησιμοποιηθούν ως δείκτης στον πίνακα σελίδων που χωρά σε μια σελίδα.
- Επειδή βρήκαμε ότι το πλήθος των bits που χρησιμοποιούνται για σελίδες είναι 12 έχουμε τουλάχιστον σελιδοποίηση με ένα επίπεδο 8 bits και ένα με 4 bits (δηλαδή συνολικά 2 επίπεδα).



7^η Λύση Άσκησης (5/5)

Από αυτά το επίπεδο των 4 bits είναι το εξωτερικό (πλησιέστερα προς το segment) και το επίπεδο των 8 bits το εσωτερικό (πλησιέστερα προς το offset):

3 bits	4 bits	8 bits	9 bits
Seg	Outer Page #	Inner Page #	Page offset



8^η Άσκηση

- Σε μια αρχιτεκτονική χρησιμοποιούνται λογικές διευθύνσεις των 32 bits χωρισμένες ως εξής:
 - *4-bit segment number* / *12-bit page number* / *16-bit offset*.
 - Ποιο είναι το μέγεθος σελίδας;
 - Ποιο είναι το μέγιστο μήκος τμήματος;



9^η Άσκηση

- Υποθέστε ότι μια διεργασία χωρίζεται σε 4 ίσου μεγέθους τμήματα, και ότι το σύστημα δημιουργεί έναν πίνακα σελίδων 8 θέσεων για κάθε τμήμα (συνδυασμός κατάτμησης και σελιδοποίησης). Υποθέστε επίσης ότι το μέγεθος σελίδας είναι 2 Kb.
- **Ποιο είναι το μέγιστο μέγεθος κάθε τμήματος;**
- **Ποια είναι ο μέγιστος λογικός χώρος διευθύνσεων για κάθε διεργασία;**
- **Ποια είναι η μορφή μιας λογικής διεύθυνσης στην περίπτωση αυτή;**



10^η Άσκηση

- Θεωρείστε τον ακόλουθο πίνακα τμημάτων:

τμήμα	βάση	μήκος
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

Ποιες φυσικές διευθύνσεις αντιστοιχούν στις παρακάτω λογικές διευθύνσεις;

i. 0,430

ii. 1,10

iii. 2,500

iv. 3,400

v. 4,112



11^η Άσκηση

Τρεις διεργασίες A,B,C φορτώνονται στην κεντρική μνήμη. Οι απαιτήσεις τους φαίνονται στον παρακάτω πίνακα:

Process	Code Segment	Data Segment	Stack Segment
A	492	438	2009
B	4034	1030	610
C	8900	914	1120

Το Λ.Σ. υποστηρίζει σελιδοποίηση με μέγεθος σελίδας 512 bytes. Ποιο ποσό φυσικής μνήμης σπαταλιέται για καθεμία από τις διεργασίες λόγω κατακερματισμού; Τι είδους είναι ο κατακερματισμός;

Υποθέστε ότι το Λ.Σ. (και το hardware) μπορεί να υποστηρίξει κατάτμηση με μέγεθος σελίδας 512 bytes. Κάθε τμήμα μπορεί να φορτωθεί ξεχωριστά. **Ποιο ποσό φυσικής μνήμης σπαταλιέται λόγω κατακερματισμού** (θεωρείστε ότι κάθε τμήμα καταλαμβάνει ακέραιο αριθμό σελίδων και ότι ξεκινά στα όρια της σελίδας);



12^η Άσκηση

- Ένα σύστημα χρησιμοποιεί ως εικονικές διευθύνσεις 2048 σελίδες μεγέθους η κάθε μία 256 bytes και αντιστοιχίζεται σε μια φυσική μνήμη 512 πλαισίων. Η μικρότερη μονάδα προσπέλασης είναι 1 byte.
 - Μια διεργασία P1 χρησιμοποιεί 1053 bytes. Πόσα πλαίσια σελίδων θα απαιτήσει από την MMU (Memory Management Unit – μονάδα διαχείρισης μνήμης);
 - Μια άλλη διεργασία P2 απαιτεί 4000 bytes. Θεωρείστε ότι η P2 έχει ήδη αποκτήσει μέγεθος φυσικής μνήμης 2048 bytes, που ξεκινά από τη διεύθυνση 0. Έστω ότι ο επεξεργαστής χρειάζεται να προσπελάσει τη διεύθυνση: 10η σελίδα, μετατόπιση 34, στο πρόγραμμα της διεργασίας P2. Μπορεί να υπάρξει σφάλμα σελίδας; Δικαιολογείστε την απάντησή σας.



13^η Άσκηση

- Ένα σύστημα εικονικής μνήμης χρησιμοποιεί σελιδοποίηση με εικονικές διευθύνσεις μεγέθους 64 bits και μέγεθος σελίδας 16KB. Κάθε είσοδος στον πίνακα σελίδων απαιτεί 128 bits. Ο πίνακας σελίδων πρέπει να χωρά σε μία σελίδα.

Πόσα επίπεδα πινάκων σελίδων απαιτούνται και πόσες εισοδοι υπάρχουν στον πίνακα σελίδων σε κάθε επίπεδο;



14^η Άσκηση

- Σε ένα σύστημα που χρησιμοποιεί σελιδοποίηση κάθε είσοδος στον πίνακα σελίδων έχει μέγεθος 4 bytes. Το μέγεθος σελίδας είναι 4K. Αν ο εικονικός χώρος διευθύνσεων χρησιμοποιεί 64 bits και με την προϋπόθεση ότι ο πίνακας σελίδων χωρά σε μια σελίδα:
 - **Μπορεί το σύστημα να χρησιμοποιεί απλή σελιδοποίηση; Να δικαιολογήσετε την απάντησή σας.**
 - **Αν απαιτείται σελιδοποίηση πολλαπλών επιπέδων , διατηρώντας τις προηγούμενες παραδοχές (δηλαδή ότι ο εικονικός χώρος διευθύνσεων χρησιμοποιεί 64 bits και κάθε πίνακας σελίδων χωρά σε μια σελίδα μεγέθους 4K) να βρείτε πόσα τουλάχιστον επίπεδα πρέπει να χρησιμοποιήσει μια σελιδοποίηση πολλαπλών επιπέδων.**



15^η Άσκηση

- Δίνεται η παρακάτω μορφή λογικής διεύθυνσης:

2 bits	16 bits	8 bits
Seg	Page #	Page offset

Να βρεθούν :

1. Το πλήθος των segments.
2. Το μέγιστο μέγεθος κάθε segment.
3. Το μέγεθος κάθε σελίδας.
4. Το μέγιστο πλήθος σελίδων ανά segment.
5. Το μέγιστο μέγεθος κάθε πίνακα σελίδας (ανά τμήμα). Υποθέστε ότι κάθε καταχώρηση στον πίνακα σελίδων (PTE) απαιτεί 4 bytes.



16^η Άσκηση

Ένα σύστημα χρησιμοποιεί χώρο διευθύνσεων 32 bits και έχει μέγεθος σελίδας 8K. Ο πίνακας σελίδων βρίσκεται εξ ολοκλήρου στο υλικό και κάθε είσοδός του έχει μήκος 32 bits. Όταν μια διεργασία καθίσταται εκτελέσιμη ο πίνακας σελίδων αντιγράφεται από το υλικό στη μνήμη με ταχύτητα 100 nsec για κάθε είσοδό του. Αν κάθε διεργασία εκτελείται για 100 msec (περιλαμβάνεται ο χρόνος φόρτωσης του πίνακα σελίδων), **να βρεθεί το ποσοστό % του χρόνου της CPU που αφιερώνεται για τη φόρτωση του πίνακα σελίδων**, για κάθε διεργασία.



17^η Άσκηση

- Ένα σύστημα κρατά τους πίνακες σελίδων των διεργασιών στην κύρια μνήμη. Ο χρόνος ανάγνωσης μιας εισόδου του πίνακα σελίδων είναι 50 nsec. Για να μειωθεί η καθυστέρηση το σύστημα χρησιμοποιεί TLB στο οποίο η αναζήτηση διαρκεί 10 nsec.
- Ποιο πρέπει να είναι το ποσοστό επιτυχίας (hit – rate) του TLB ώστε ο χρόνος ανάγνωσης να μειωθεί σε 20 nsec;



18^η Άσκηση (1/2)

Ένα σύστημα σελιδοποίησης χρησιμοποιεί διευθύνσεις των 16 bits, με μέγεθος σελίδας 4K και συνολικά 8 πλαίσια φυσικής μνήμης. Δίνονται οι πίνακες σελίδων των διεργασιών P1 και P2.

	P1		P2
0	0	0	3
1	4	1	1
2	5	2	7
3	2	3	6



18^η Άσκηση (2/2)

Να βρεθούν οι φυσικές διευθύνσεις των παρακάτω λογικών διευθύνσεων:

- Λογική διεύθυνση 15000 της διεργασίας P1.
- Λογική διεύθυνση 12000 της διεργασίας P2.

Να δώσετε αναλυτικά τα βήματα που οδηγούν στο τελικό αποτέλεσμα .



19^η Άσκηση (1/2)

Ένα σύστημα χρησιμοποιεί απλή σελιδοποίηση μνήμης με μέγεθος σελίδας 1024bytes. Το μέγιστο μέγεθος του χώρου διευθύνσεων μιας διεργασίας είναι 16 MB ενώ η μνήμη του συστήματος είναι 2MB. Ο πίνακας σελίδων της τρέχουσας διεργασίας περιέχει τα εξής:

page	Frame number
0	4
1	8
2	16
3	17
4	9



19^η Άσκηση (2/2)

- Πόσα bits απαιτούνται για κάθε είσοδο του πίνακα σελίδων;
- Ποιο είναι το μέγιστο μέγεθος εισόδων σε ένα πίνακα σελίδων;
- Πόσα bits υπάρχουν σε μια εικονική διεύθυνση;
- Σε ποια φυσική διεύθυνση αντιστοιχεί η εικονική διεύθυνση 1524;
- Σε ποια εικονική διεύθυνση μεταφράζεται η φυσική διεύθυνση 10020;



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
Ευρωπαϊκό Κοινωνικό Ταμείο

