

# Διαχείριση Μνήμης

Συστήματα Υπολογιστών

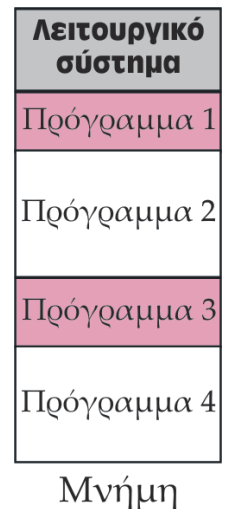
Παναγιώτης Παπαδημητρίου

[papadimitriou@uom.edu.gr](mailto:papadimitriou@uom.edu.gr)

<https://sites.google.com/site/panagpapadimitriou/>

# Διαχείριση Μνήμης

- Μονοπρογραμματισμός
  - Η μνήμη χωρίζεται σε δύο τμήματα, ένα για το Λ/Σ και ένα για τη φόρτωση μίας διεργασίας
  - Το Λ/Σ μπορεί να φορτώσει μόνο μια διεργασία στη μνήμη του Υ/Σ
  - Χαμηλή απόδοση, ειδικά με διεργασίες που αναλώνονται σε Ε/Ε
- Πολυπρογραμματισμός
  - Τεμαχισμός της μνήμης σε διάφορα τμήματα, ώστε κάθε διαφορετική διεργασία να καταλαμβάνει διαφορετικό τμήμα
  - Το Λ/Σ μπορεί να φορτώσει περισσότερες από μια διεργασίες ταυτόχρονα στη μνήμη του Υ/Σ
  - Κάθε χρονική στιγμή, υπάρχουν πολλές διεργασίες που είναι φορτωμένες και εκτελούνται από το Υ/Σ



# Πολυπρογραμματισμός

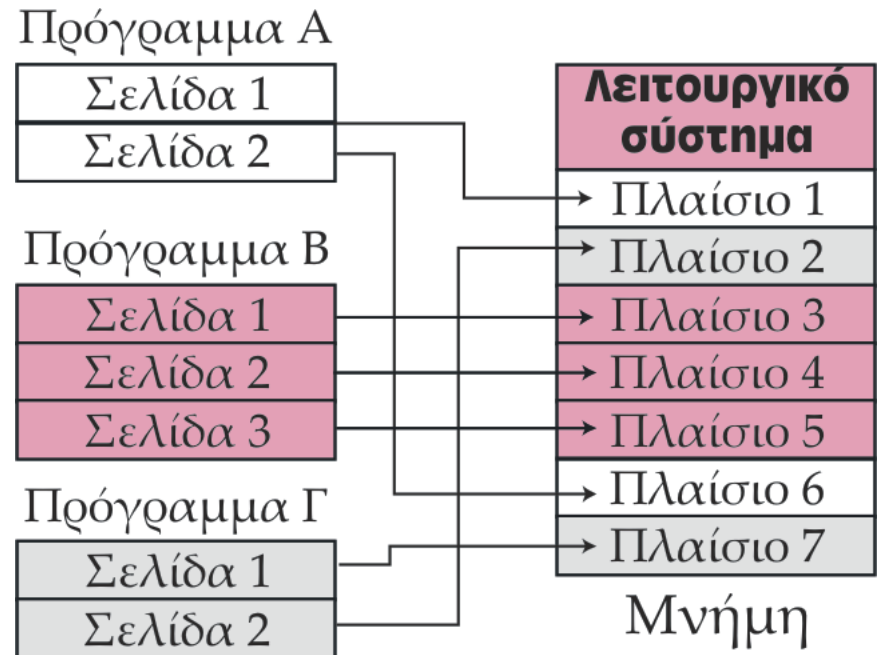
- Τεμαχισμός της μνήμης:
  - η μνήμη διαιρείται σε τμήματα σταθερού ή μεταβλητού μήκους (το καθένα αποθηκεύει τα δεδομένα ενός προγράμματος)



α. Η ΚΜΕ ξεκινάει την εκτέλεση του προγράμματος 1      β. Η ΚΜΕ ξεκινάει την εκτέλεση του προγράμματος 2

# Σελιδοποίηση (Paging)

- Σελιδοποίηση:
  - Η μνήμη χωρίζεται σε ισομεγέθη τμήματα που ονομάζονται πλαίσια (frames)
  - Το πρόγραμμα διαιρείται σε τμήματα ίδιου μεγέθους που ονομάζονται σελίδες (pages)
- Οι σελίδες ενός προγράμματος:
  - δεν χρειάζεται να είναι συνεχόμενες
  - βρίσκονται όλες στη μνήμη για την εκτέλεση



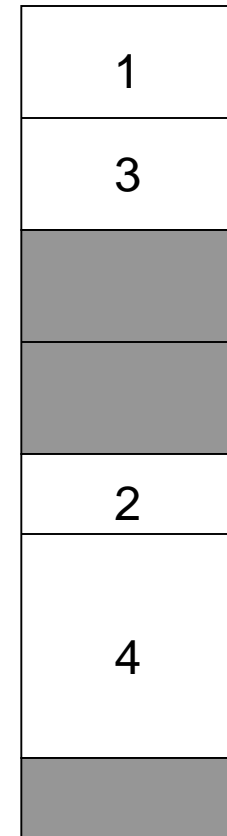
# Κατάτμηση (Segmentation)

- Η κατάτμηση αποτελεί μία εναλλακτική τεχνική τεμαχισμού της μνήμης:
  - Το πρόγραμμα χωρίζεται σε τμήματα διαφορετικού μεγέθους (σε αντίθεση με τη σελιδοποίηση όπου τα τμήματα έχουν το ίδιο μέγεθος)
- Τεχνική διαχείρισης μνήμης που μπορεί να εναρμονιστεί με την εικόνα του προγραμματιστή για τον τεμαχισμό του προγράμματος:
  - Κυρίως πρόγραμμα (main program)
  - Υποπρογράμματα, συναρτήσεις (procedures, functions)
  - Αντικείμενα (objects)
  - Καθολικές μεταβλητές (global variables)
  - Τοπικές μεταβλητές (local variables)

# Παράδειγμα Κατάτμησης



Λογικός χώρος προγράμματος

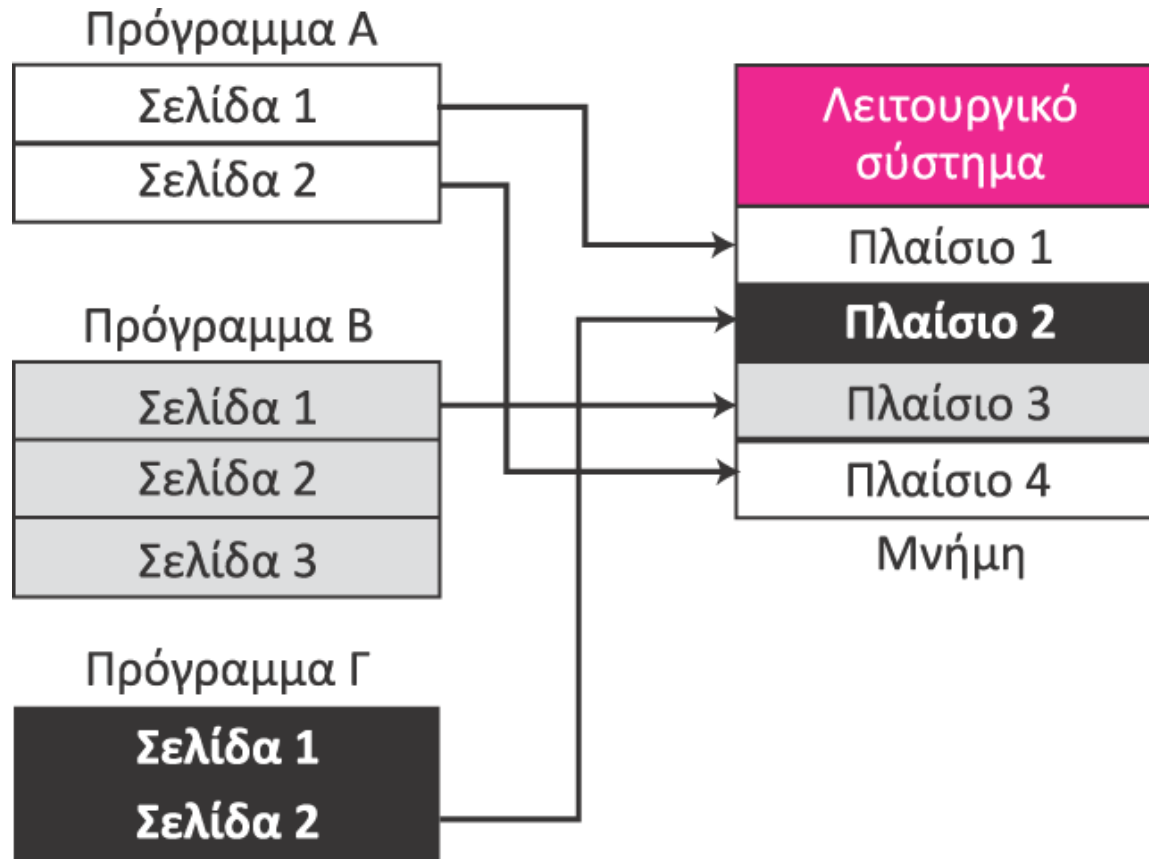


Φυσικός χώρος μνήμης

# Εναλλαγή (Swapping)

- Δεν χρειάζεται να φορτώνονται στη μνήμη όλα τα τμήματα ενός προγράμματος:
  - Τα τμήματα που δεν είναι φορτωμένα στη μνήμη αποθηκεύονται στο σκληρό δίσκο
- Εναλλαγή τμημάτων προγράμματος:
  - Τμήματα του προγράμματος μεταφέρονται από τη μνήμη στο δίσκο και αντίστροφα
- Η εναλλαγή μπορεί να συνδυαστεί τόσο με τη σελιδοποίηση όσο και με την κατάτμηση:
  - Σελιδοποίηση κατά απαίτηση
  - Κατάτμηση κατά απαίτηση

# Σελιδοποίηση κατά Απαίτηση





# Εικονική Μνήμη (Virtual Memory)

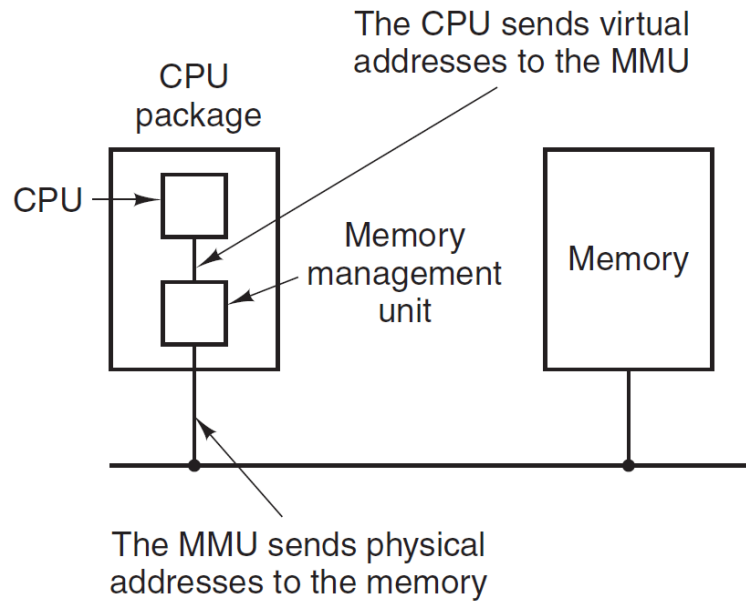
# Εικονική Μνήμη

- Η εικονική μνήμη αποτελεί ένα εικονικό (ή λογικό) χώρο διευθύνσεων που καλύπτει τόσο την κύρια μνήμη όσο και μέρος κάποιας μονάδας αποθήκευσης (συνήθως του σκληρού δίσκου)
- Η χρήση της εικονικής μνήμης απλουστεύει την εκτέλεση ενός προγράμματος χωρίς να έχει φορτωθεί ολόκληρο στη φυσική μνήμη του υπολογιστή
- Η εικονική μνήμη μπορεί να υλοποιηθεί με:
  - Σελιδοποίηση κατά απαίτηση
  - Κατάτμηση κατά απαίτηση

# Χώρος Εικονικών και Φυσικών Διευθύνσεων

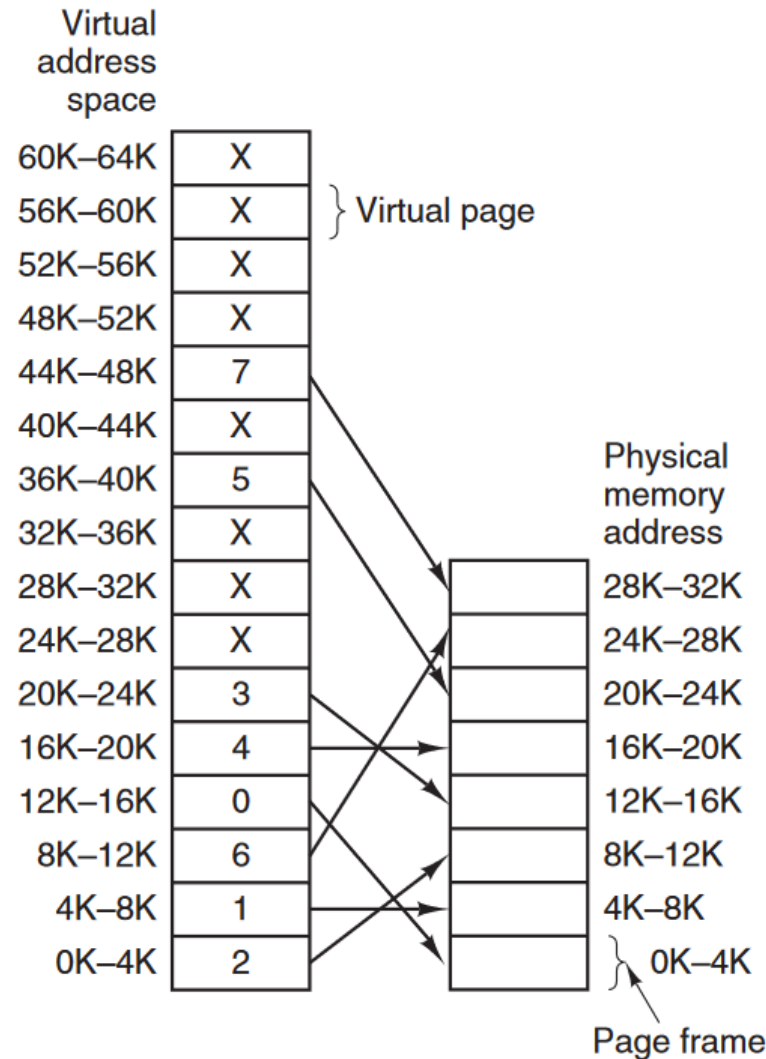
- Διαχωρισμός μεταξύ του χώρου **εικονικών** διευθύνσεων και του χώρου **φυσικών** διευθύνσεων:
  - **Λογική ή εικονική διεύθυνση** (logical or virtual address): Η διεύθυνση που βλέπει το πρόγραμμα του χρήστη
  - **Φυσική διεύθυνση** (physical address): Η διεύθυνση όπως την αντιλαμβάνεται η μονάδα μνήμης του υπολογιστή
- Με τις εικονικές διευθύνσεις υποστηρίζονται χώροι διευθύνσεων που ξεπερνούν τη διαθέσιμη φυσική μνήμη, οπότε το μέγεθος ενός προγράμματος μπορεί να ξεπερνά κατά πολύ τη διαθέσιμη φυσική μνήμη

# Μονάδα Διαχείρισης Μνήμης

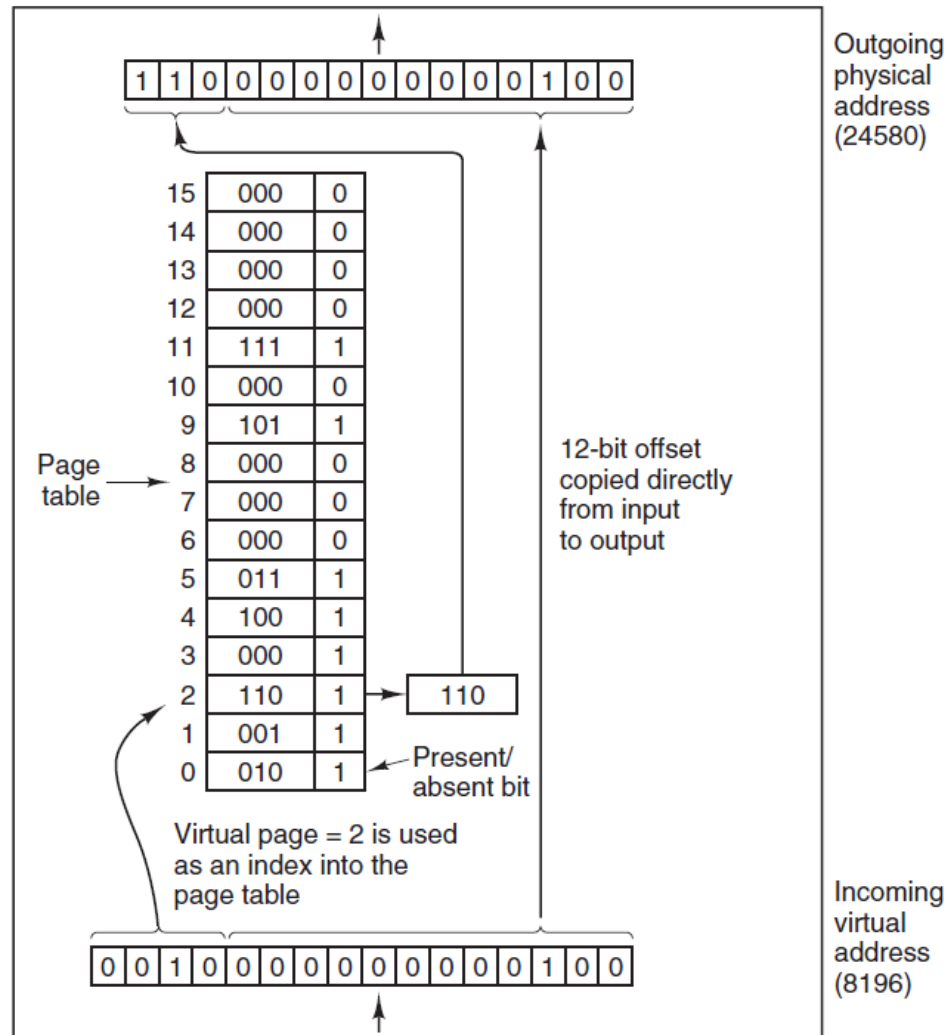


- Η Μονάδα Διαχείρισης Μνήμης (MMU) είναι υπεύθυνη για την αντιστοίχιση εικονικών διευθύνσεων σε φυσικές διευθύνσεις
- Η MMU αποτελεί συνήθως τμήμα της ΚΜΕ

# Απεικόνιση Εικονικής και Φυσικής Μνήμης



# Πίνακας Σελίδων (Page Table)



# Bit Εγκυρότητας

- Με κάθε καταχώρηση του πίνακα σελίδων συσχετίζεται ένα bit εγκυρότητας ( $1 \Rightarrow$  στη μνήμη,  $0 \Rightarrow$  στο μέσο αποθήκευσης)
- Κατά τη μετάφραση των διευθύνσεων, αν το bit εγκυρότητας στην καταχώρηση του πίνακα σελίδων είναι 0, τότε προκαλείται σφάλμα σελίδας

# Σφάλμα Σελίδας (Page Fault)

- Όταν προκύπτει σφάλμα σελίδας, το Λ/Σ εκτελεί τα παρακάτω βήματα:
  - Επιλέγει ένα άδειο πλαίσιο (τι γίνεται αν δεν υπάρχει άδειο πλαίσιο;)
  - Μεταφέρει τη σελίδα σε αυτό το πλαίσιο
  - Τροποποιεί την αντίστοιχη καταχώρηση του πίνακα σελίδων, ώστε να δείχνει στο πλαίσιο όπου είναι αποθηκευμένη η σελίδα
  - Ορίζει το bit εγκυρότητας σε 1, ώστε να δείχνει ότι η σελίδα είναι στη μνήμη
  - Επανεκκινεί τη διεργασία που προκάλεσε το σφάλμα σελίδας



# Αντικατάσταση Σελίδας

- Όταν δεν υπάρχει άδειο πλαίσιο στο οποίο να φορτωθεί μια σελίδα που μεταφέρεται από το μέσο αποθήκευσης, επιλέγεται ένα δεσμευμένο πλαίσιο
- Χρειάζεται ένας αλγόριθμος ο οποίος να οδηγεί σε ελάχιστο αριθμό σφαλμάτων σελίδας, και επομένως σε ελάχιστες μεταφορές σελίδων από και προς το μέσο αποθήκευσης

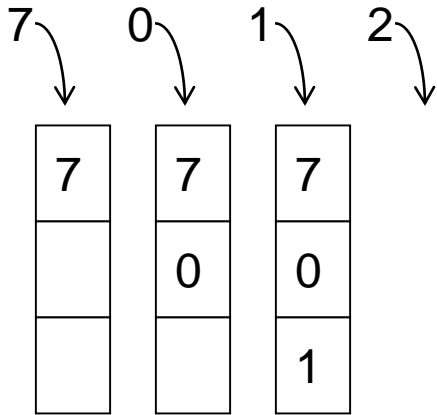
# Διαδικασία Αντικατάστασης Σελίδας

1. Εντοπισμός της θέσης της σελίδας στο μέσο αποθήκευσης
2. Εύρεση ελεύθερου πλαισίου:
  - Αν υπάρχει ελεύθερο πλαίσιο, χρησιμοποίησέ το
  - Αν δεν υπάρχει ελεύθερο πλαίσιο, χρησιμοποίησε αλγόριθμο εντοπισμό του πλαισίου θύματος για την επιλογή του πλαισίου που θα αντικατασταθεί
3. Μεταφορά της σελίδας στο ελεύθερο πλαίσιο
4. Ενημέρωση του πίνακα σελίδων
5. Επανεκκίνηση της διεργασίας

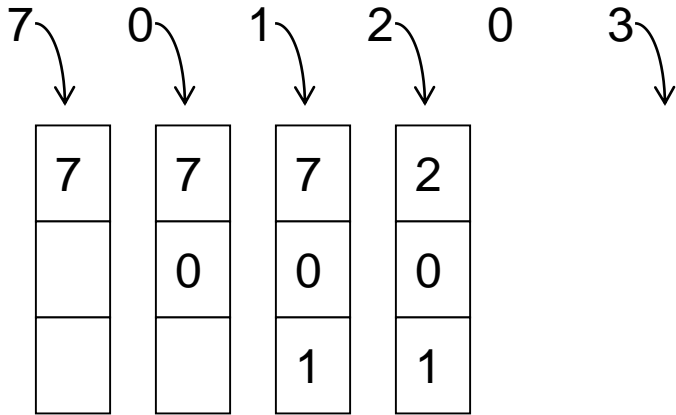
# Αλγόριθμος First-In-First-Out (FIFO)

- Η επιλογή της σελίδας (θύματος) προς αντικατάσταση γίνεται με βάση την «ηλικία» της σελίδας
- Ως θύμα επιλέγεται κάθε φορά η πιο παλιά σελίδα, δηλαδή αυτή που έχει παραμείνει στη μνήμη το μεγαλύτερο χρονικό διάστημα
- Δεν χρειάζεται η διατήρηση πληροφορίας για τη συχνότητα χρήσης των σελίδων και το πόσο πρόσφατα έχουν χρησιμοποιηθεί
- Ο αλγόριθμος FIFO παρουσιάζει το παράδοξο του Belady

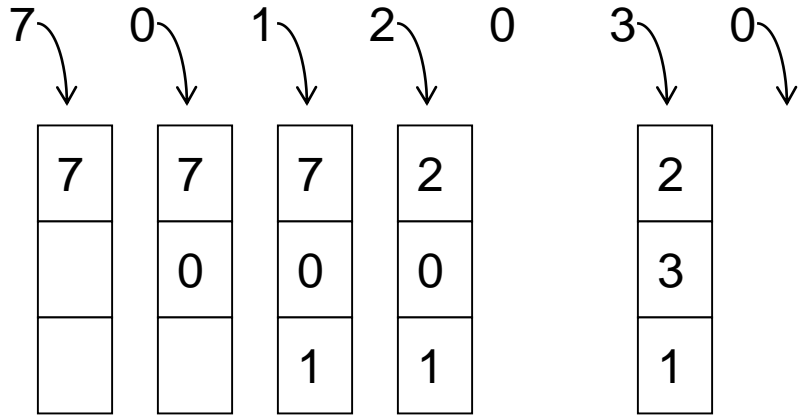
# Παράδειγμα Αλγορίθμου FIFO



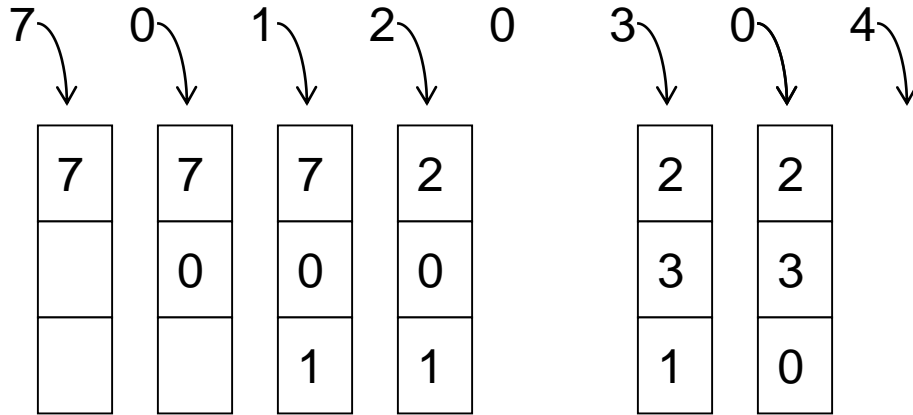
# Παράδειγμα Αλγορίθμου FIFO



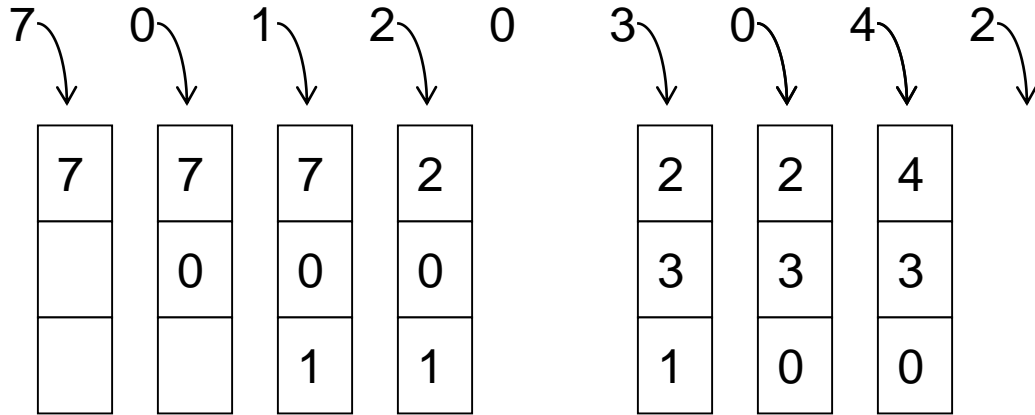
# Παράδειγμα Αλγορίθμου FIFO



# Παράδειγμα Αλγορίθμου FIFO

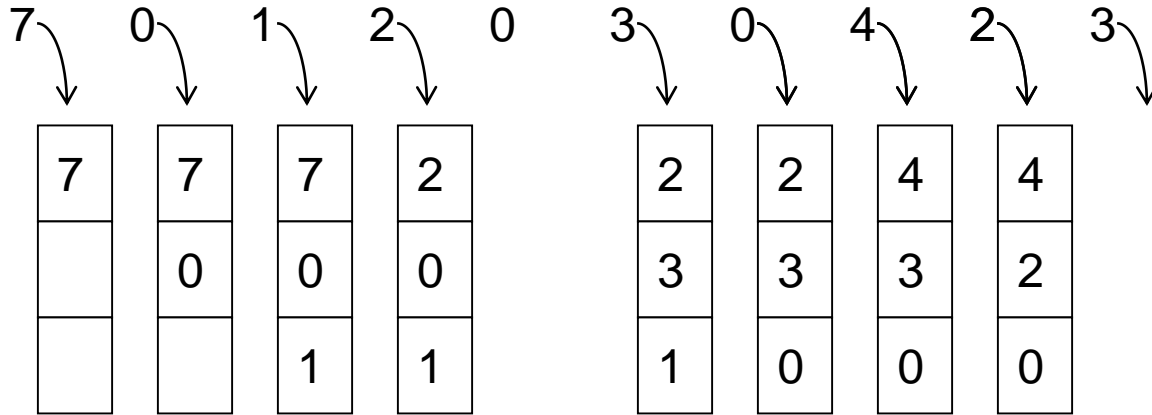


# Παράδειγμα Αλγορίθμου FIFO

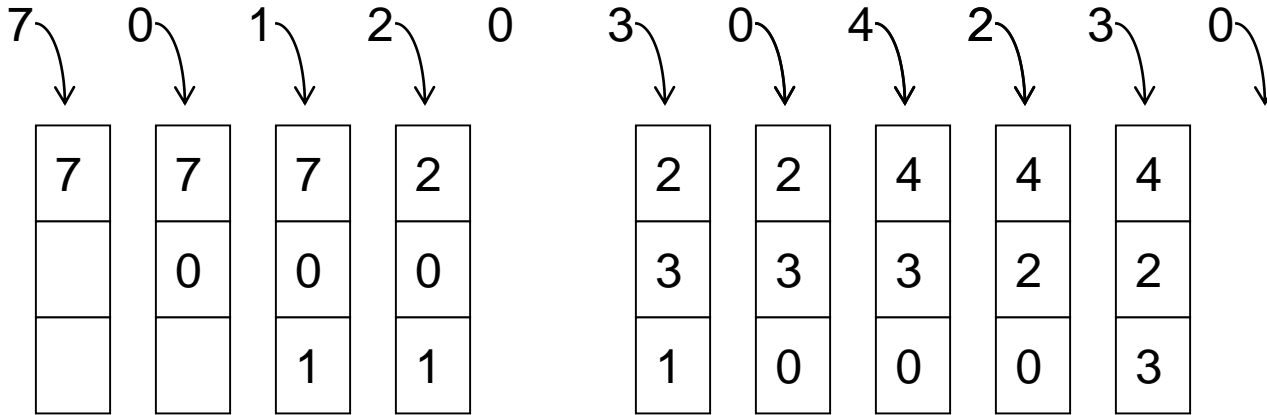




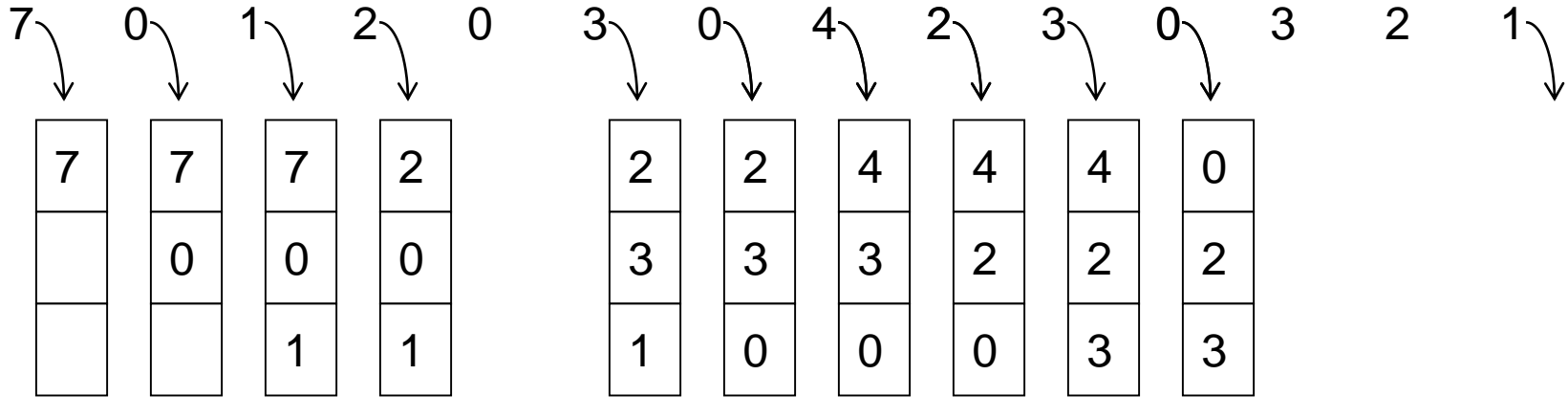
# Παράδειγμα Αλγορίθμου FIFO



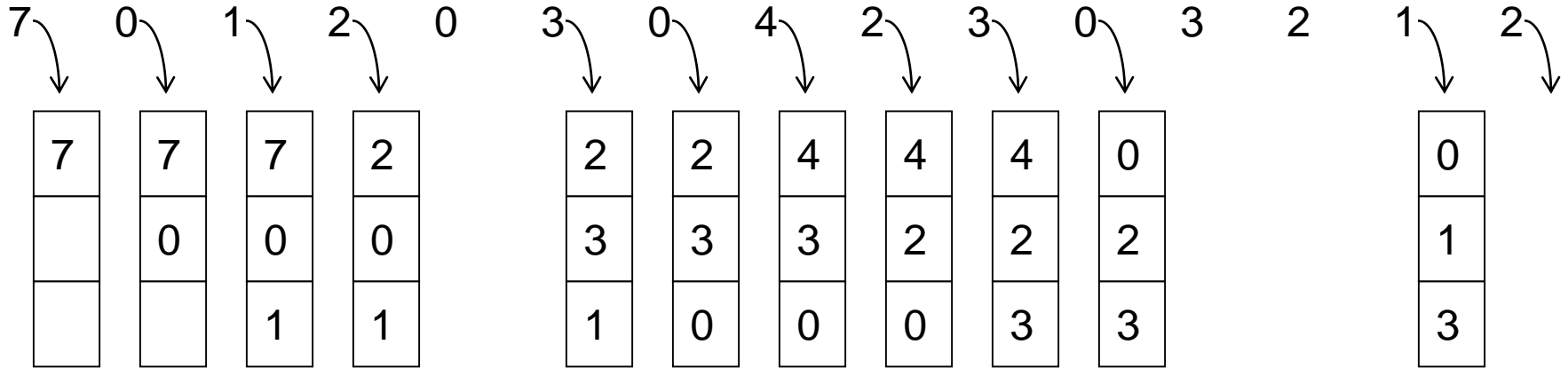
# Παράδειγμα Αλγορίθμου FIFO



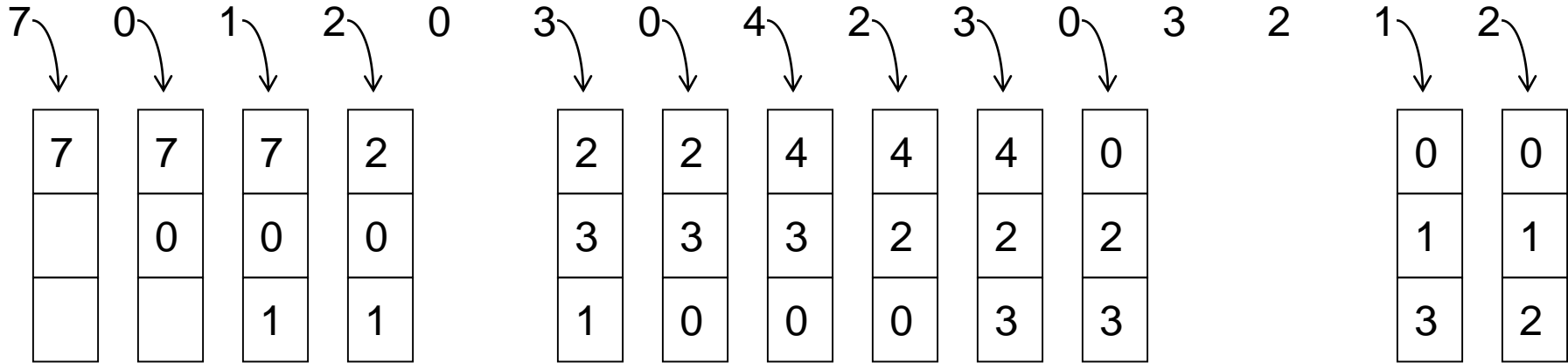
# Παράδειγμα Αλγορίθμου FIFO



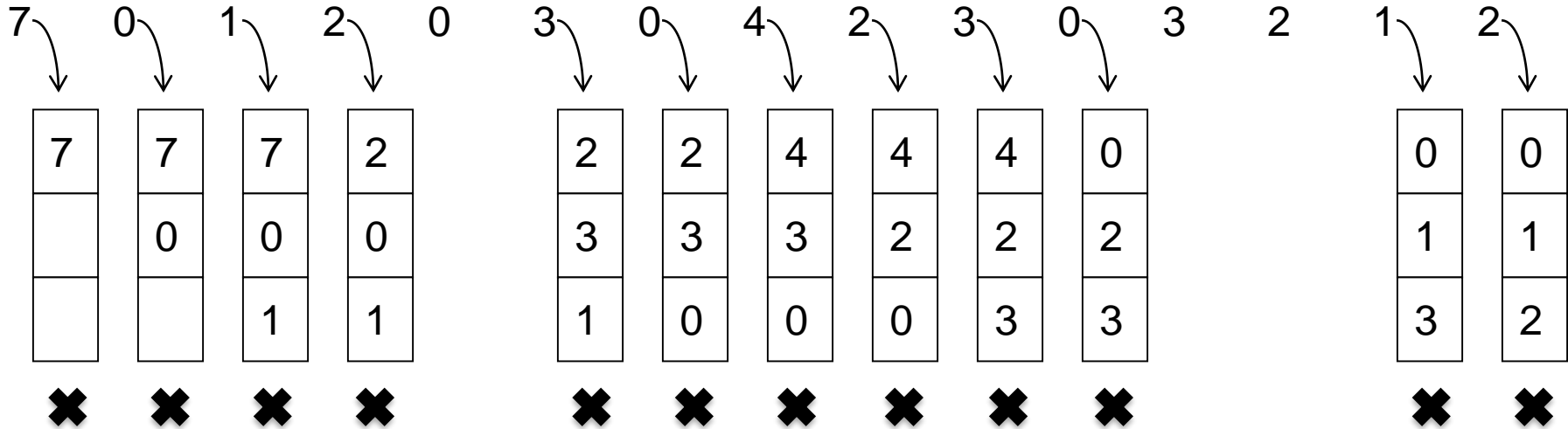
# Παράδειγμα Αλγορίθμου FIFO



# Παράδειγμα Αλγορίθμου FIFO

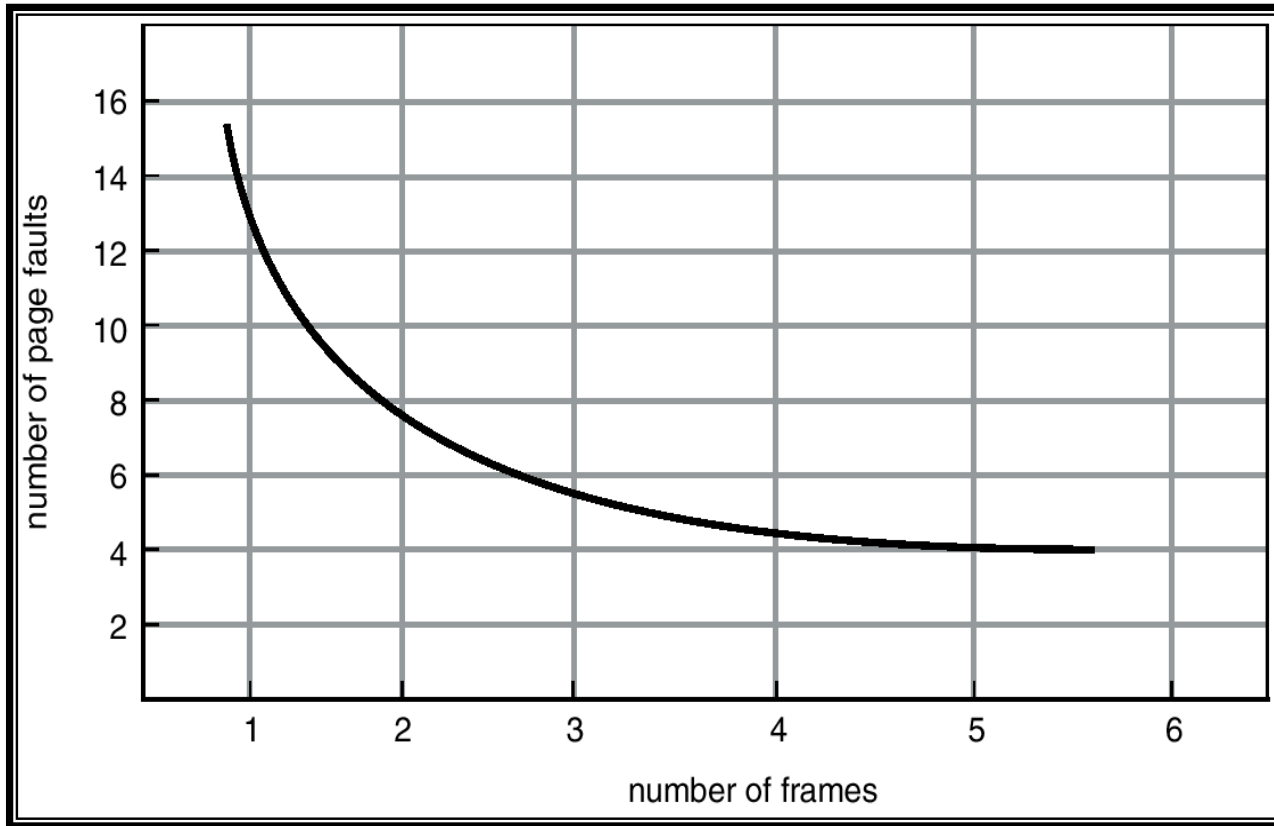


# Παράδειγμα Αλγορίθμου FIFO



12 σφάλματα σελίδας

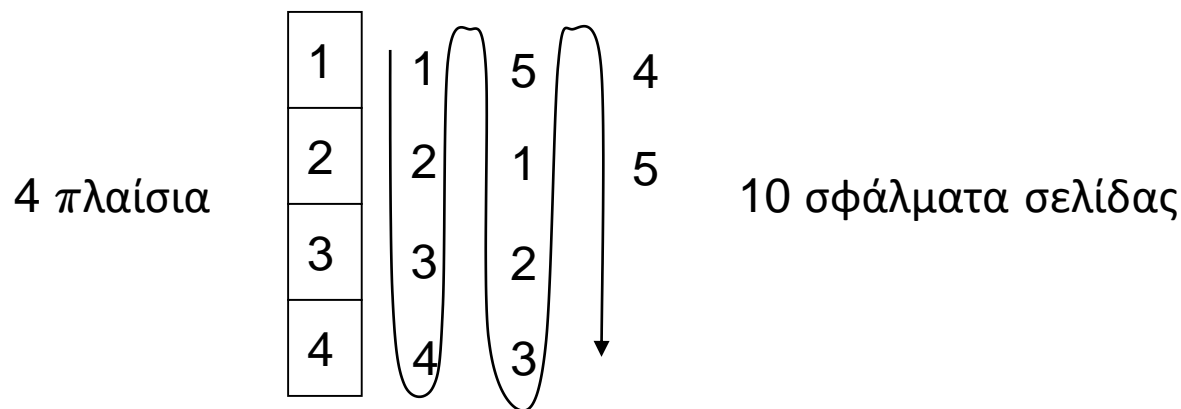
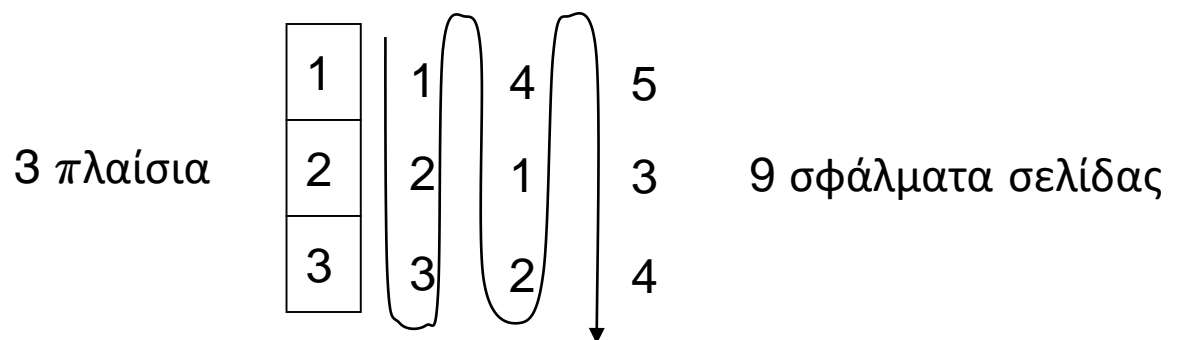
# Αναμενόμενη Συμπεριφορά Αλγορίθμου



- Μεγαλύτερος αριθμός πλαισίων αναμένεται να οδηγήσει σε λιγότερα σφάλματα σελίδας

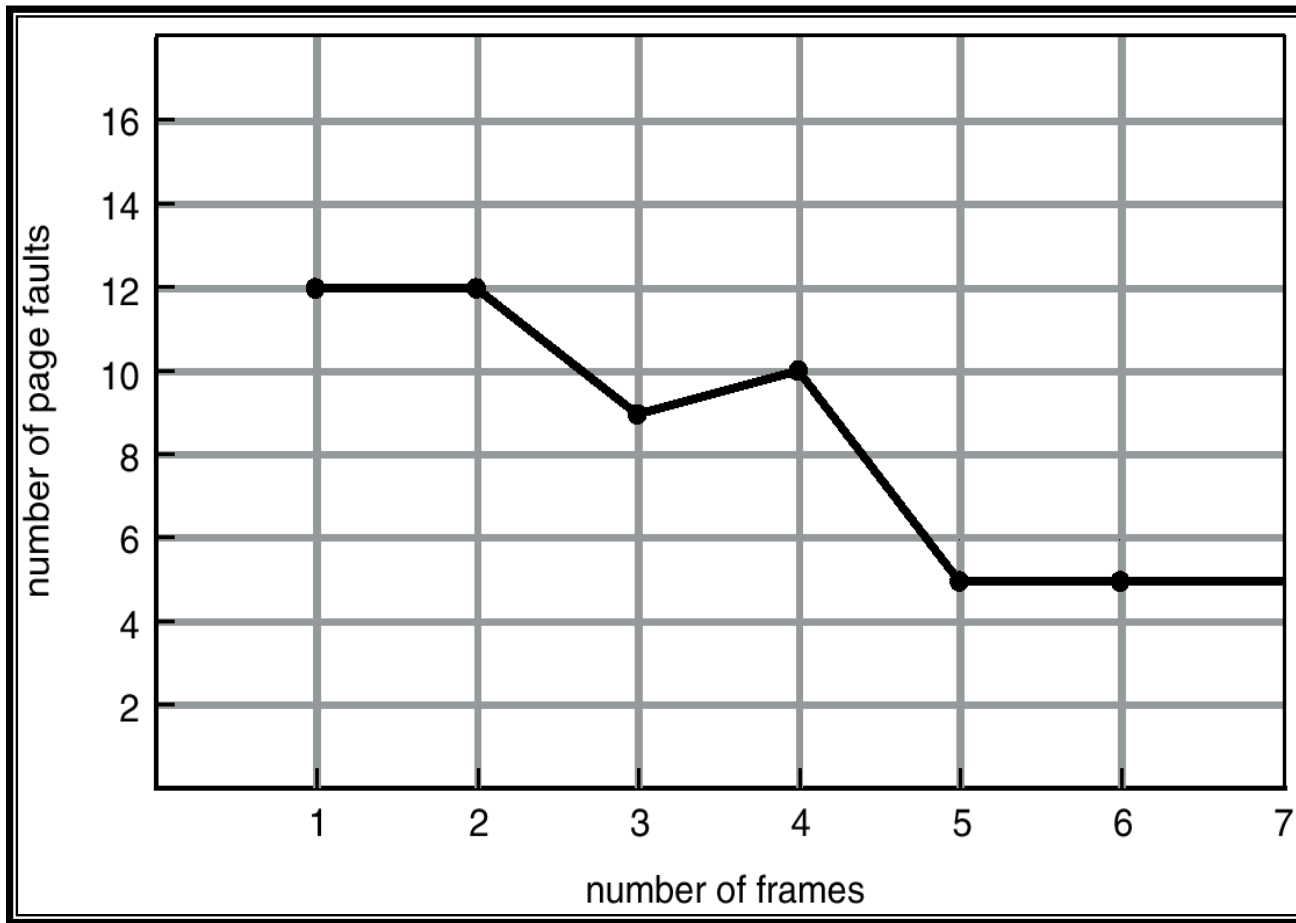
# Το Παράδοξο του Belady (1)

- Σελίδες αναφοράς: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5





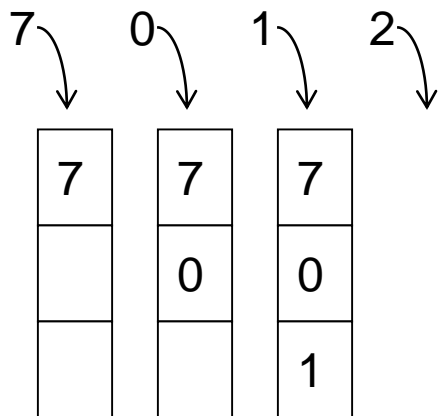
## Το Παράδοξο του Belady (2)



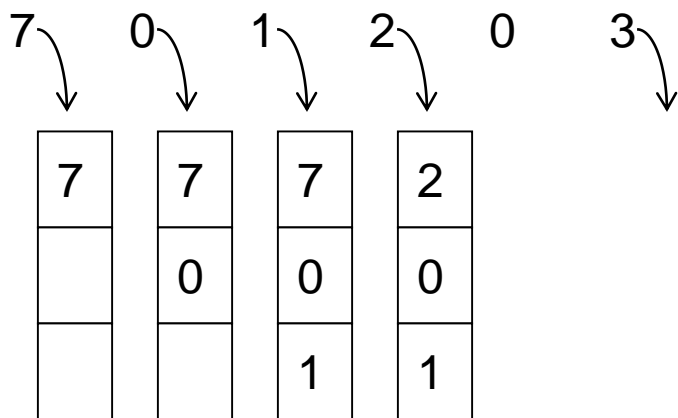
# Αλγόριθμος Least Recently Used (LRU)

- Επιλέγεται η σελίδα στην οποία έγινε αναφορά λιγότερο πρόσφατα
- Κάθε καταχώρηση στον πίνακα σελίδων έχει μια χρονοσφραγίδα (timestamp)
- Επιλογή της σελίδας με τη μικρότερη χρονοσφραγίδα
- Ο αλγόριθμος LRU δεν πάσχει από το παράδοξο του Belady

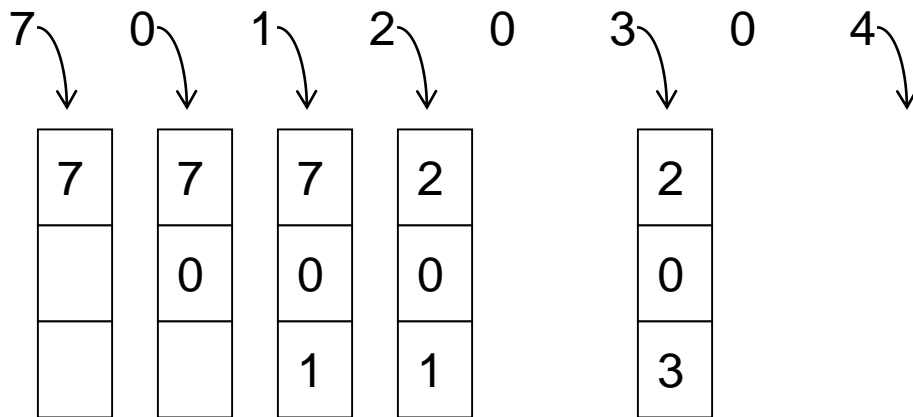
# Αντικατάσταση Σελίδων με τον LRU



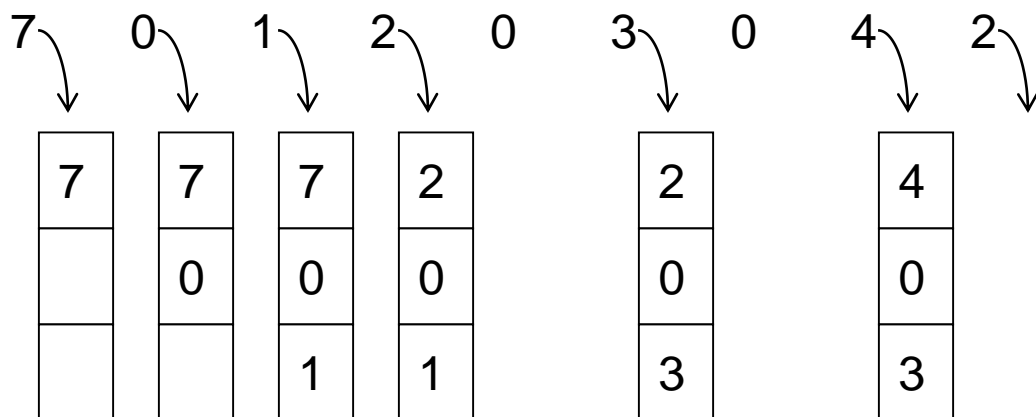
# Αντικατάσταση Σελίδων με τον LRU



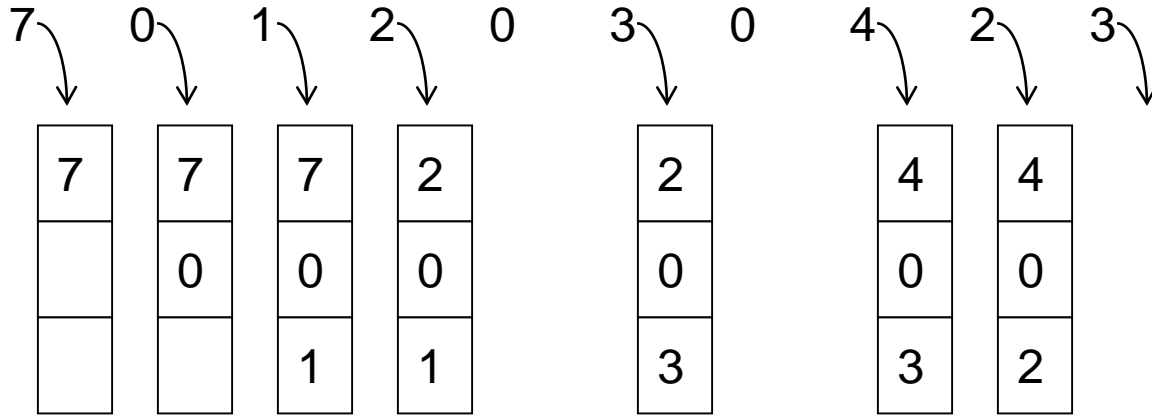
# Αντικατάσταση Σελίδων με τον LRU



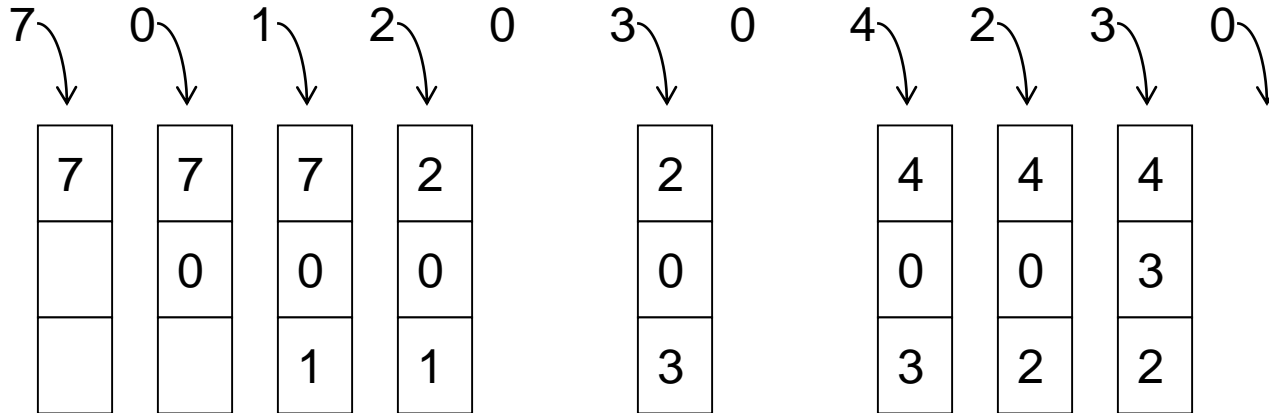
# Αντικατάσταση Σελίδων με τον LRU



# Αντικατάσταση Σελίδων με τον LRU

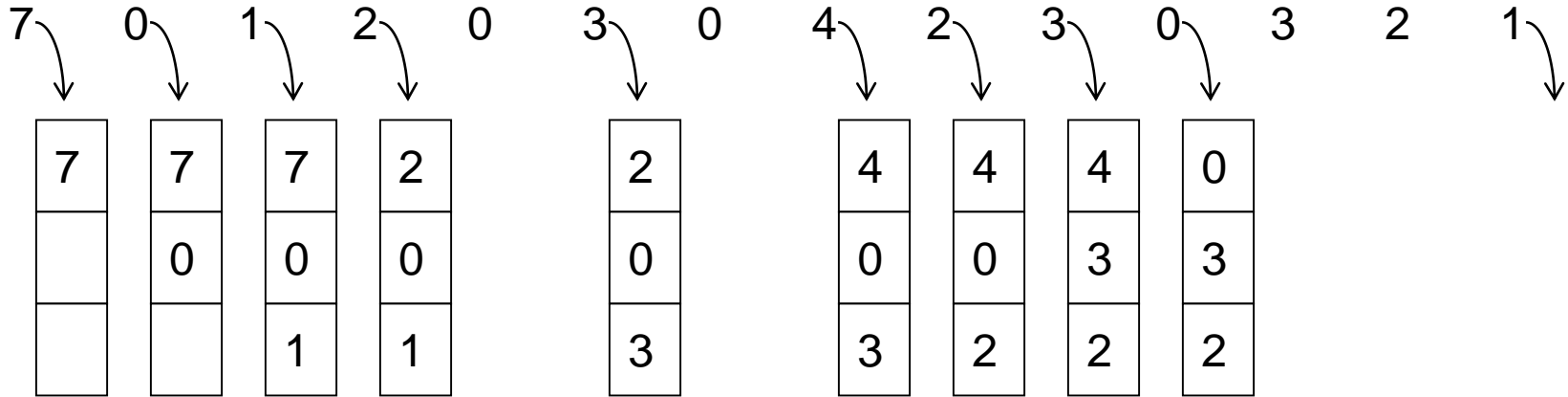


# Αντικατάσταση Σελίδων με τον LRU

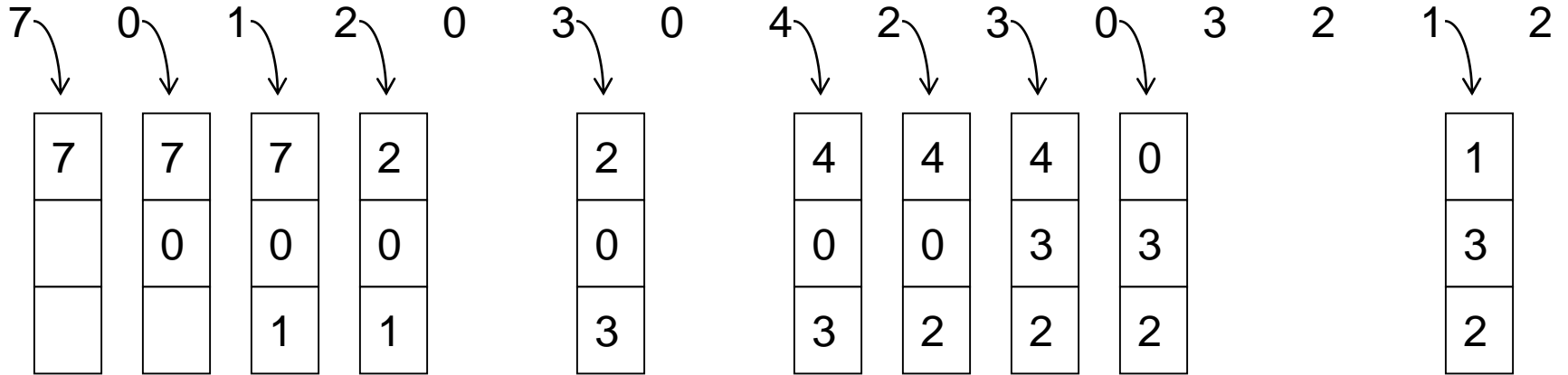




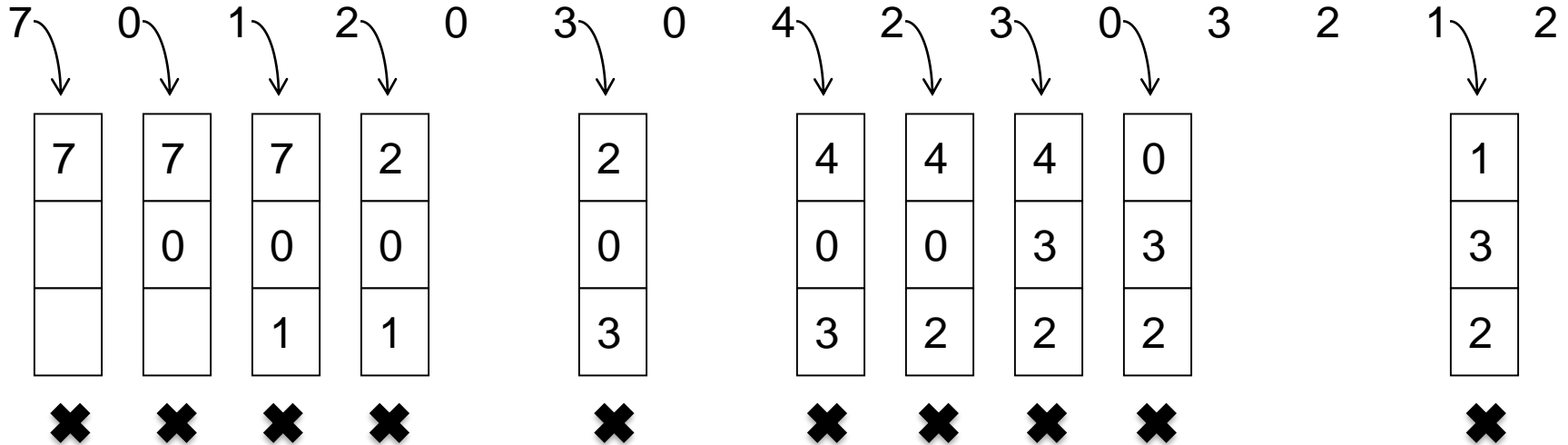
# Αντικατάσταση Σελίδων με τον LRU



# Αντικατάσταση Σελίδων με τον LRU



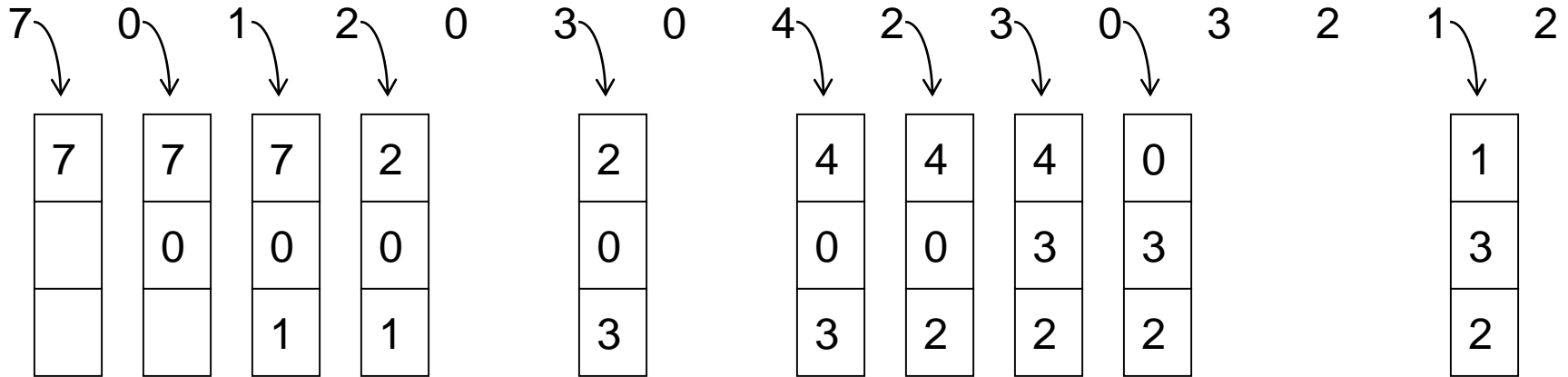
# Αντικατάσταση Σελίδων με τον LRU



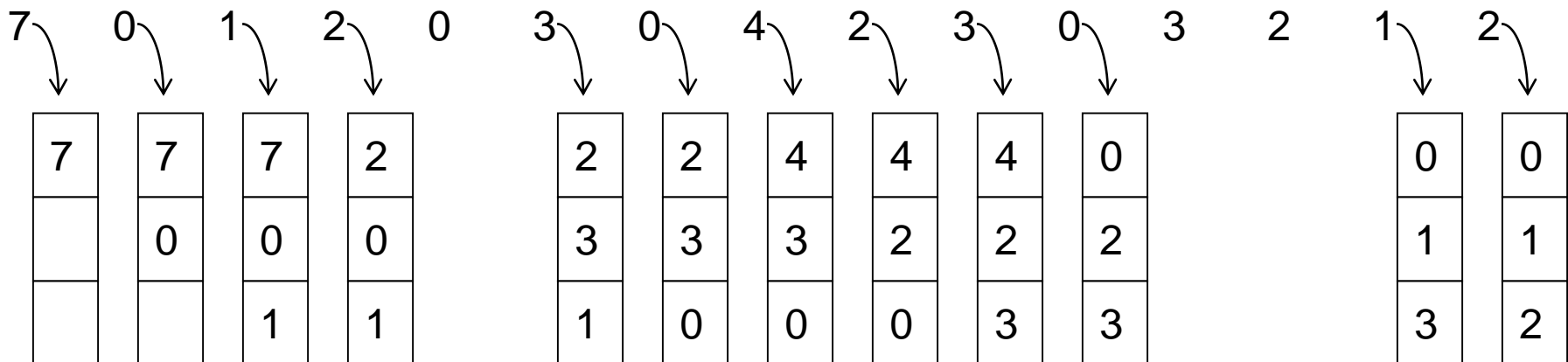
10 σφάλματα σελίδας

# Σύγκριση LRU και FIFO

LRU: 10 σφάλματα σελίδας



FIFO: 12 σφάλματα σελίδας



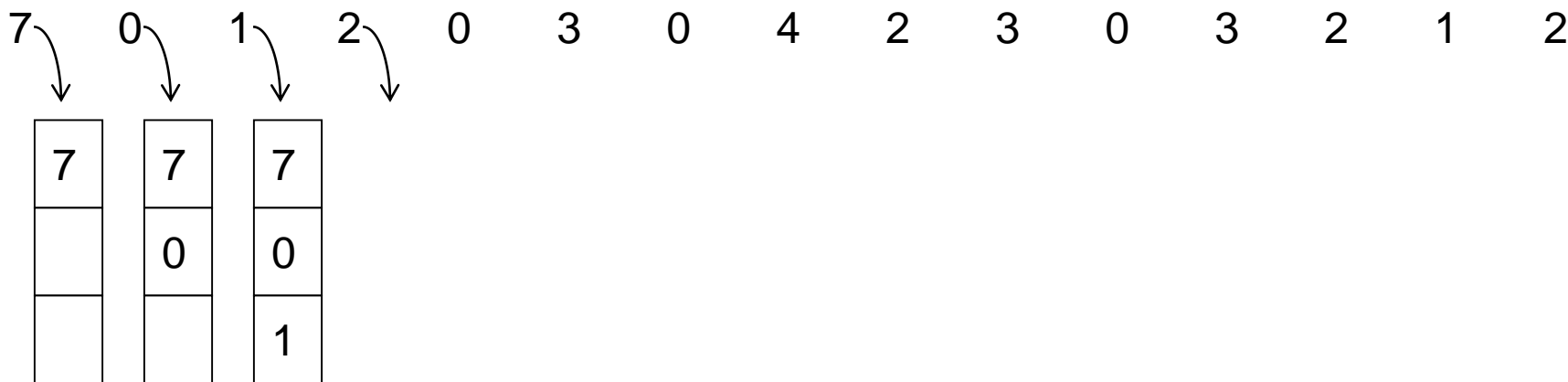
# Αλγόριθμοι Μέτρησης Συχνότητας Χρήσης

- Για κάθε σελίδα διατηρείται μετρητής για το πλήθος των αναφορών που έχουν γίνει στη σελίδα
- **Αλγόριθμος LFU** (Least Frequently Used): αντικαθιστά τη σελίδα της οποίας ο μετρητής έχει τη μικρότερη τιμή
  - Λογική: Σελίδες που έχουν χρησιμοποιηθεί συχνά, έχουν μεγάλη πιθανότητα να χρησιμοποιηθούν και στο μέλλον

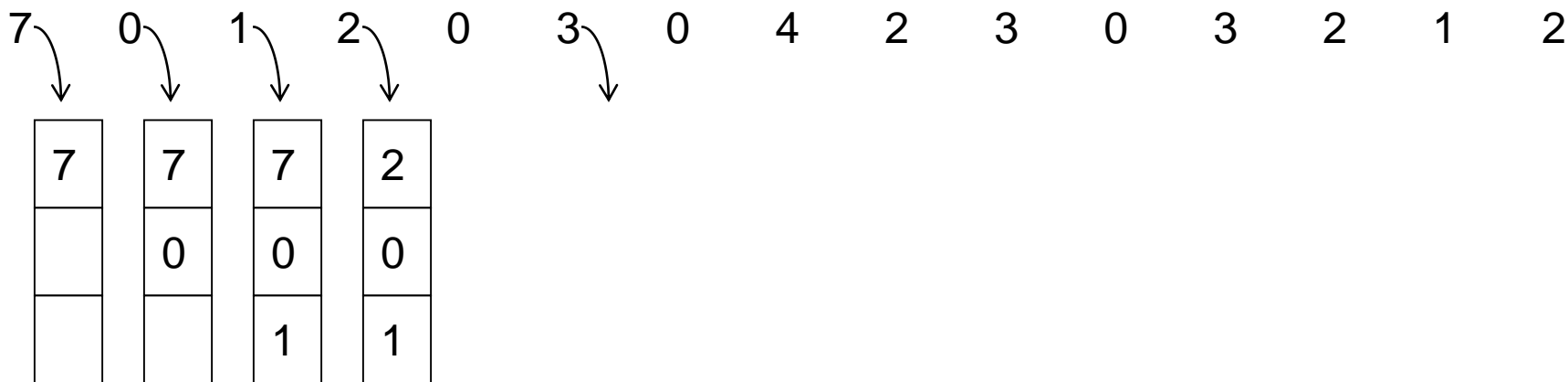
# Ο Βέλτιστος Αλγόριθμος

- Ποια είναι η ιδιότητα του ιδανικού ή βέλτιστου αλγορίθμου, που ελαχιστοποιεί τον αριθμό σφαλμάτων;
  - Αν ο αλγόριθμος αντικαθιστά κάθε φορά εκείνη την σελίδα που δεν πρόκειται να χρησιμοποιηθεί από το πρόγραμμα για το μεγαλύτερο χρονικό κατά την προσεχή εκτέλεση
- Πως όμως μπορούμε να υλοποιήσουμε αυτή την ιδιότητα;
- Ο βέλτιστος αλγόριθμος χρησιμοποιείται ουσιαστικά ως σημείο αναφοράς για την αποδοτικότητα των υπόλοιπων (ρεαλιστικά υλοποιήσιμων) αλγορίθμων (π.χ. FIFO, LRU, LFU)

# Βέλτιστη Αντικατάσταση Σελίδων

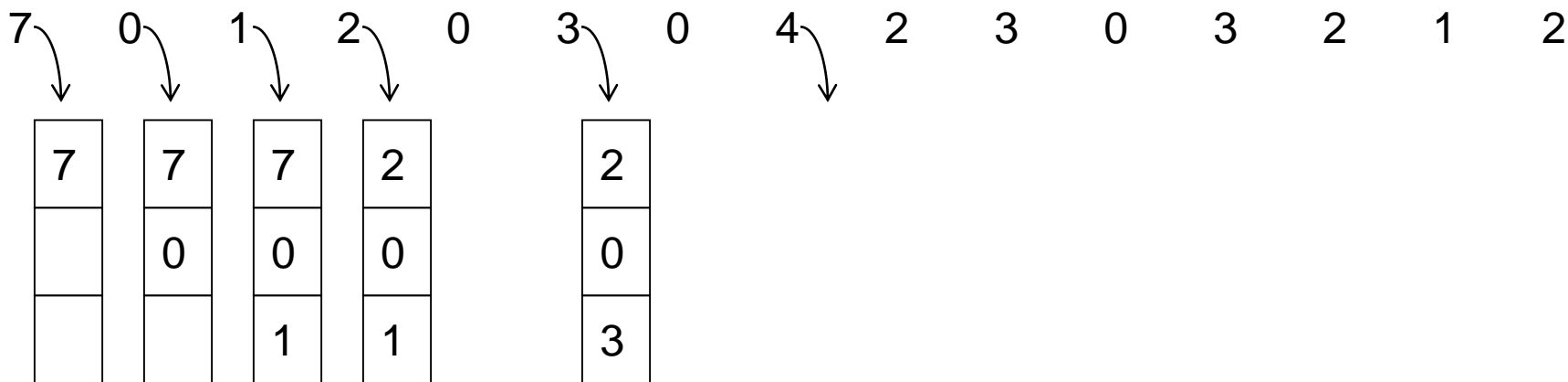


# Βέλτιστη Αντικατάσταση Σελίδων

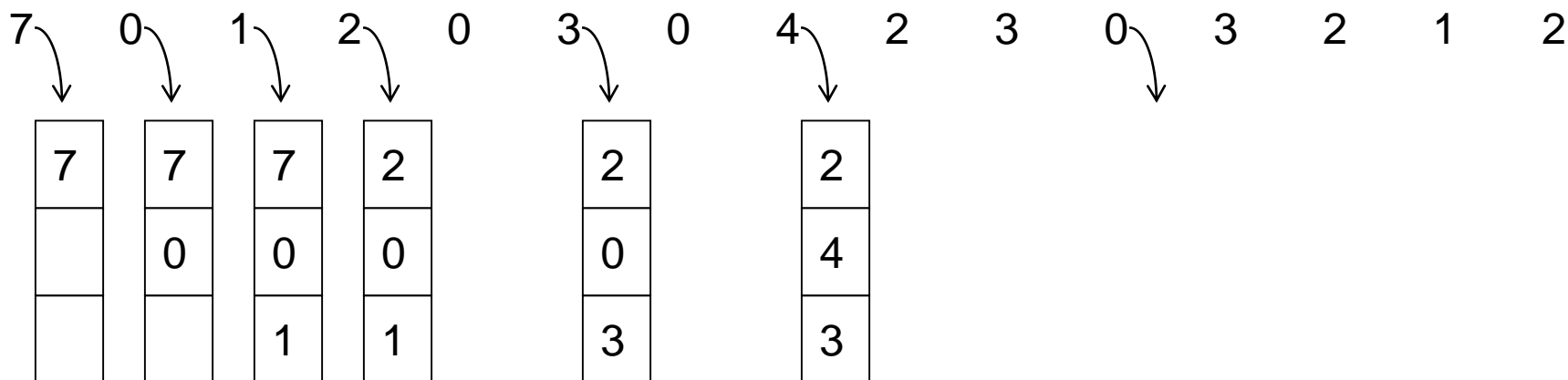




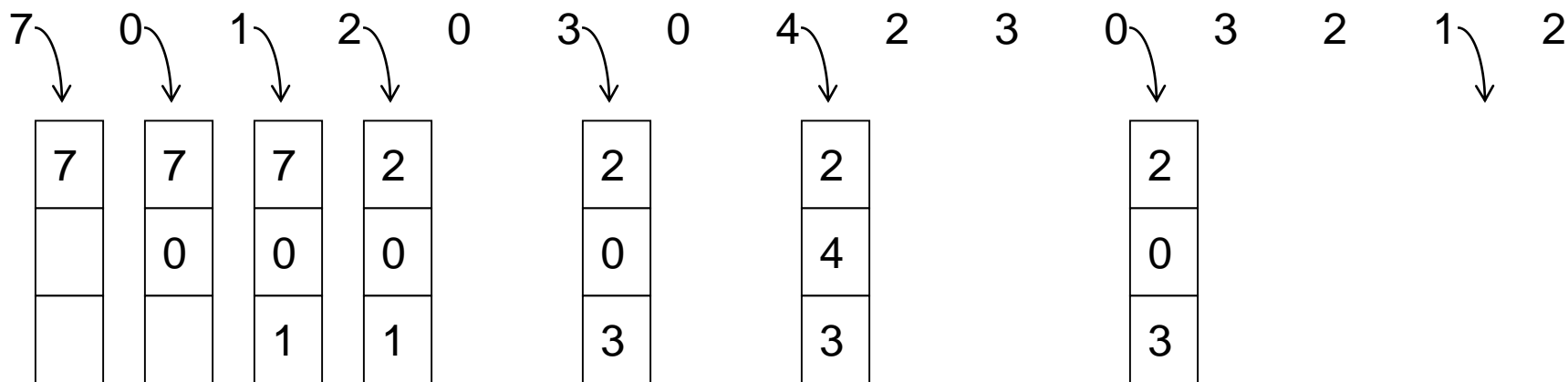
# Βέλτιστη Αντικατάσταση Σελίδων



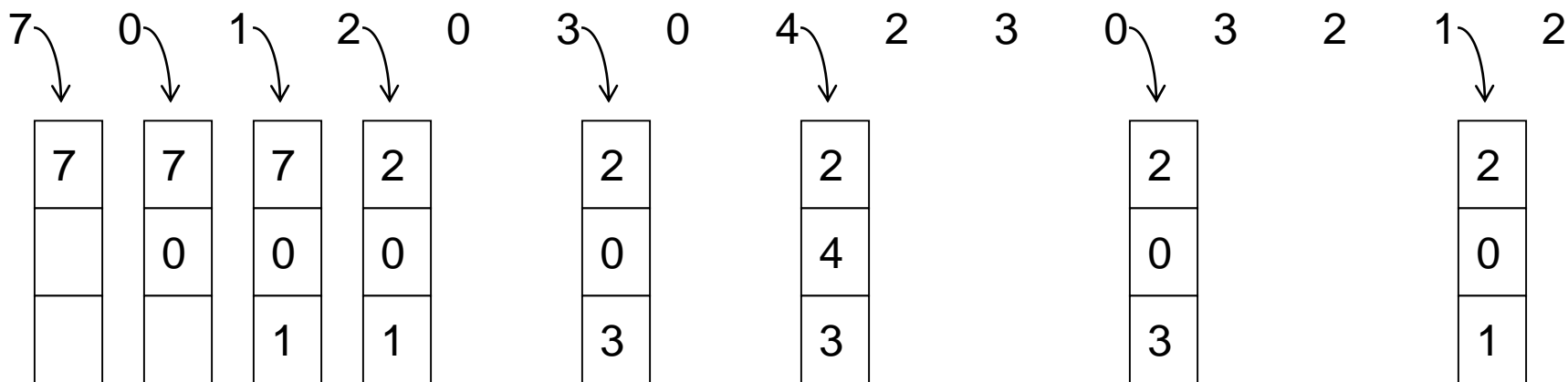
# Βέλτιστη Αντικατάσταση Σελίδων



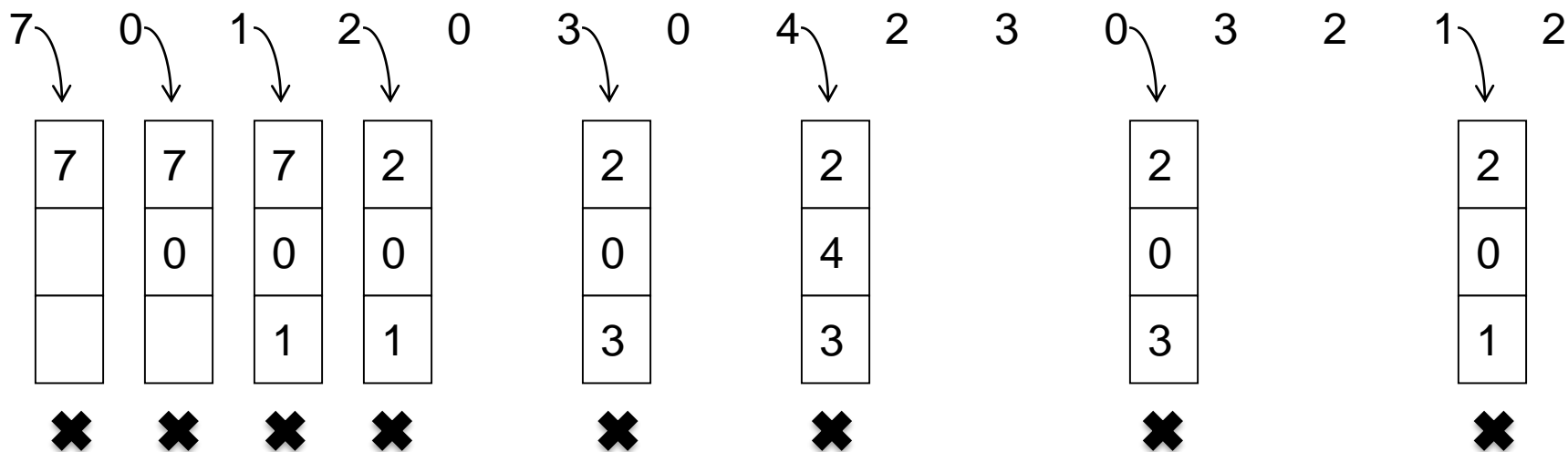
# Βέλτιστη Αντικατάσταση Σελίδων



# Βέλτιστη Αντικατάσταση Σελίδων



# Βέλτιστη Αντικατάσταση Σελίδων



8 σφάλματα σελίδας

# Γενικές και Τοπικές Πολιτικές Αντικατάστασης

- **Γενική αντικατάσταση** (global replacement): Μια διεργασία επιλέγει ένα πλαίσιο για αντικατάσταση από τη λίστα όλων των πλαισίων
  - Μια διεργασία μπορεί να πάρει ένα πλαίσιο από μια άλλη διεργασία
- **Τοπική αντικατάσταση** (local replacement): Κάθε διεργασία επιλέγει από το σύνολο των πλαισίων που έχουν ανατεθεί σε αυτή
  - Η εκτέλεση μιας διεργασίας δεν επηρεάζει την κατάσταση των σελίδων/πλαισίων των υπολοίπων

# Πτώση της Απόδοσης (Thrashing)

- Αν σε μια διεργασία δεν έχουν εκχωρηθεί αρκετά πλαίσια, αυτό μπορεί να οδηγήσει σε μεγάλο αριθμό σφαλμάτων σελίδας
- Η ΚΜΕ σπαταλάει το χρόνο της διεργασίας μεταφέροντας σελίδες από/προς το μέσο αποθήκευσης (thrashing)
  - Μη αποδοτική χρήση της ΚΜΕ
  - Πτώση της απόδοσης της διεργασίας (π.χ. καθυστέρηση της ολοκλήρωσής της)

# Καταπολέμηση Thrashing

- Στο σημείο που ξεκινάει η εμφάνιση του φαινομένου του thrashing, θα πρέπει να μειωθεί ο βαθμός πολυπρογραμματισμού
- Το φαινόμενο του thrashing μπορεί να περιοριστεί με χρήση τοπικών πολιτικών τοποθέτησης
- Για να αποτραπεί το thrashing πρέπει η κάθε διεργασία να καταλαμβάνει όσα πλαίσια χρειάζεται



# Καταπολέμηση Thrashing

- Το φαινόμενο του thrashing μπορεί να καταπολεμηθεί με την παρακολούθηση του ρυθμού σφαλμάτων σελίδας των διεργασιών
- Όταν είναι μεγάλος, γνωρίζουμε ότι η διεργασία χρειάζεται περισσότερα πλαίσια, ενώ όταν είναι μικρός πιθανόν να έχει παραπάνω πλαίσια τα οποία θα μπορούσαν να εκχωρηθούν σε άλλες διεργασίες
- Μπορούν να θεσπιστούν άνω και κάτω όρια στον επιθυμητό ρυθμό σφαλμάτων σελίδας

Ερωτήσεις