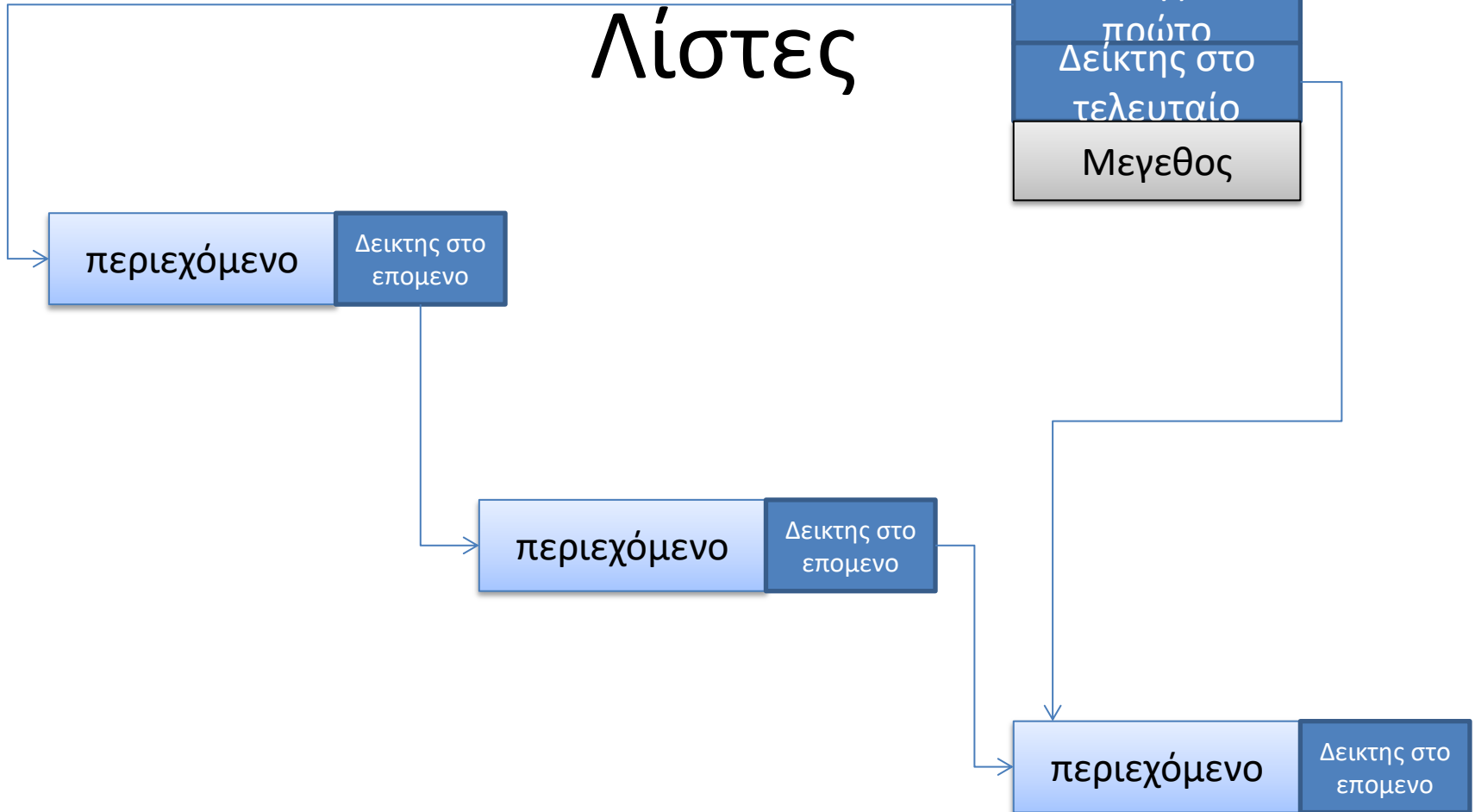


Εργαστήριο 7

- Δομές δεδομένων (Λίστες)

Λίστες



Create list

Επιστρέ-
φουμε
pointer
στην
αρχή
του
κομματι-
ού



Δείκτης στο πρώτο
Δείκτης στο τελευταίο
Μεγεθος

NULL

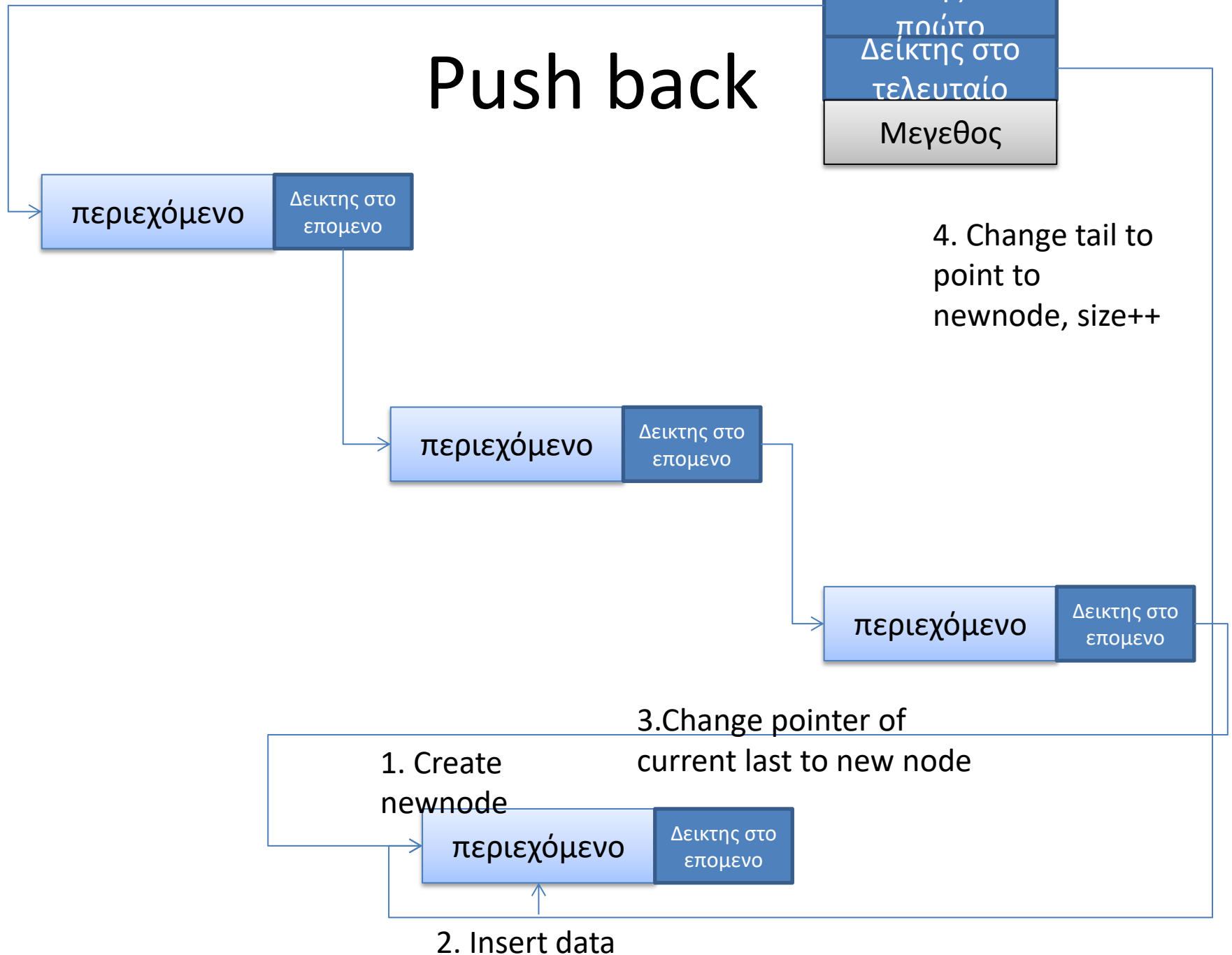
NULL

0

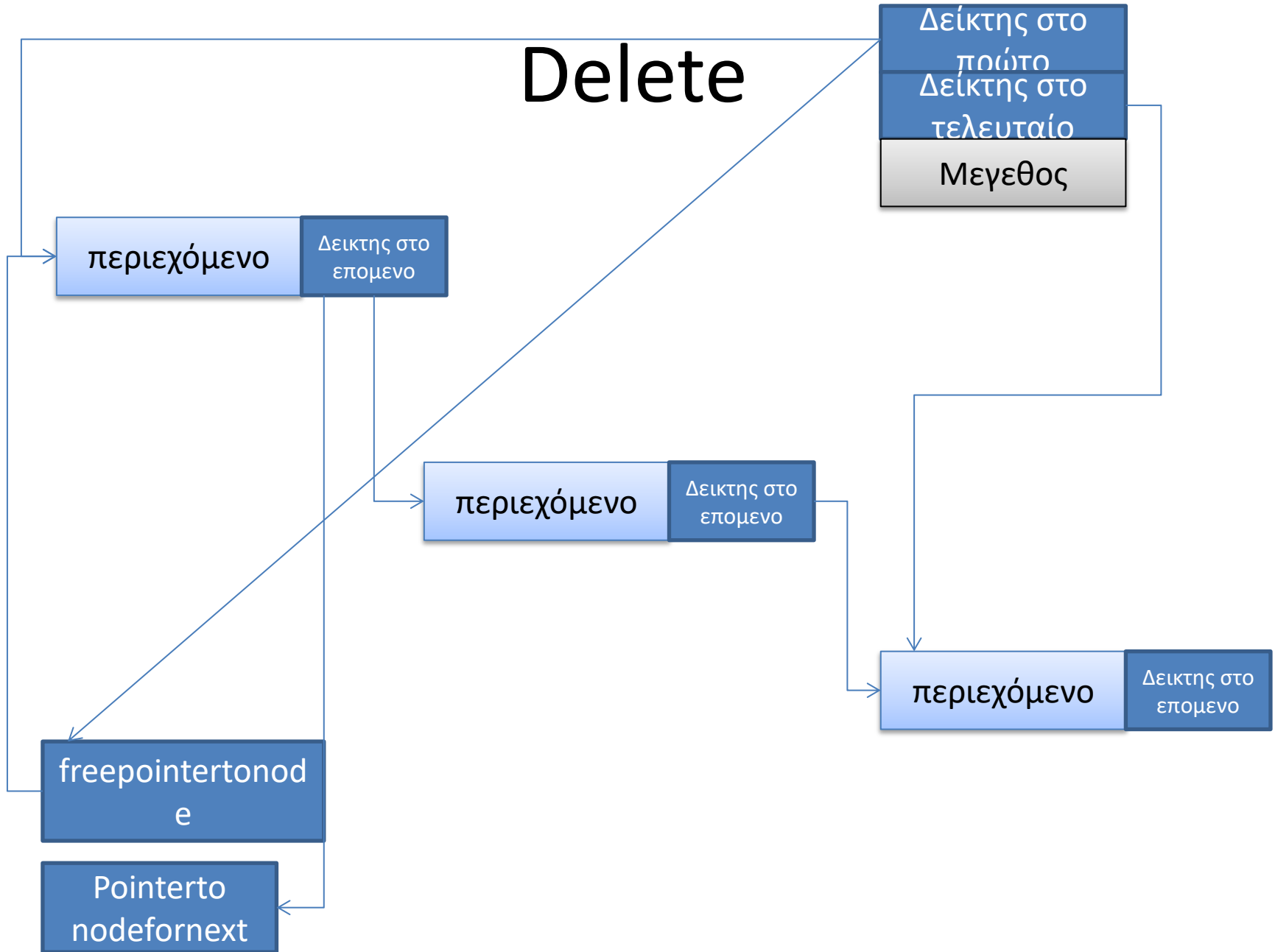
Push back



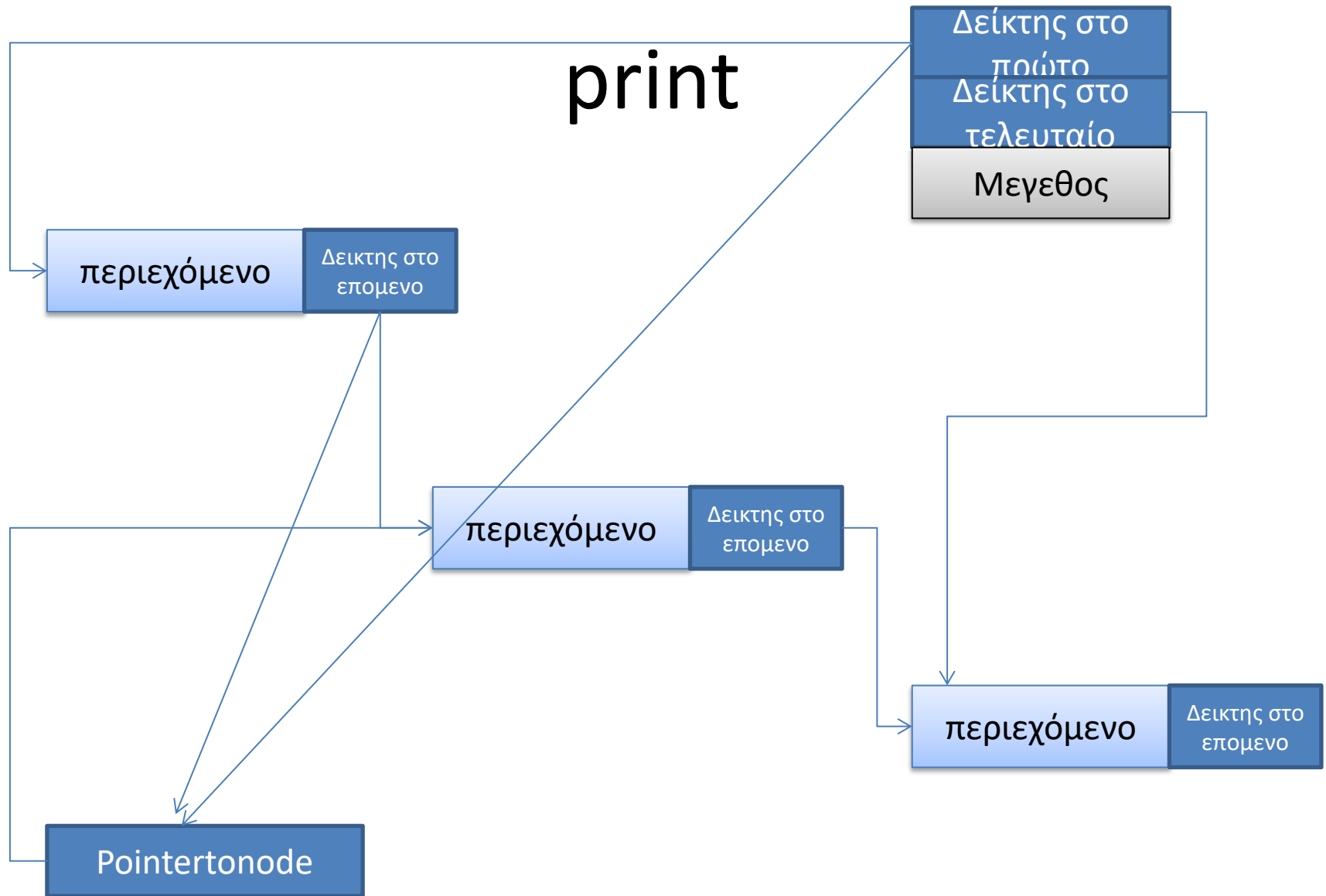
4. Change tail to
point to
newnode, size++



Delete



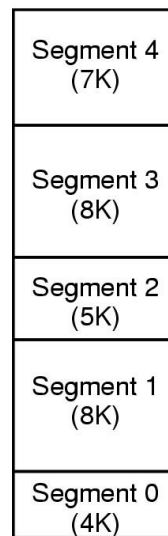
Δομή περιγραφής λίστας



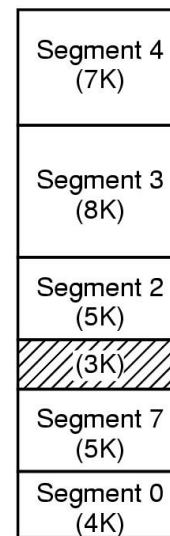
```
While (tmp!=NULL)  
    Tmp->name  
    Tmp=tmp->next
```

Εργαστήριο 7

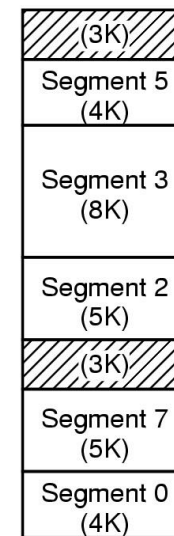
- Θυμηθείτε το πρόβλημα της τμηματοποίησης στα ΛΣ για την διαχείριση της μνήμης
- Κάπως πρέπει να κρατάει το ΛΣ ποια είναι τα άδεια και τα γεμάτα κομμάτια
- Αυτή η πληροφορία προφανώς μεταβάλλεται δυναμικά



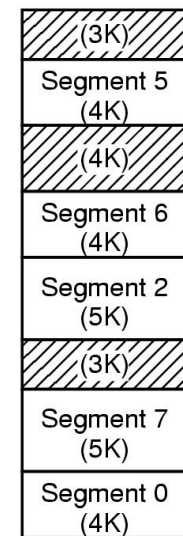
(a)



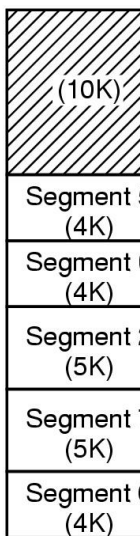
(b)



(c)



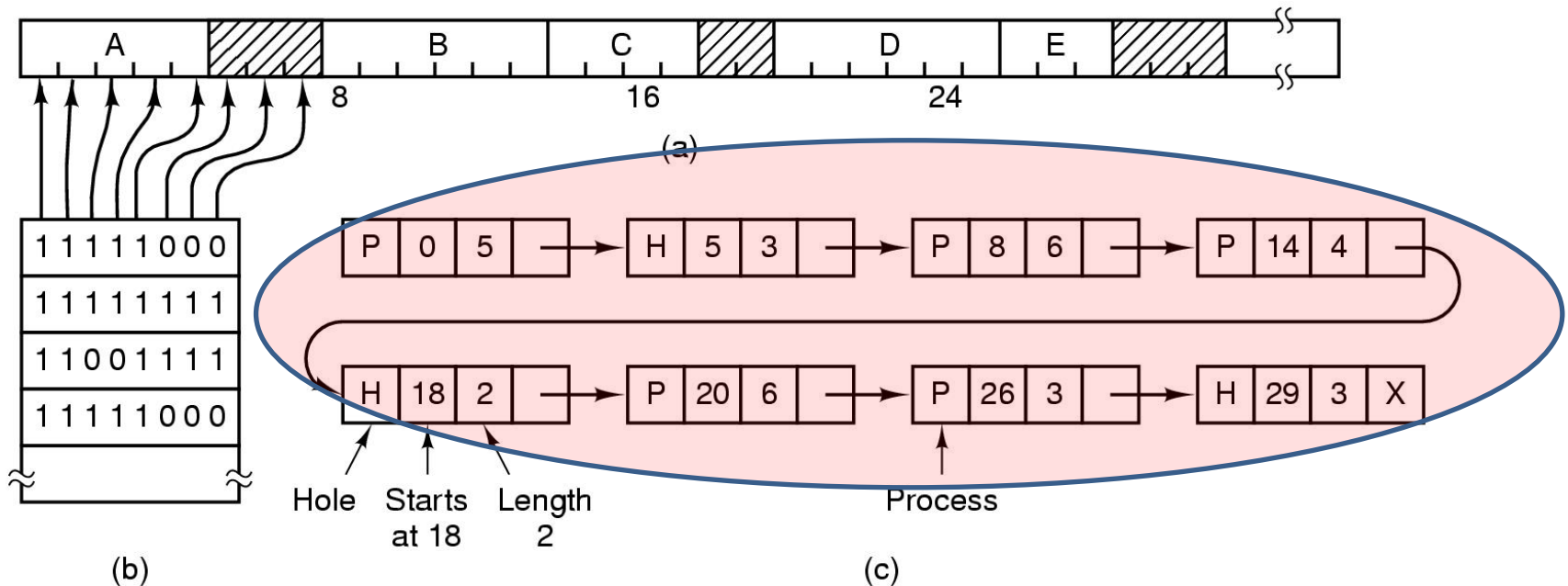
(d)



(e)

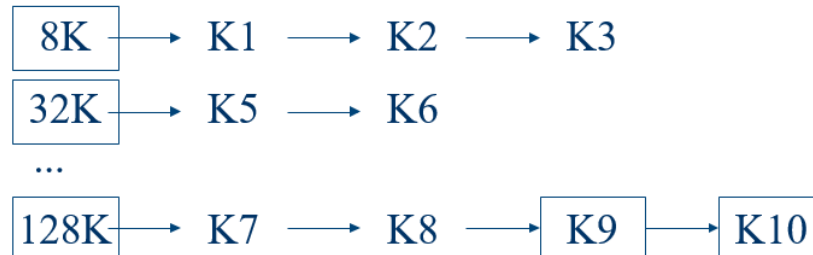
Εργαστήριο 7

- Λίστες που δείχνουν σε κενά τμήματα



Εργαστήριο 7 extensions

- Συνδυασμός πίνακα με λίστες
- Σε πολλές περιπτώσεις για να μη χρειάζεται να ψάχνετε σε όλα τα περιεχόμενα μπορείτε αν έχει νόημα να χωρίσετε τις λίστες σε «υπολίστες» με βάση κάποιο στατικό διαχωρισμό
- Για παράδειγμα στη περίπτωση του memory fragmentation να έχουμε μία λίστα για κάθε τύπο μεγέθους (ή κάποιο πολλαπλάσιο αυτού)



Εργαστήριο 7 extensions

- Εύρεση σημείου μέσα στη λίστα
 - Με βάση το περιεχόμενο
 - Με βάση τη θέση
 - Αν και δεν είναι πολύ χρήσιμο
- Συνδυασμός πίνακα με λίστες
- Υλοποίηση με συνάρτηση σαν argument
 - <https://medium.com/@jraleman/c-programming-language-passing-a-function-as-a-parameter-90d52fe842ea>
 - Αυτό χρειάζεται για να απλοποιήσετε την υλοποίησή σας
 - Πχ να έχετε μια γενική συνάρτηση που διαπερνάει τα στοιχεία της λίστας και να κάνει κάποια εργασία που καθορίζεται από τη συνάρτηση argument
 - Match
 - Free
 - κλπ