

# Συναρτήσεις

(Κλήσεις και επιστροφές από συναρτήσεις)

<https://mixstef.github.io/courses/comparch/>

Μ.Στεφανιδάκης

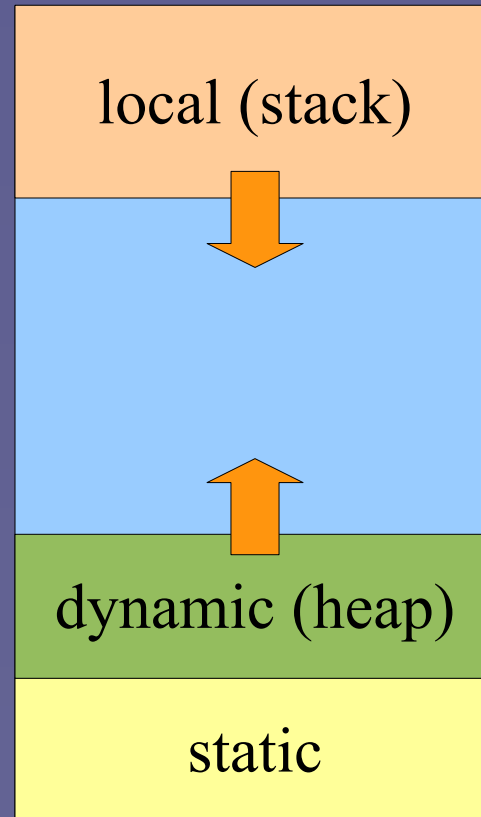


# Χώροι διευθύνσεων

- Στις δομημένες γλώσσες προγραμματισμού
- Δεδομένα
  - Στατικές μεταβλητές (static)
  - Τοπικές μεταβλητές (local)
  - Heap (δέσμευση με malloc)
- Εκτελέσιμος κώδικας προγράμματος
  - Εντολές μηχανής
    - Παραδοσιακά το τμήμα αυτό ονομάζεται “text”

# Τυπική οργάνωση χώρου δεδομένων

υψηλότερη διεύθυνση μνήμης προγράμματος



χαμηλότερη διεύθυνση μνήμης προγράμματος

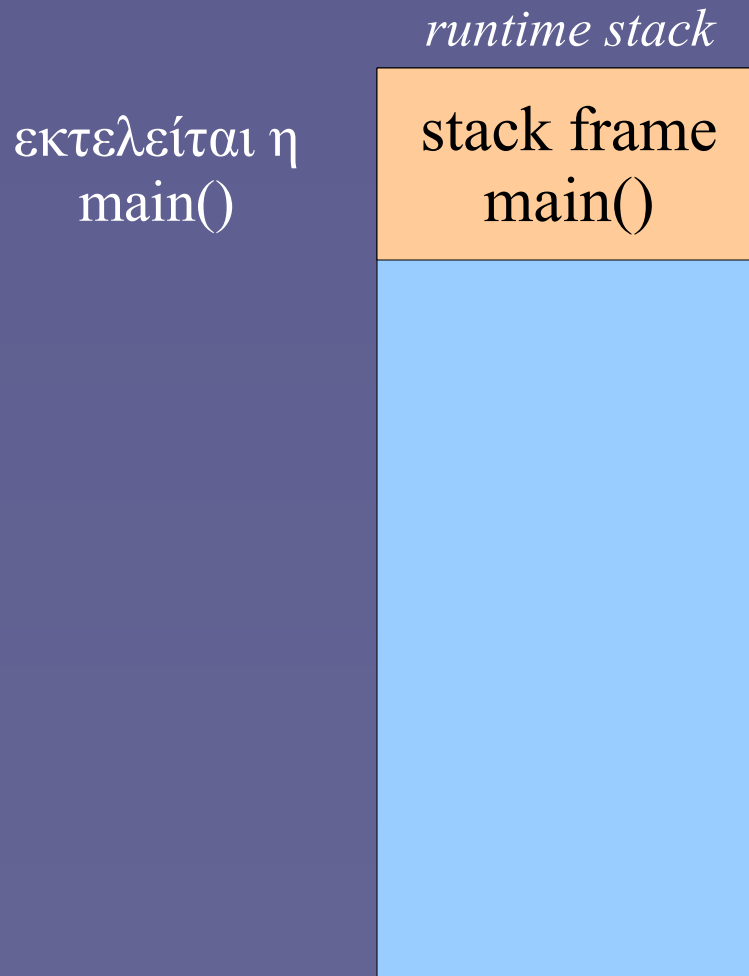
# (Runtime) stack

- Ο χώρος στη μνήμη για την αποθήκευση
  - Των τοπικών μεταβλητών των συναρτήσεων
  - Της διεύθυνσης επιστροφής
  - Των παραμέτρων κλήσης της συνάρτησης
  - Των επιστρεφόμενων αποτελεσμάτων
  - Στις σύγχρονες αρχιτεκτονικές πολλά από τα παραπάνω παραμένουν όσο το δυνατόν στους καταχωρητές
- Stack pointer (sp)
  - Καταχωρητής γενικού ή ειδικού σκοπού (ανάλογα με την αρχιτεκτονική ISA) που περιέχει τη διεύθυνση της κορυφής στη στοίβα
    - Όπου θα γραφούν δεδομένα (push) ή θα αναγνωστούν (pop) δεδομένα
    - Η τιμή του προσαρμόζεται κατάλληλα μετά από κάθε push/pop

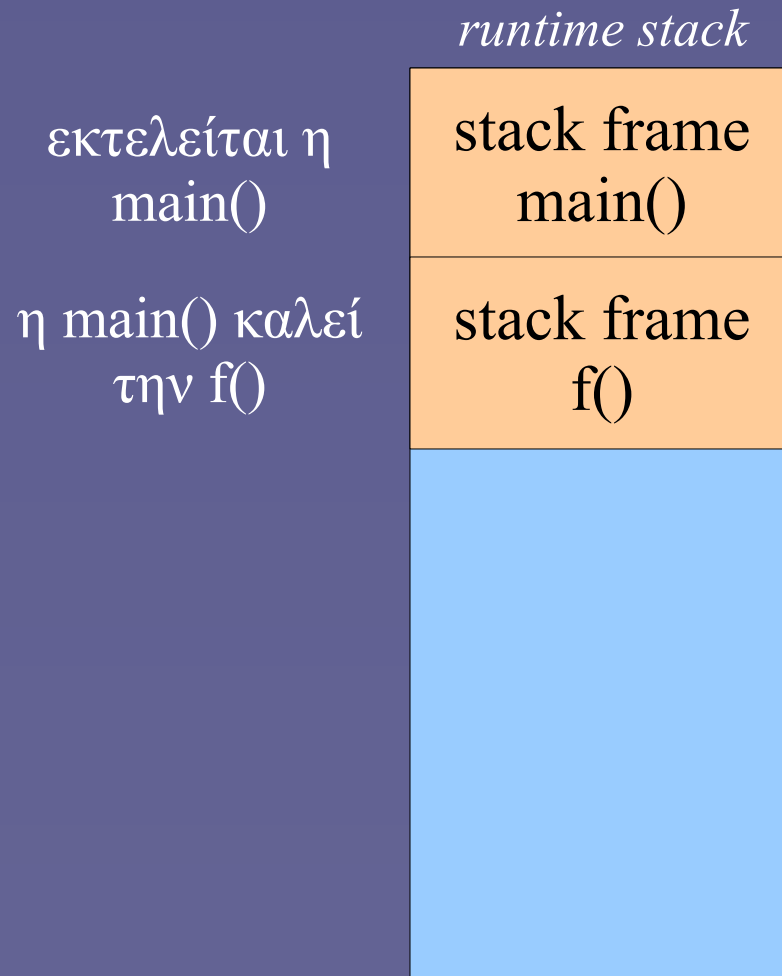
# Stack Frame (ή Activation Record)

- Περιβάλλον εκτέλεσης **μιας κλήσης** συνάρτησης
  - **Χώρος που δεσμεύεται στο runtime stack**
  - Περιέχει το «περιβάλλον εκτέλεσης» μιας συνάρτησης
    - Τοπικές μεταβλητές
    - Παράμετροι εισόδου
    - Διεύθυνση επιστροφής
  - Κάθε κλήση συνάρτησης δημιουργεί ένα **καινούργιο** stack frame (αμέσως μετά το τρέχον stack frame)
- **Frame pointer (fp)**
  - Καταχωρητής (συνήθως γενικού σκοπού) που περιέχει τη διεύθυνση της αρχής του stack frame
  - Όλες οι προσπελάσεις σε τοπικές μεταβλητές γίνονται σε σχέση ( $\pm$ ) με την τρέχουσα τιμή του fp

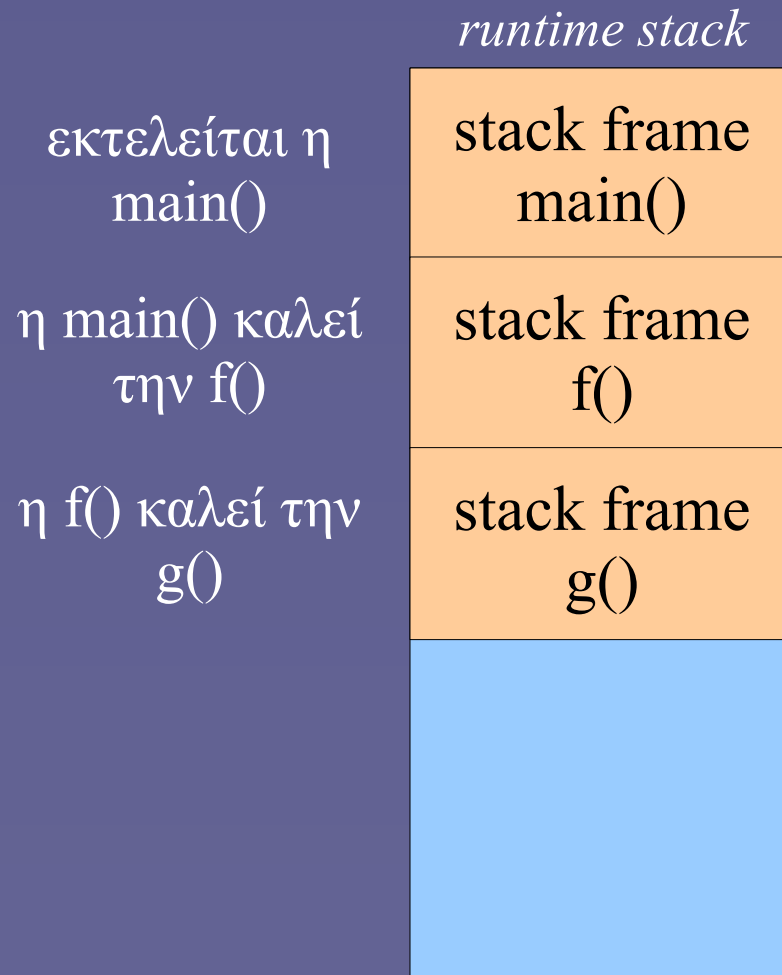
# Stack frames (Activation records)



# Stack frames (Activation records)

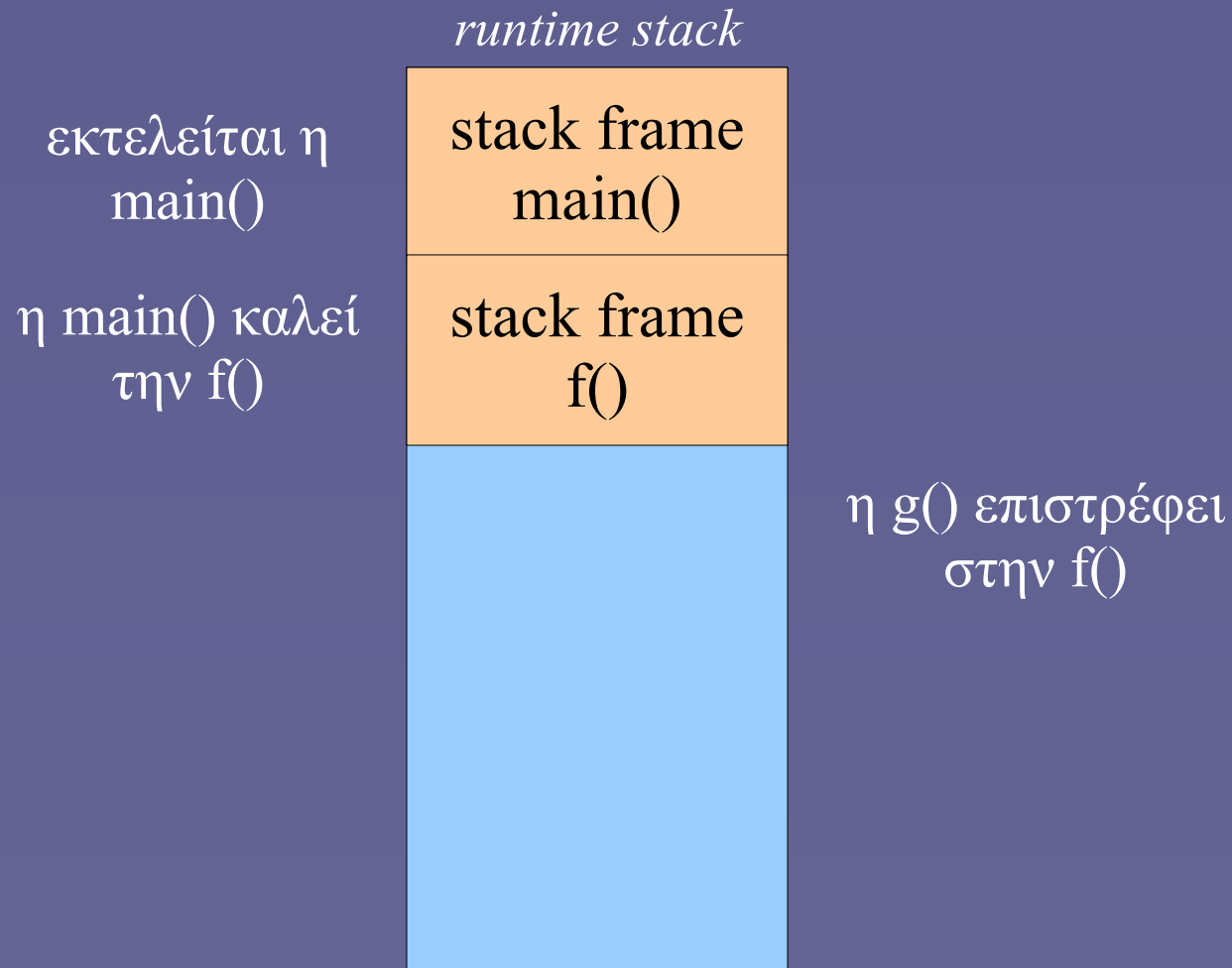


# Stack frames (Activation records)

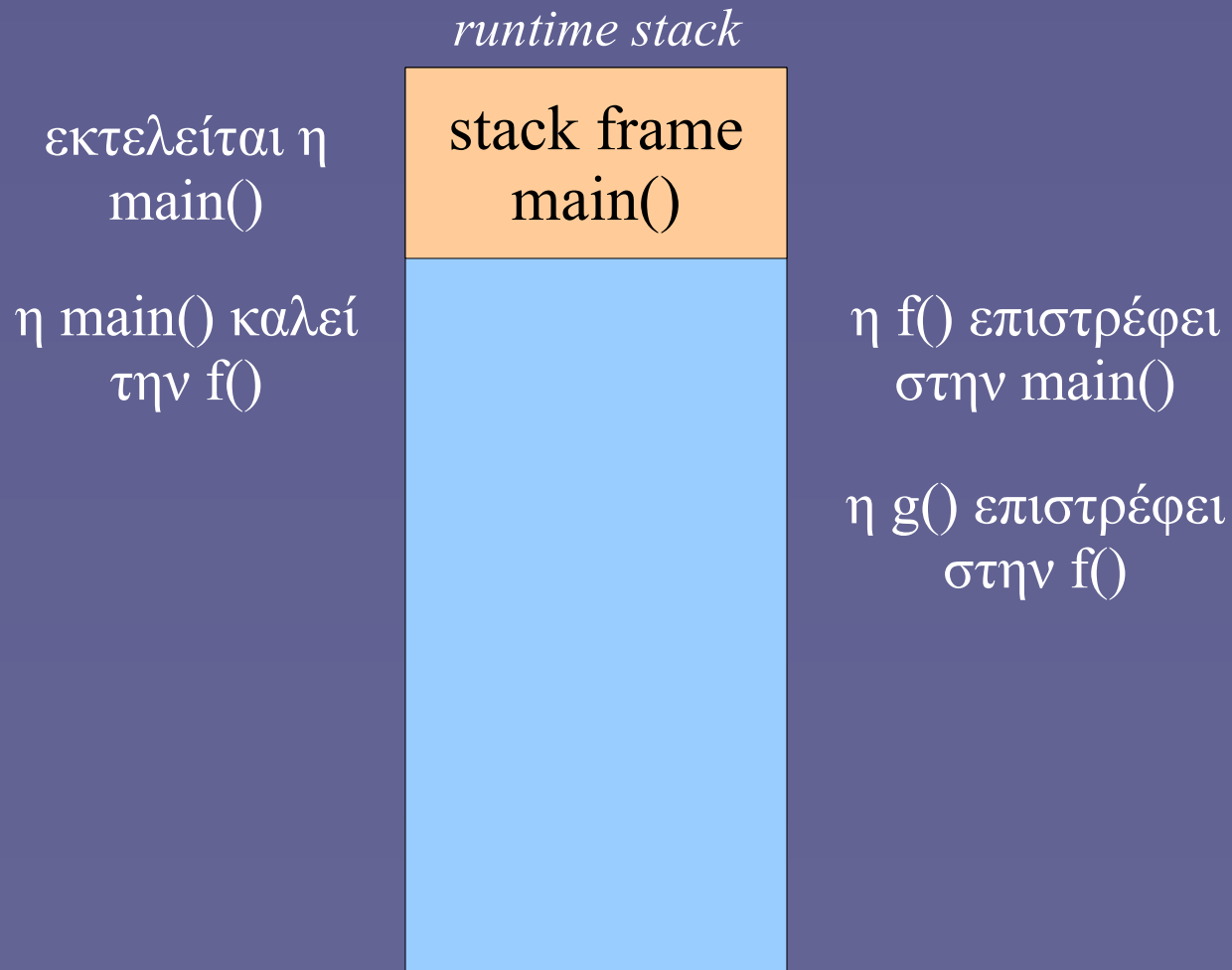




# Stack frames (Activation records)



# Stack frames (Activation records)



# Διαδικασία κλήσης συνάρτησης (1)

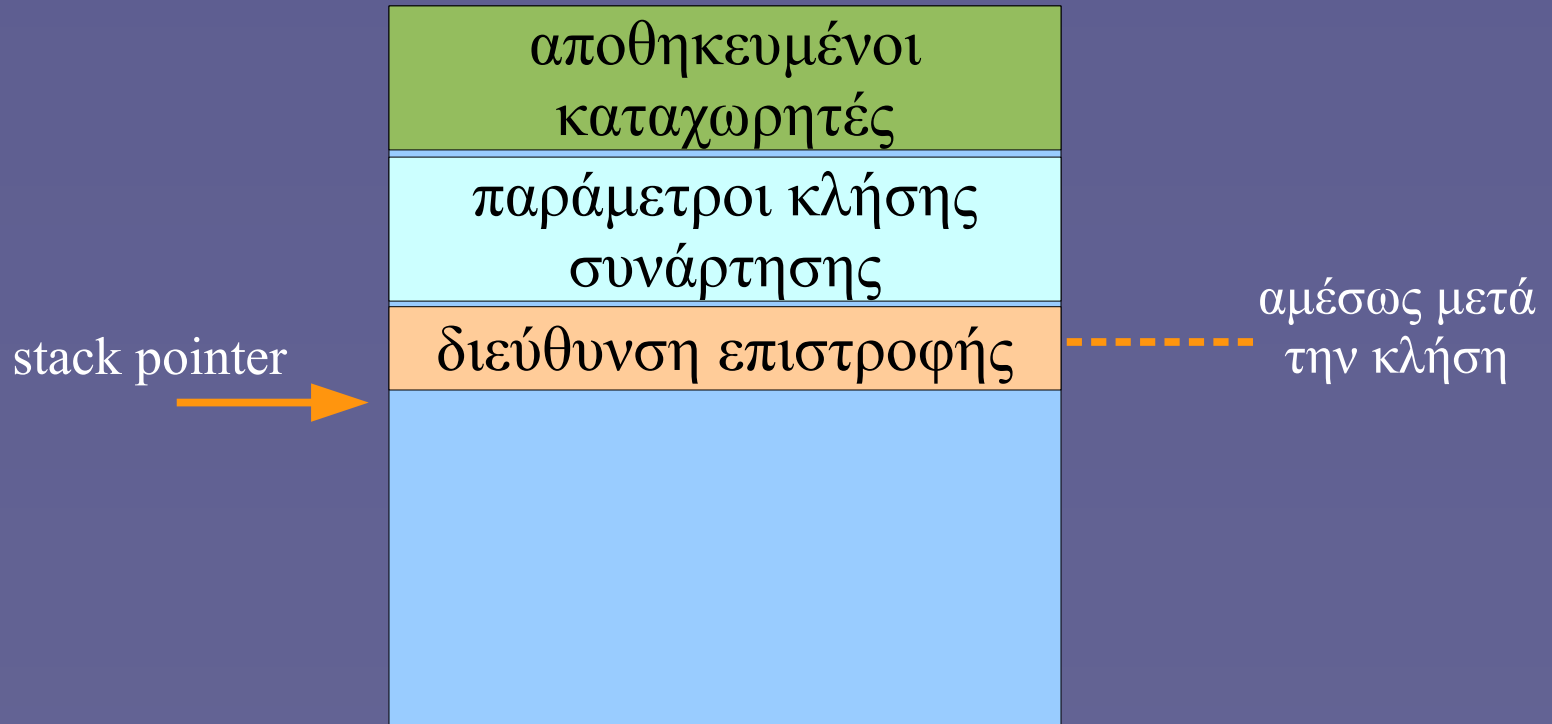
- Η καλούσα συνάρτηση
  - Αποθηκεύει καταχωρητές που πιθανόν να αλλάξει η καλούμενη συνάρτηση
  - Περνάει τις παραμέτρους κλήσης στο stack frame (ή σε καταχωρητές)
  - Χρησιμοποιεί εντολή μηχανής τύπου “call”
    - Διακλάδωση στην καλούμενη συνάρτηση
    - Με ταυτόχρονη αποθήκευση της διεύθυνσης επιστροφής (stack frame ή καταχωρητές)
- Μετά την επιστροφή
  - Αποκατάσταση τιμών καταχωρητών, αποδέσμευση χώρου stack

# Διαδικασία κλήσης συνάρτησης (2)

- Η καλούμενη συνάρτηση
  - Δημιουργεί χώρο για τις τοπικές μεταβλητές στο stack
  - Υπολογίζει τη νέα τιμή του fp και αποθηκεύει την παλιά
  - Αποθηκεύει καταχωρητές που πιθανόν να αλλάξει
- Στη συνέχεια εκτελείται ο κώδικας της συνάρτησης
- Πριν το τέλος
  - Αποδέσμευση χώρου τοπικών μεταβλητών, αποκατάσταση τιμών καταχωρητών, fp
  - Χρήση εντολής μηχανής τύπου “return”
    - Διακλάδωση στην αποθηκευμένη διεύθυνση επιστροφής

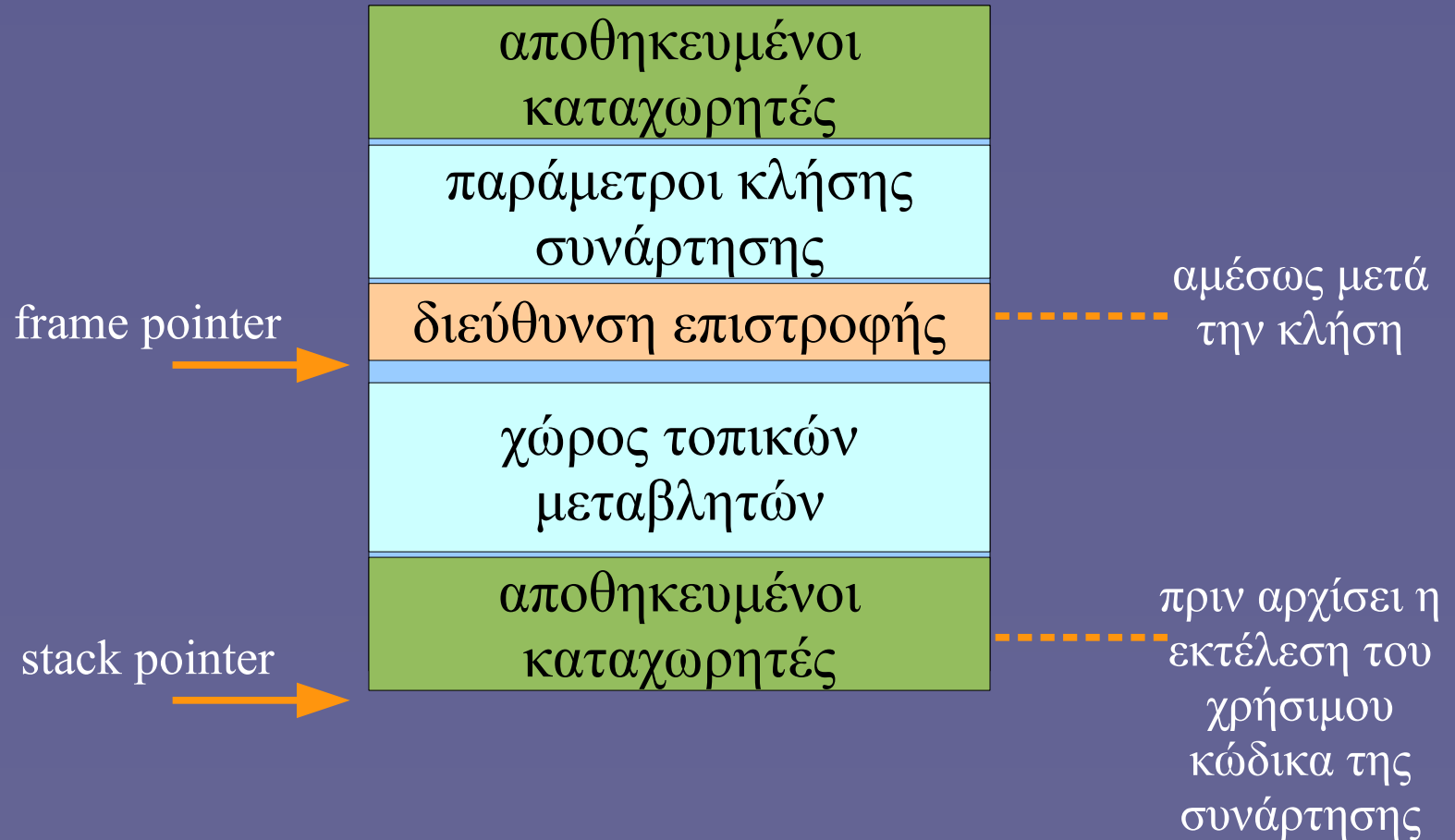
# Παράδειγμα stack frame

(η συνάρτηση μόλις έχει κληθεί)



# Παράδειγμα stack frame

(κατά την έναρξη εκτέλεσης του χρήσιμου κώδικα της συνάρτησης)



# “Calling convention”

- Ο ακριβής τρόπος κλήσης των συναρτήσεων
  - Ποιες εντολές και ποιοι καταχωρητές χρησιμοποιούνται
  - Η ακολουθία ενεργειών πριν και μετά την κλήση/επιστροφή
  - Πώς είναι η ακριβής μορφή του stack frame
- Εξαρτάται από την αρχιτεκτονική (ISA) και το λειτουργικό σύστημα
  - Application Binary Interface (ABI)

# Συναρτήσεις και RISC-V ISA

- Δεν υπάρχουν ειδικοί καταχωρητές για τη στοίβα
- Συγκεκριμένοι καταχωρητές γενικού σκοπού χρησιμοποιούνται «κατά σύμβαση» κατά την κλήση των συναρτήσεων
  - $x1 \rightarrow ra$  (return address)
  - $x2 \rightarrow sp$  (stack pointer)
  - $x8 \rightarrow fp$  (ή  $s0$ ) (frame pointer)
  - $x10-x17 \rightarrow a0-a7$  (argument registers, ορίσματα και επιστρεφόμενη τιμή)



# Συναρτήσεις και RISC-V ISA

- «Κατά σύμβαση» η καλούμενη συνάρτηση πρέπει να διατηρήσει την τιμή ορισμένων καταχωρητών
  - Να τους αποθηκεύσει και να τους αποκαταστήσει στην αρχική τιμή πριν επιστρέψει
  - `sp`, `fp` και ορισμένοι άλλοι καταχωρητές
  - Ο κώδικας που εκτελεί τα παραπάνω ονομάζεται «πρόλογος» και «επίλογος» της συνάρτησης
    - Περικλείει τον κώδικα του χρήστη στη συνάρτηση

# Εντολές κλήσης και επιστροφής

- Κλήση συνάρτησης με εντολή jal (jump and link)
  - `jal ra, funcaddr` (ψευδοεντολή call funcaddr)
    - Λειτουργία: `ra`  $\leftarrow$  `pc` + 4 (next instruction), `pc`  $\leftarrow$  `pc`  $\pm$  offset to `funcaddr`
- Επιστροφή με εντολή jalr (jump and link register)
  - `jalr x0, ra, 0` (ψευδοεντολή jr ra ή ret)
    - Λειτουργία: `x0`  $\leftarrow$  `pc` + 4 (next instruction), `pc`  $\leftarrow$  `ra`  $\pm$  0

# Διαχείριση στοίβας

- Η στοίβα «μεγαλώνει» προς χαμηλότερες διευθύνσεις
  - π.χ. `addi sp, sp, -32` (δέσμευση χώρου για stack frame)
  - και `addi sp, sp, 32` (αποδέσμευση χώρου stack frame πριν την επιστροφή)
- Η προσπέλαση των τοπικών μεταβλητών της συνάρτησης γίνεται με τη βοήθεια του fp (ή αλλιώς, s0)
  - π.χ. `sw a1, -20(s0)`
    - Λειτουργία: `a1`  $\rightarrow$  `mem[s0 - 20]`