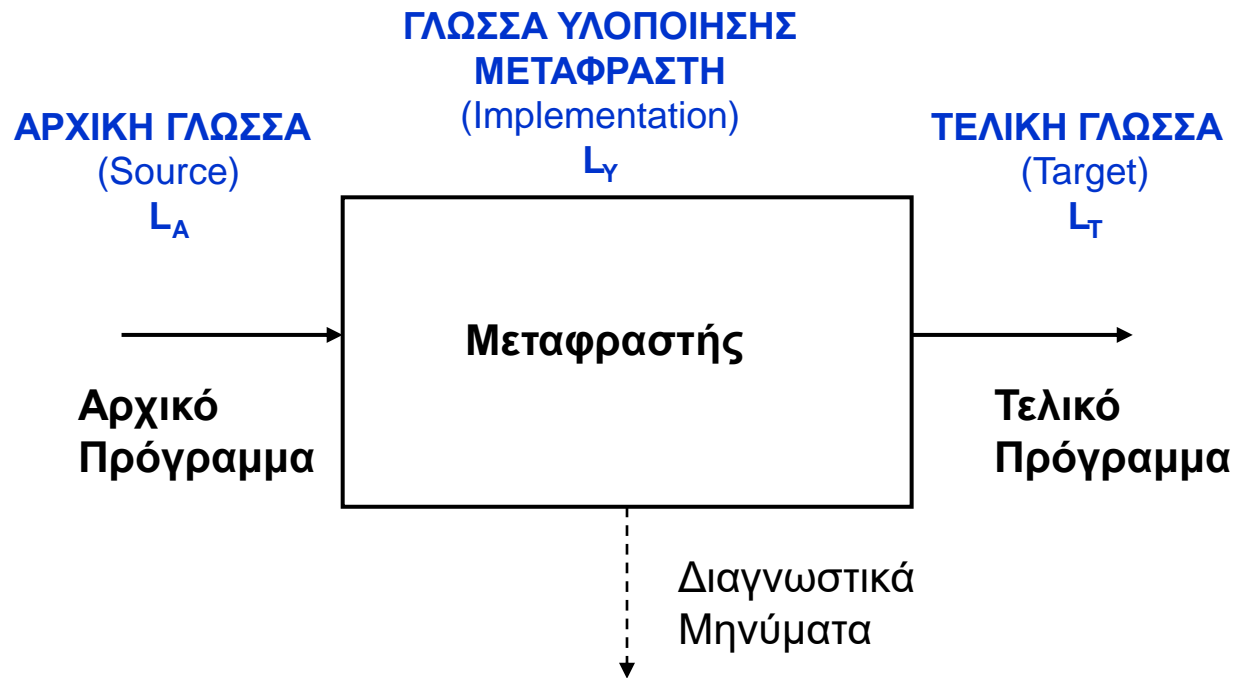


# Κεφάλαιο 2: Μεταφραστές

*Αρχές Γλωσσών Προγραμματισμού και Μεταφραστών*

Γιάννης Γαροφαλάκης, Σπύρος Σιούτας

# Λειτουργία Μετάφρασης



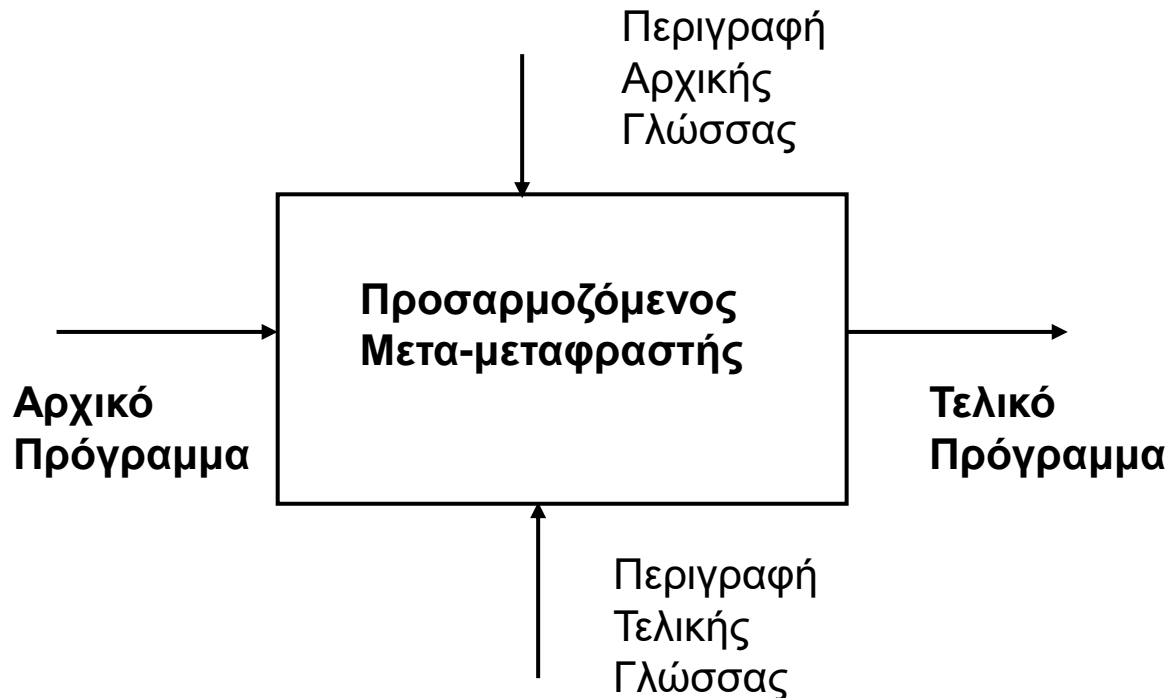
Συμβολισμός:  $C_Y^{A,T}$

# Είδη Μεταφραστών (1)

- Απλοί
- Αντίστροφοι (decompilers)
  - Από Γλώσσα Μηχανής σε Γλώσσα Υψηλού Επιπέδου
  - Χρήσιμοι για κατανόηση και debugging προγραμμάτων σε Γλώσσα Μηχανής
  - Π.χ. **JD (Java Decompiler)**
- Μετα-Μεταφραστές (meta-compilers)
  - Χρήση: Υλοποίηση μεταφραστών
  - Παραμετρικοί ως προς αρχική και τελική γλώσσα
  - Χρησιμοποιούν *μετα-γλώσσα* για την περιγραφή τους
  - Χρησιμότητα: Λιγότερη δουλειά για την κατασκευή του μεταφραστή

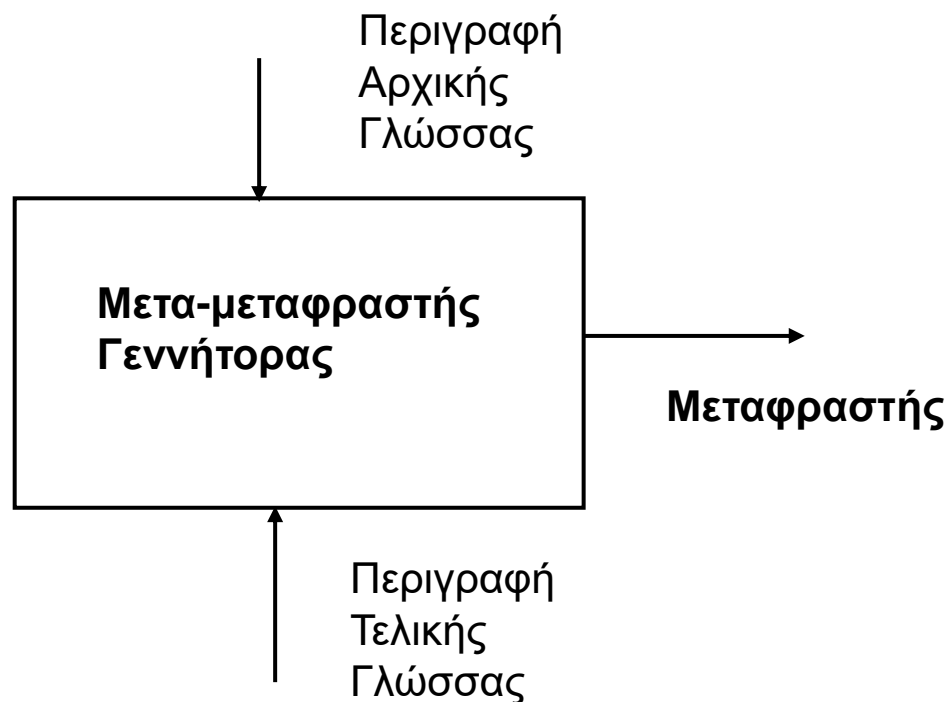
# Είδη Μεταφραστών (2)

- Δύο κατηγορίες μετα-μεταφραστών:
  - **Προσαρμοζόμενοι** - λειτουργούν ως ένας μεταφραστής για το ζεύγος «αρχική γλώσσα» - «τελική γλώσσα» (adjustable meta-compilers)



# Είδη Μεταφραστών (2)

- **Μετα-μεταφραστές γεννήτορες** - παράγουν το Μεταφραστή (meta-compiler generators)



# Ειδικές περιπτώσεις προγραμμάτων που θεωρούνται Μεταφραστές (1)

## ■ Προεπεξεργαστές (preprocessors)

- Αρχική και Τελική είναι Γλώσσες Υψηλού Επιπέδου (ΓΥΕ)
- Υποστήριξη πρόσθετων χαρακτηριστικών της ΓΠ
- Αφαιρεί σχόλια και κενά, Ομαδοποιεί χαρακτήρες σε *tokens*
- Π.χ. προεπεξεργαστές της C, της PHP

## ■ Συμβολομεταφραστές (assemblers)

- Από Συμβολική Γλώσσα (assembly) σε Γλώσσα Μηχανής
- Διευκολύνει τη διόρθωση (η assembly είναι εύκολα αναγνώσιμη)
- Απομονώνει το μεταφραστή από αλλαγές στη μορφή των αρχείων της γλώσσας μηχανής (μόνο ο assembler πρέπει να αλλάξει)

# Ειδικές περιπτώσεις προγραμμάτων που θεωρούνται Μεταφραστές (2)

- **Γεννήτορες Προγραμμάτων (program generators)**
  - Για την αυτόματη κατασκευή προγραμμάτων / τμημάτων μεταφραστή
  - Π.χ. **flex**, **bison** (γραμμένοι σε C), για κατασκευή λεξικών και συντακτικών αναλυτών γραμμένων σε C, C++ ή Java
  - *Είσοδος*: Προδιαγραφές προγράμματος σε Αρχική Γλώσσα (συνήθως όχι ΓΠ, π.χ. αλγοριθμική περιγραφή)
  - *Έξοδος*: Πρόγραμμα σε Τελική Γλώσσα (Υψηλού Επιπέδου)

# Χρήσιμα εργαλεία στην ανάπτυξη λογισμικού (1)

- **Βιβλιοθήκη χρόνου εκτέλεσης (run-time library)**
  - Π.χ. της C περιέχει τον εκτελέσιμο κώδικα για τις προκαθορισμένες συναρτήσεις (printf, ...)
- **Συνδέτης (linker)**
  - Συνδέει 1 ή περισσότερα μεταφρασμένα προγράμματα και run-time libraries για να δώσει εκτελέσιμο κώδικα
- **Φορτωτής (loader)**
  - Τοποθετεί τον εκτελέσιμο κώδικα στη μνήμη για να εκτελεστεί
  - Συνήθως είναι τμήμα του Λειτουργικού Συστήματος (ΛΣ)



# Χρήσιμα εργαλεία στην ανάπτυξη λογισμικού (2)

## ■ Εκδότες Προγραμμάτων (program editors)

- Καταγραφή και διόρθωση προγραμμάτων

## ■ Εντοπιστές σφαλμάτων (debuggers)

- Βηματική εκτέλεση προγράμματος
- Παρακολούθηση τιμών μεταβλητών
- Τοποθέτηση σημείων διακοπής εκτέλεσης

## ■ Στατιστικοί αναλυτές (profilers)

- Μέτρηση κλήσεων/εκτέλεσης τμημάτων προγράμματος
- Μέτρηση χρόνου εκτέλεσης εντολών
- Ανακάλυψη τμημάτων που δεν εκτελέστηκαν

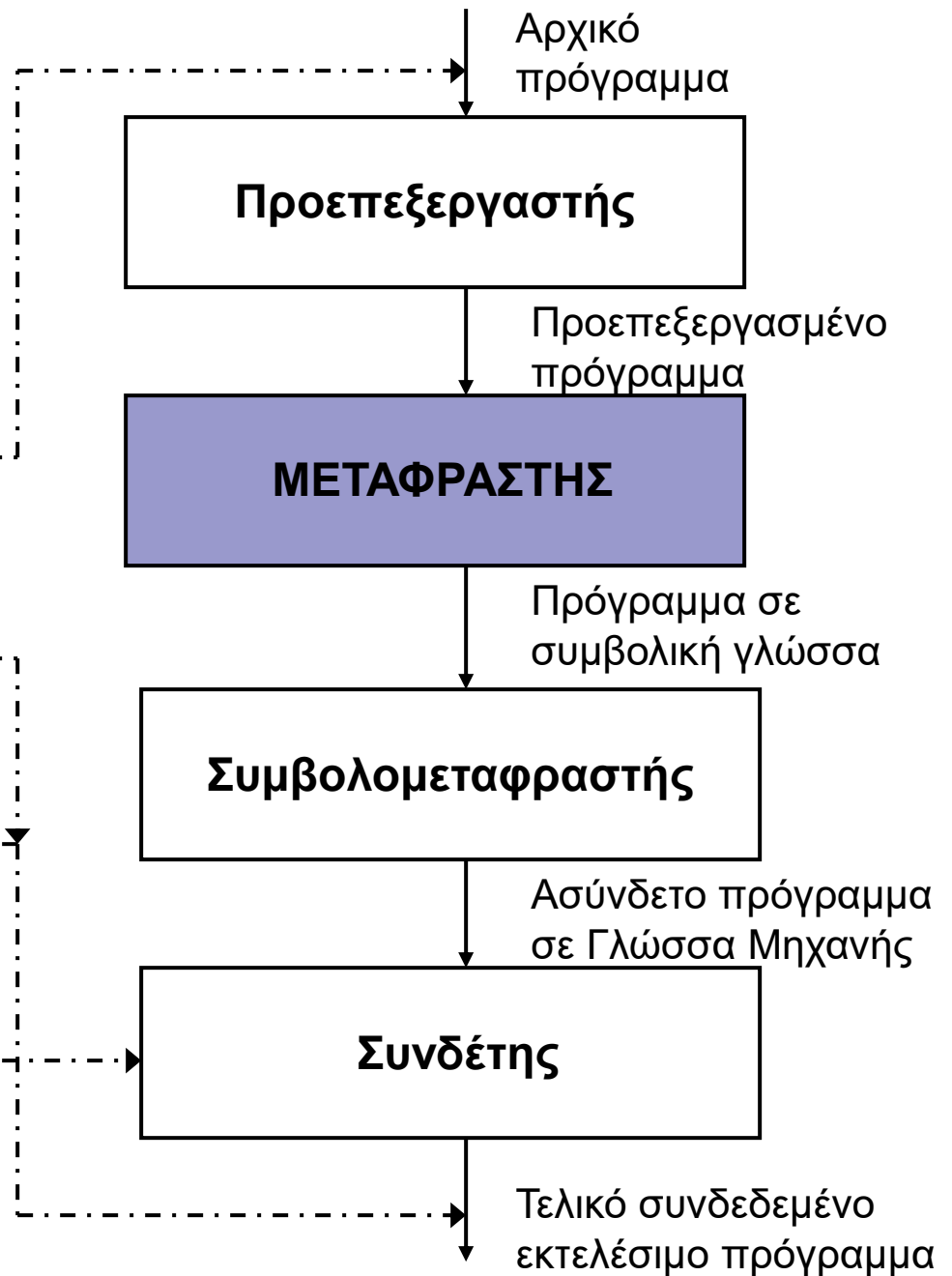
# Ολοκληρωμένο Περιβάλλον Προγραμματισμού

Εκδότης προγράμματος

Εντοπιστής σφαλμάτων

Στατιστικός αναλυτής

Βιβλιοθήκη χρόνου εκτέλεσης



# Κατασκευή Μεταφραστή

- Κυρίως χειρονακτική εργασία
- Σήμερα γίνεται υβριδικά: Ορισμένα τμήματα (λεξικός και συντακτικός αναλυτής) αυτόματα, τα υπόλοιπα χειρονακτικά

## Ορισμός προβλήματος

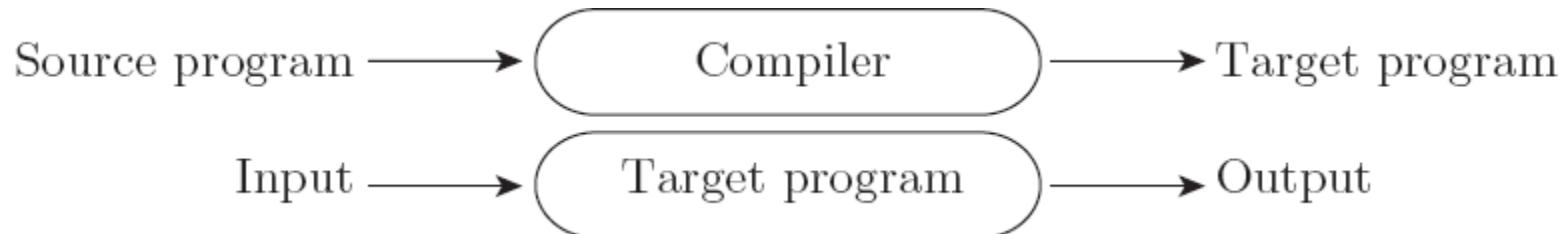
- **Σύνταξη** (λεξική και συντακτική ανάλυση)
  - Τυπικοί συμβολισμοί: Πεπερασμένα αυτόματα, BNF, Συντακτικά Διαγράμματα
- **Σημασιολογία**
  - Πολύ πιο δύσκολος ...

# Διερμηνείς (Interpreters)

- Μετάφραση βήμα – βήμα
- Μετάφραση στην Αρχική Γλώσσα ή σε ενδιάμεση
- ΠΛΕΟΝΕΚΤΗΜΑΤΑ:
  - Γραμμένοι σε ΓΥΕ → μεταφέρονται σε πολλές μηχανές
  - Πιο εύκολοι στην κατασκευή
  - Καλύτερος έλεγχος και αναφορά σφαλμάτων
  - Προσφέρουν αυξημένη ασφάλεια
  - Κατάλληλοι για εφαρμογές στο Internet
- ΜΕΙΟΝΕΚΤΗΜΑ:
  - Πολύ αργοί

# Μετάφραση vs Διερμηνεία (1)

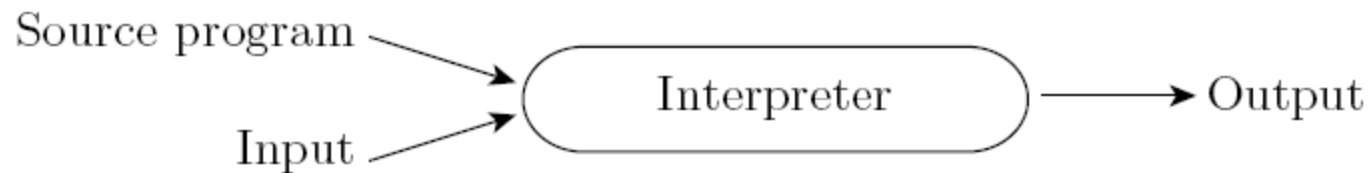
- Δεν είναι αντίθετα
- Δεν υπάρχει σαφής διάκριση
- Καθαρή μετάφραση:
  - Ο μεταφραστής μεταφράζει τον αρχικό κώδικα υψηλού επιπέδου σε ισοδύναμο τελικό πρόγραμμα (συνήθως σε γλώσσα μηχανής) και δεν χρησιμοποιείται πλέον:



# Μετάφραση vs Διερμηνεία (2)

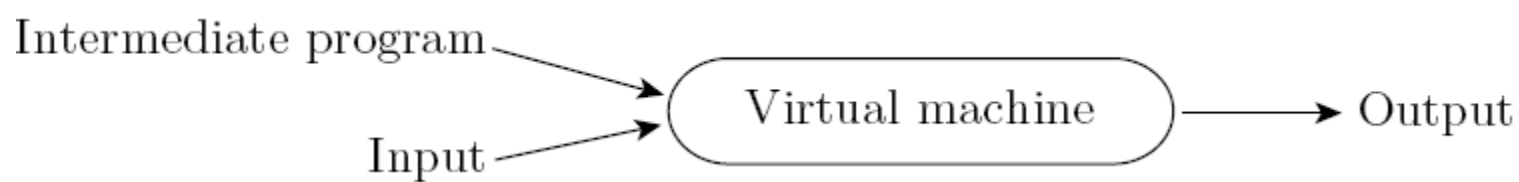
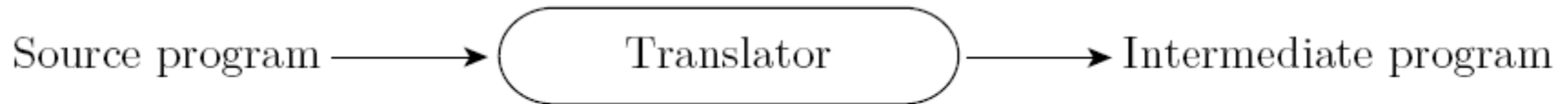
## ■ Καθαρή Διερμηνεία:

- Ο διερμηνέας σχετίζεται με την εκτέλεση του προγράμματος
- Είναι το κέντρο ελέγχου κατά τη διάρκεια της εκτέλεσης



# Μετάφραση vs Διερμηνεία (3)

- Μία κοινή πρακτική είναι αρχικά μετάφραση ή απλή προ-επεξεργασία, ακολουθούμενη από διερμηνεία.
- Οι περισσότερες υλοποιήσεις γλωσσών περιέχουν μια μίξη από μετάφραση και διερμηνεία.



# Μετάφραση vs Διερμηνεία (4)

- *Virtual Machine (VM)*: για εκτέλεση προγραμμάτων σε περιβάλλον ανεξάρτητο από Η/Υ πλατφόρμα. Λειτουργούν ως διερμηνείς.
- Π.χ. **Java Virtual Machine (JVM)**. Επιτρέπει στον Η/Υ να τρέχει προγράμματα **Java**, ή και άλλων ΓΠ που είναι interoperable με Java, π.χ. προγράμματα **Kotlin, Scala**.
- **Java bytecode** είναι το σύνολο εντολών της JVM.
- Ο προγραμματιστής Java (ή Kotlin ή Scala) δεν χρειάζεται να γνωρίζει την Java bytecode, όπως ο προγραμματιστής C δεν χρειάζεται να γνωρίζει assembly.



# Μετάφραση vs Διερμηνεία (5)

- Η μετάφραση ΔΕΝ χρειάζεται να παράγει γλώσσα μηχανής για κάποιο είδος hardware.
- Η μετάφραση είναι *μεταγλώττιση* από μια γλώσσα σε μια άλλη, με πλήρη ανάλυση της εισόδου
- Η μετάφραση συνεπάγεται σημασιολογική κατανόηση για ότι επεξεργάζεται. Η προ-επεξεργασία όχι
- Ένας προ-επεξεργαστής συχνά αφήνει λάθη να περάσουν.

# Μετάφραση vs Διερμηνεία (6)

- Πολλές μεταφρασμένες γλώσσες έχουν διερμηνευμένα κομμάτια, π.χ. τα formats σε Fortran ή C.
- Οι περισσότερες μεταφρασμένες γλώσσες χρησιμοποιούν «εικονικές εντολές»
  - Λειτουργίες συνόλων στην Pascal
  - Χειρισμός strings στην Basic
- Μερικοί μεταφραστές παράγουν μόνο εικονικές εντολές π.χ. Pascal P-code, Java bytecode, Microsoft COM+

# Μετάφραση vs Διερμηνεία (7)

- Source-to-source μετάφραση (C++)
  - Οι C++ υλοποιήσεις που βασίζονται στον πρώιμο AT&T μεταφραστή, δημιουργούν ένα ενδιάμεσο πρόγραμμα σε C, αντί ενός προγράμματος σε γλώσσα assembly.

