

Διάλεξη 1η: Εισαγωγή

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Εισαγωγή στην Επιστήμη Υπολογιστών



- Διδάσκων:
 - Πολύβιος Πρατικάκης: polyvios@csd.uoc.gr
- Ώρες γραφείου διδάσκοντος: Δευτέρα, 14:15-16:00, K-327.
- Σελίδα μαθήματος: <http://www.csd.uoc.gr/~hy100>



Καλώς ήρθατε στο CSD!

- Επιστήμη Υπολογιστών: Γεφυρώνει τα Μαθηματικά, την Επιστήμη, και την Τεχνολογία.
 - Η μελέτη των θεωρητικών θεμελίων της πληροφορίας και των υπολογισμών,
 - η τεχνολογία, σχεδίαση και οργάνωση των μηχανών και συστημάτων που εκτελούν υπολογισμούς, και
 - οι πρακτικές εφαρμογές που προκύπτουν από την υλοποίηση και εφαρμογή των αρχών αυτών σε υπολογιστικά συστήματα.
- Σε αυτό το μάθημα, θα μάθετε να
 - γράφετε κώδικα: καλές πρακτικές, αποφυγή λαθών, αποδοτικό πρόγραμμα
 - διαβάζετε κώδικα: κατανόηση του υπολογισμού, εύρεση λαθών
 - σκέπτεστε υπολογιστικά: ποιά είναι η διαδικασία που λύνει ένα πρόβλημα
 - λύνετε προβλήματα: κατανομή σε υποπροβλήματα, οργάνωση της λύσης



Οργάνωση του μαθήματος

- Παρακολούθηση και φόρτος μελέτης
 - Δύο διαλέξεις
 - Επιπλέον μελέτη τουλάχιστον 4-6 ώρες
- Ασκήσεις για το σπίτι (40%)
 - Προγραμματιστικές (κώδικας)
 - Παράδοση: TBD
- Διαγωνίσματα
 - Πρόοδος (20%): στη μέση του εξαμήνου
Κάποιο Σάββατο του Νοεμβρίου
Η ημερομηνία θα ανακοινωθεί αρκετά νωρίτερα
Ρυθμίστε το ημερολόγιό σας
 - Τελικό Διαγώνισμα (40%): στο τέλος του εξαμήνου, όλη η ύλη
 - Επαναληπτικό Διαγώνισμα (40%): το Σεπτέμβρη, όλη η ύλη



- Εισαγωγή στο εργαστήριο, 1-2 διαλέξεις
- Μεταπτυχιακοί βοηθοί
 - Βοηθούν στις ασκήσεις
 - Λύνουν απορίες για τη C
- Προγραμματιστικές ασκήσεις
 - Οι προθεσμίες θα είναι στη σελίδα του μαθήματος
 - Πρέπει να τρέχουν στα μηχανήματα του εργαστηρίου



- Αντιγραφή είναι
 - η χρήση κώδικα άλλου (συμφοιτητή, τρίτου, internet, κλπ.)
 - η συνεργασία για την επίλυση άσκησης
 - η χρήση κειμένου άλλου
 - η επικοινωνία με οποιονδήποτε πλην των διδασκόντων κατά τη διάρκεια εξέτασης
- Η αντιγραφή ανιχνεύεται εύκολα
 - Αυτόματη ανάλυση και σύγκριση με κώδικα συμφοιτητών, παρόμοιο κώδικα στο internet, παλαιότερες λύσεις
 - Αδυναμία εξήγησης του κώδικα στο εργαστήριο



- Short-term gain, Long-term pain
 - Γνώσεις προγραμματισμού χρειάζονται σε ολόκληρη τη φοίτηση
 - Η Εισαγωγή στην Επιστήμη Υπολογιστών είναι βάση για τα περισσότερα μαθήματα του τμήματος
- Η αντιγραφή θέλει δύο
 - Μην αφήνετε τους άλλους να αντιγράψουν από εσάς
 - Κλειδώστε την περιοχή σας στο εργαστήριο
 - Μην δίνετε τις λύσεις σας σε επόμενες χρονιές



Σκοπός μαθήματος

- Ο τρόπος σκέψης της Επιστήμης Υπολογιστών
 - Υπολογιστικά προβλήματα
 - Αλγοριθμική σκέψη, λύση προβλημάτων
- Εισαγωγή στον Προγραμματισμό
 - Εκμάθηση της γλώσσας C (γραφή, ανάγνωση)
 - Υλοποίηση αλγορίθμων
 - Καλές προγραμματιστικές συνήθειες
- Βασικές υπολογιστικές έννοιες
 - Δεδομένα και οργάνωση της μνήμης
 - Υπολογιστικές μηχανές και περιγραφή υπολογισμών
- Βάση για επαγγελματικές δεξιότητες
 - Σχεδιασμός και οργάνωση του προγράμματος
 - Ανάγνωση και κατανόηση του προγράμματος
 - Πώς μαθαίνεται μια γλώσσα προγραμματισμού



Περιεχόμενα Μαθήματος

- 1 Εισαγωγή
- 2 Αλγόριθμοι και Προγράμματα
- 3 Τύποι Μεταβλητών, Τελεστές, Είσοδος/Εξοδος
- 4 Εντολές Ελέγχου Ροής
- 5 Εντολές Επανάληψης
- 6 Συναρτήσεις
- 7 Συναρτήσεις και Μεταβλητές
- 8 Αλφαριθμητικά
- 9 Πίνακες
- 10 Αλγόριθμοι Πινάκων
- 11 Δείκτες, part I
- 12 Δείκτες, part II
- 13 Διαγώνισμα Προόδου
- 14 Δυναμική Διαχείριση Μνήμης
- 15 Διαχείριση Αρχείων
- 16 Αναδρομή
- 17 Ταξινόμηση και Αναζήτηση
- 18 Quicksort
- 19 Δομημένοι τύποι (structs)
- 20 Αναδρομικές Δομές
- 21 Απλά Συνδεδεμένες Λίστες
- 22 Διπλά Συνδεδεμένες Λίστες
- 23 Άλλα Χαρακτηριστικά της C
- 24 Άλλες προγραμματιστικές τεχνικές
- 25 Τελικό Διαγώνισμα



- Εγκαταστήστε ένα καλό περιβάλλον προγραμματισμού στο σπίτι
- Διάφορες διανομές του Linux
 - Ubuntu, Debian, Fedora, ...
- Compiler, IDE (Integrated Development Environment)
 - Editors: emacs, vim, pico, (XCode, Eclipse) ...
 - GCC (GNU Compiler Collection): μεταγλωττιστής της C
- Για τα περισσότερα προγράμματα, ασκήσεις, παραδείγματα κλπ του μαθήματος αρκεί οποιοδήποτε από τα παραπάνω
 - Στο εργαστήριο/φροντιστήριο υποστηρίζεται ο GCC



- Στις ώρες των εργαστηρίων, άμεσες ερωτήσεις στους βοηθούς
- Mailing list: hy100-list@csd.uoc.gr
 - Κίνηση σχετική με το μάθημα, όχι forum γενικού ενδιαφέροντος
 - Netiquette
 - Εγγραφή με email στο majordomo@csd.uoc.gr, χωρίς Subject, με κείμενο
subscribe hy100-list
- Στα φροντιστήρια, στις ώρες γραφείου των βοηθών και του διδάσκοντος
- Στο **Google**
 - Μάθετε να χρησιμοποιείτε τις μηχανές αναζήτησης σωστά
 - Ψάξτε online πριν στείλετε ερωτήσεις σε όλους στη λίστα



- Ρωτήστε, ενημερωθείτε, διαβάστε, ζητήστε βοήθεια
 - Active learning
- Εκμεταλλευτείτε τα εργαστήρια, τους βοηθούς, τις ώρες γραφείου
- Ασχοληθείτε με τον προγραμματισμό ενεργά
 - Όχι μόνο τις ασκήσεις
 - Όχι μόνο την τελευταία στιγμή
 - 6 ώρες μελέτη την εβδομάδα $\neq$ 24 ώρες την τελευταία μέρα του μήνα
- Ο προγραμματισμός είναι σαν το ποδήλατο
 - Όσα βιβλία και να διαβάσει κανείς, δεν ξέρει
 - Μαθαίνεται μόνο κάνοντας, στην πράξη
 - Στην αρχή όλοι πέφτουν



Υπολογιστική σκέψη

- Στόχος: να μάθετε να σκέπτεστε σαν επιστήμονες υπολογιστών
 - Βάλτε τον υπολογιστή να κάνει αυτό που θέλετε



Υπολογιστική σκέψη

- Στόχος: να μάθετε να σκέπτεστε σαν επιστήμονες υπολογιστών
 - Βάλτε τον υπολογιστή να κάνει αυτό που θέλετε

Παράδειγμα 1

Ο αριθμός $\sqrt{y}$ είναι ο x , τέτοιος ώστε $x^2 = y$



Υπολογιστική σκέψη

- Στόχος: να μάθετε να σκέπτεστε σαν επιστήμονες υπολογιστών
 - Βάλτε τον υπολογιστή να κάνει αυτό που θέλετε

Παράδειγμα 1

Ο αριθμός $\sqrt{y}$ είναι ο x , τέτοιος ώστε $x^2 = y$

Παράδειγμα 2

- 1 Μάντεψε κάποιο x
- 2 Αν $x^2 \simeq y$ τότε ρίζα είναι το x
- 3 Αν όχι, επανέλαβε από το παραπάνω βήμα με $x = \frac{x+y/x}{2}$



Υπολογιστική σκέψη

- Στόχος: να μάθετε να σκέπτεστε σαν επιστήμονες υπολογιστών
 - Βάλτε τον υπολογιστή να κάνει αυτό που θέλετε
- Τί είναι υπολογισμός;
 - Διατύπωση: περιγραφή μιας ακολουθίας βημάτων
 - Συγκεκριμένη σειρά: αποφασίζω ποιό είναι το επόμενο βήμα και το εκτελώ
 - Συνθήκη τερματισμού: πότε σταματώ και ποιά είναι η απάντηση



Υπολογιστική σκέψη

- Στόχος: να μάθετε να σκέπτεστε σαν επιστήμονες υπολογιστών
 - Βάλτε τον υπολογιστή να κάνει αυτό που θέλετε
- Τί είναι υπολογισμός;
 - Διατύπωση: περιγραφή μιας ακολουθίας βημάτων
 - Συγκεκριμένη σειρά: αποφασίζω ποιό είναι το επόμενο βήμα και το εκτελώ
 - Συνθήκη τερματισμού: πότε σταματώ και ποιά είναι η απάντηση
- Πώς αυτοματοποιούμε τους υπολογισμούς της τετραγωνικής ρίζας;
 - Κατασκευάζουμε μια μηχανή που τους εκτελεί
 - Ένα κύκλωμα για να ελέγχει τη συνθήκη
 - Ένα κύκλωμα για να υπολογίζει το κλάσμα
 - κλπ.

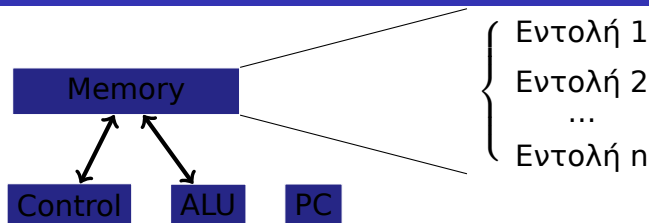


Οι πρώτοι υπολογιστές

- Αρχικοί υπολογιστές: ένα κύκλωμα που εκτελεί μια συγκεκριμένη σειρά από υπολογισμούς
 - Bombe, ENIAC, αριθμομηχανές
- Λέγονται *Fixed-program computers*
 - Κατασκευασμένοι για μια μόνο λειτουργία (πρόσθεση, αφαίρεση, τετραγωνική ρίζα, κλπ)
 - Για κάθε πρόβλημα, χρειάζεται η κατασκευή νέου κυκλώματος
- Ιδανικό κύκλωμα:
 - Παίρνει την περιγραφή ενός άλλου κυκλώματος, και υπολογίζει το αποτέλεσμα που θα έδινε αυτό
- Ακόμη καλύτερα:
 - Παίρνει την περιγραφή μιας σειράς βημάτων, και υπολογίζει το αποτέλεσμά τους
- Οι σημερινοί υπολογιστές: *stored-program computers*
 - Μπορούμε να δώσουμε στον υπολογιστή μια σειρά από εντολές που περιγράφουν τη διαδικασία που θέλουμε να εκτελέσει
 - Λέγονται και *μηχανές von Neumann*



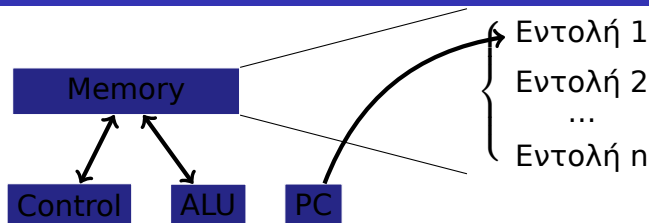
Περιγραφή ενός stored-program computer



- Πρόγραμμα: μια σειρά από εντολές (σαν συνταγή)
 - Πεπερασμένος αριθμός εντολών
 - Περιγράφουν όλους τους δυνατούς υπολογισμούς
- Ποιές είναι οι κατάλληλες εντολές;
 - 1936, Alan Turing: Όλοι οι υπολογισμοί μπορούν να περιγραφούν με 6 απλές εντολές
 - Όποιος υπολογισμός μπορεί να περιγραφεί με μια γλώσσα προγραμματισμού, μπορεί με όλες



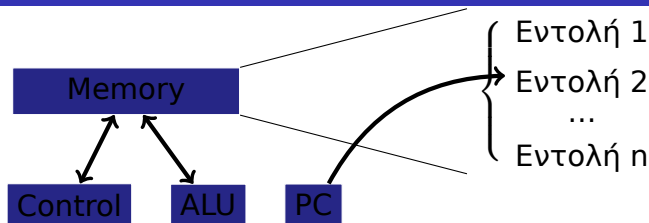
Περιγραφή ενός stored-program computer



- Πρόγραμμα: μια σειρά από εντολές (σαν συνταγή)
 - Πεπερασμένος αριθμός εντολών
 - Περιγράφουν όλους τους δυνατούς υπολογισμούς
- Ποιές είναι οι κατάλληλες εντολές;
 - 1936, Alan Turing: Όλοι οι υπολογισμοί μπορούν να περιγραφούν με 6 απλές εντολές
 - Όποιος υπολογισμός μπορεί να περιγραφεί με μια γλώσσα προγραμματισμού, μπορεί με όλες



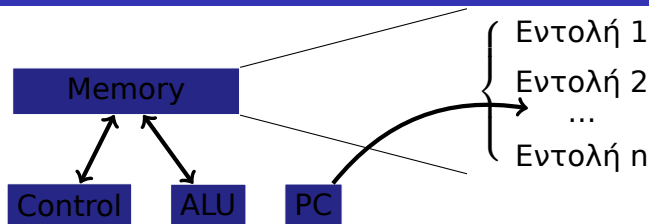
Περιγραφή ενός stored-program computer



- Πρόγραμμα: μια σειρά από εντολές (σαν συνταγή)
 - Πεπερασμένος αριθμός εντολών
 - Περιγράφουν όλους τους δυνατούς υπολογισμούς
- Ποιές είναι οι κατάλληλες εντολές;
 - 1936, Alan Turing: Όλοι οι υπολογισμοί μπορούν να περιγραφούν με 6 απλές εντολές
 - Όποιος υπολογισμός μπορεί να περιγραφεί με μια γλώσσα προγραμματισμού, μπορεί με όλες



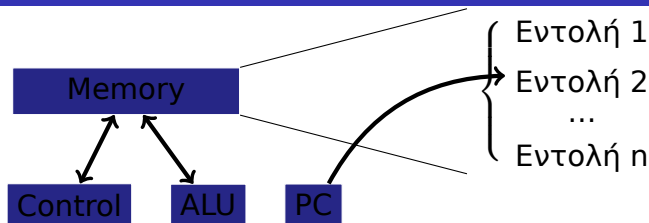
Περιγραφή ενός stored-program computer



- Πρόγραμμα: μια σειρά από εντολές (σαν συνταγή)
 - Πεπερασμένος αριθμός εντολών
 - Περιγράφουν όλους τους δυνατούς υπολογισμούς
- Ποιές είναι οι κατάλληλες εντολές;
 - 1936, Alan Turing: Όλοι οι υπολογισμοί μπορούν να περιγραφούν με 6 απλές εντολές
 - Όποιος υπολογισμός μπορεί να περιγραφεί με μια γλώσσα προγραμματισμού, μπορεί με όλες



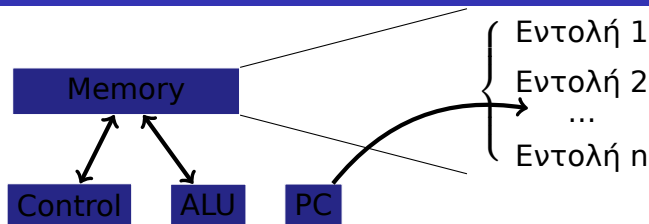
Περιγραφή ενός stored-program computer



- Πρόγραμμα: μια σειρά από εντολές (σαν συνταγή)
 - Πεπερασμένος αριθμός εντολών
 - Περιγράφουν όλους τους δυνατούς υπολογισμούς
- Ποιές είναι οι κατάλληλες εντολές;
 - 1936, Alan Turing: Όλοι οι υπολογισμοί μπορούν να περιγραφούν με 6 απλές εντολές
 - Όποιος υπολογισμός μπορεί να περιγραφεί με μια γλώσσα προγραμματισμού, μπορεί με όλες



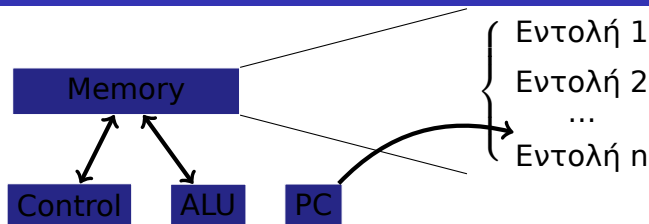
Περιγραφή ενός stored-program computer



- Πρόγραμμα: μια σειρά από εντολές (σαν συνταγή)
 - Πεπερασμένος αριθμός εντολών
 - Περιγράφουν όλους τους δυνατούς υπολογισμούς
- Ποιές είναι οι κατάλληλες εντολές;
 - 1936, Alan Turing: Όλοι οι υπολογισμοί μπορούν να περιγραφούν με 6 απλές εντολές
 - Όποιος υπολογισμός μπορεί να περιγραφεί με μια γλώσσα προγραμματισμού, μπορεί με όλες



Περιγραφή ενός stored-program computer



- Πρόγραμμα: μια σειρά από εντολές (σαν συνταγή)
 - Πεπερασμένος αριθμός εντολών
 - Περιγράφουν όλους τους δυνατούς υπολογισμούς
- Ποιές είναι οι κατάλληλες εντολές;
 - 1936, Alan Turing: Όλοι οι υπολογισμοί μπορούν να περιγραφούν με 6 απλές εντολές
 - Όποιος υπολογισμός μπορεί να περιγραφεί με μια γλώσσα προγραμματισμού, μπορεί με όλες



Γλώσσες προγραμματισμού

- Το πρόγραμμα είναι μια “συνταγή”
- Για να περιγράψουμε μια συνταγή χρειάζεται μια γλώσσα:
 - Ένα σύνολο από μικρά, “βασικά” βήματα
 - Π.χ: γλώσσα μαγειρικών συνταγών: “καθαρίζω, ξεφλουδίζω, κόβω, ανακατεύω, βάζω στο φούρνο, βράζω, κλπ”
 - Ένα τρόπο να περιγράψουμε δεδομένα
 - Π.χ: “μια κούπα ζάχαρη, 100 γραμμάρια βούτυρο, 2 αυγά, 1 πλάκα σοκολάτα”
- Για να περιγράψουμε έναν υπολογισμό, το ίδιο
- Μια γλώσσα έχει συντακτικό και γραμματική
 - Κανόνες για το ποιά πρόταση είναι μέρος της γλώσσας
 - Π.χ: “Πρόσθεσε δύο συν δύο” είναι πρόταση της ελληνικής γλώσσας
 - αλλά “Add two plus two” δεν είναι
- Μια γλώσσα έχει σημασιολογία
 - Τι σημαίνει “κούπα”, “ζάχαρη”, “βράζω”, κλπ;
 - Τι σημαίνει “πρόσθεσε”, “δυσ”, κλπ;



Το μάθημα του προγραμματισμού

- Πώς φτιάχνουμε καλές συνταγές;
- Θα χρησιμοποιήσουμε τη γλώσσα C για να περιγράψουμε υπολογισμούς
- Σκοπός όμως είναι ο τρόπος σκέψης
 - Αν μάθετε να σκέφτεστε σωστά, και ξέρετε να εκφράζετε τη σκέψη σας σε C, θα μπορείτε να μάθετε μια άλλη γλώσσα σε 1-2 εβδομάδες μόνοι σας
 - Δεν υπάρχει “η καλύτερη γλώσσα”
 - Για κάθε είδος προβλήματος μπορεί να βολεύει άλλη γλώσσα
 - Υπάρχουν όμως “χειρότερες” γλώσσες...



Για την επόμενη φορά

- Εξοικειωθείτε με το Linux
- Εγγραφείτε στη λίστα του μαθήματος
- Κλειδώστε την περιοχή σας
- Μάθετε έναν text editor
- Εγκαταστήστε κάποιο περιβάλλον προγραμματισμού στον υπολογιστή σας

