

Προγραμματισμος : Θεμέλιο

Μέρος 1ο :

Ανάλυση, αλγόριθμος, κώδικας, συναρτήσεις, βιβλιοθήκες, μεταγλώττιση. Βελτιστοποίηση & παράλληλος προγραμματισμός. Επιλογή γλώσσας προγραμματισμού.

Ανάλυση & αλγόριθμος

Ορισμός : Αλγόριθμος ενός προβλήματος είναι η αναλυτική περιγραφή της ακολουθίας των βημάτων που απαιτούνται για την επίλυση του προβλήματος.

Πιο απλά είναι η αναλυτική περιγραφή της μεθόδου επίλυσης ενός προβλήματος.

Ειδικά για τον προγραμματισμό υπολογιστών, ο αλγόριθμος είναι μια περιγραφή της λογικής του προγράμματος.

Ανάλυση & αλγόριθμος

Παράδειγμα

Υπάρχει, και εάν ναι, που, η υπακολουθία "FLARE"
στην κάτωθι αλληλουχία :

```
MGLPPLLSLP SNSAPRSLGR VETPPEVVDF MVSLAEAPRG GRVLEPACAH  
GPFLRAFREA HGTGYRFVGV EIDPKALDLP PWAEGILADF LLWEPGEAFD  
LILGNPPYGI VGEASKYPIH VFKAVKDLYK KAFSTWKGY NLYGAFLEKA  
VRLKPGGVL VFVVPATWLV LEDFALLREF LAREGKTSVY YLGEVFPQKK  
VSAVVIRFQK SGKGLSLWDT QESESGFTPI LWAEYPHWEG EIIRFETEET  
RKLEISGMPL GDLFHIRFAA RSPEFKKHPA VRKEPGPGLV PVL'TGRNLKP  
GWVDYEKNHS GLWMPKERAK ELRDFYATPH LVVAHTKGTR VVAAWDERAY  
PWREEFHLLP KEGVRLDPSS LVQWLNSEAM QKHVRTLYRD FVPHLTLRML  
ERLPVRREYG FHTSPESARN F
```

Ανάλυση & αλγόριθμος

Παράδειγμα

- Για κάθε θέση (i) της αλληλουχίας D :
- Για κάθε θέση (k) της αλληλουχίας S :
 - Ισχύει $D[i+k] == S(k)$ για κάθε (k) ?
 - * Εάν ναι : η θέση (i) είναι λύση
 - * Εάν όχι : αύξησε το (i) κατά 1 και επανέλαβε (μέχρι το τέλος της αλληλουχίας D).

Ανάλυση & αλγόριθμος

Παράδειγμα

```
for ( i=0 ; i < μήκος(D) ; i++ )
{
    for ( k=0 ; k < μήκος(S) ; k++ )
        if ( D[i+k] != S[k] )
            break;

    if ( k == μήκος(S) )
        τύπωσε "Λύση για τη θέση (i)"
}
```

Προγράμματα υπολογιστών

Ορισμός : Ένα πρόγραμμα είναι ένας αλγόριθμος κωδικοποιημένος σε μια γλώσσα προγραμματισμού.

Γλώσσα προγραμματισμού είναι ένα σύνολο εντολών κατανοητών από ηλεκτρονικό υπολογιστή. Υπάρχουν πολλές τέτοιες γλώσσες (π.χ. assembly, C, C++, FORTRAN, Perl, python, javascript, ...), κάθε μια με τα δικά της ιδιοτυπικά χαρακτηριστικά.

Διάκριση ανάμεσα σε κώδικα μηχανής (εκτελέσιμο πρόγραμμα) και το ίδιο πρόγραμμα γραμμένο σε κάποια υψηλού επιπέδου γλώσσα προγραμματισμού.

Συναρτήσεις.

Οι συναρτήσεις είναι αυτοτελή τμήματα κώδικα που ολοκληρώνουν μια συγκεκριμένη εργασία. Οι συναρτήσεις συνήθως παίρνουν δεδομένα, τα επεξεργάζονται και επιστρέφουν ένα αποτέλεσμα. Μόλις γραφτεί μια συνάρτηση, μπορεί να χρησιμοποιηθεί ξανά και ξανά. Οι συναρτήσεις άπαξ και έχουν γραφτεί, μπορούν να χρησιμοποιηθούν από οποιοδήποτε άλλο πρόγραμμα με απλή αναφορά του ονόματος τους.

Βιβλιοθήκες.

Μια βιβλιοθήκη προγραμματισμού είναι μια συλλογή από ήδη γραμμένο και ελεγμένο κώδικα (με την μορφή συναρτήσεων) που μπορούν να χρησιμοποιηθούν σε άλλα προγράμματα.

Αυτή η συλλογή επαναχρησιμοποιήσιμου κώδικα συνήθως στοχεύει σε συνηθισμένα/κοινά προβλήματα ώστε να μην πρέπει να ξανά-ανακαλύπτουμε τον τροχό για κάθε πρόβλημα.

Μεταγλώττιση.

Ανάμεσα σε εσάς (που γράφετε ένα πρόγραμμα σε μια υψηλού επιπέδου γλώσσα προγραμματισμού) και το μηχάνημα (το οποίο μπορεί μόνο να εκτελέσει ένα μικρό σύνολο από απλοϊκές εντολές) παρεμβάλλεται ένα άλλο πρόγραμμα, ο μεταγλωττιστής.

Ο μεταγλωττιστής με βάση το πρόγραμμα που φτιάξατε, δημιουργεί ένα καινούργιο αρχείο το οποίο περιέχει το ίδιο πρόγραμμα αλλά γραμμένο σε μια γλώσσα τόσο χαμηλού επιπέδου ώστε να είναι άμεσα εκτελέσιμη από τον υπολογιστή σας.

Παρένθεση: compiled ή interpreted ?

Υπάρχουν γλώσσες στις οποίες δεν υπάρχει διακριτό στάδιο μεταγλώττισης, οι λεγόμενες γλώσσες με διερμηνέα (interpreted languages).

Σε μία compiled language, το πρόγραμμα μετατρέπεται σε κώδικα μηχανής έτσι ώστε ο επεξεργαστής να μπορεί να το εκτελέσει. Σε μία interpreted language, οι εντολές του προγράμματος εκτελούνται απευθείας από τον interpreter της γλώσσας.

Βελτιστοποίηση.

Don't.

"Premature optimization is the root of all evil in programming."

Donald Knuth

Παράλληλος προγραμματισμός.

Ο παράλληλος προγραμματισμός είναι η διαδικασία διάσπασης ενός προβλήματος σε μικρότερες εργασίες που η κάθε μια από αυτές (τις μικρότερες εργασίες) μπορεί να εκτελεστεί ανεξάρτητα από τις υπόλοιπες. Με τον τρόπο αυτό, μπορούν να χρησιμοποιηθούν πολλοί υπολογίστες ταυτόχρονα για την λύση ενός προβλήματος.

Shared memory : Threads & OpenMP

Message-passing : MPI (Message Passing Interface)

Επιλογή γλώσσας

Ένα από τα καλύτερα κριτήρια για αρχάριους προγραμματίστες είναι το ποια γλώσσα 'φοριέται' περισσότερο στο οικείο εργαστήριο. Είναι μεγάλη βοήθεια το να υπάρχουν άμεσα διαθέσιμοι συνάδελφοι με προηγούμενη εμπειρία.

Το δεύτερο κριτήριο είναι το εάν το πρόβλημα απαιτεί ειδικά εργαλεία που υπάρχουν μόνο σε μια γλώσσα/περιβάλλον.

Επιλογή γλώσσας

Το τρίτο κριτήριο είναι ταχύτητα εκτέλεσης : για προγράμματα που πρόκειται να τρέχουν για εβδομάδες, οι `compiled languages` έχουν σαφή πλεονεκτήματα.