Εισαγωγή στις Υποδομές του Προγραμματισμού

- Τηλεπικοινωνίες, Ασφάλεια και Λειτουργικά Συστήματα
- Μάθημα: Εισαγωγή στον Προγραμματισμό (Matlab & Python)

Εισαγωγικές γνώσεις προαπαιτούμενες

- Τι είναι οι υπολογιστές και πως λειτουργούν;
- Τι είναι προγραμματισμός;
- Τι είναι κώδικας;
- Πού 'ζει' ο κώδικας που δημιουργούμε;
- Οι απαντήσεις των παραπάνω ερωτημάτων είναι απαραίτητες για την κατανόηση του περιβάλλοντος που χρησιμοποιούμε, αλλά του κώδικα που θα παράξουμε σε γλώσσα C του matlab και στη γλώσσα της python.

Προγραμματισμός

- Προγραμματισμός είναι η σύνταξη ‘οδηγιών’ σε μια μηχανή ώστε να εκτελεί συγκεκριμένες λειτουργίες ή να επιλύει προβλήματα.

- Οι ‘οδηγίες’ αποτελούν τον ‘κώδικα’ (source code) της μηχανής.

- Ο κώδικας αυτός έχει συγκεκριμένο συντακτικό ανάλογα με τη γλώσσα προγραμματισμού που επιλέγουμε

Γλώσσα προγραμματισμού

- Η γλώσσα προγραμματισμού είναι η προσπάθεια του ανθρώπου να δημιουργήσει μια γλώσσα επικοινωνίας μεταξύ του ίδιου και μηχανών ή μεταξύ μηχανών.

- Σκοπός της γλώσσας είναι:

- Να δώσει με σαφήνεια οδηγίες σε μια μηχανή
- Οι οδηγίες αυτές να μπορούν να διαβαστούν και από την μηχανή, αλλά και από τον άνθρωπο

Γλώσσα προγραμματισμού

- Όπως κάθε γλώσσα έχει συντακτικό.
- Οι οδηγίες που δίνονται με αυτό τον τρόπο είναι μονοσήμαντες, δεν επιδέχονται ερμηνείες και ταυτίζονται με την μαθηματική γλώσσα.
- Ο προγραμματιστής θα πρέπει να προσαρμοστεί με την γλώσσα και όχι το αντίστροφο.

Γλώσσα προγραμματισμού

- Οι γλώσσες προγραμματισμού χωρίζονται σε γλώσσες χαμηλού επιπέδου (low level), υψηλού επιπέδου (high level) και γλώσσες scripting.
- Οι γλώσσες χαμηλού επιπέδου μπορούν να ελέγξουν το λειτουργικό σύστημα (OS), πχ. C++, ή να δημιουργήσουν το λειτουργικό σύστημα, assembly.

Γλώσσα προγραμματισμού

- Οι γλώσσες υψηλού επιπέδου (high level) χρησιμοποιούν βιβλιοθήκες που είναι δημιουργημένες από γλώσσες low level ή από επίσης γλώσσες high level.
- Οι γλώσσες υψηλού επιπέδου είναι πιο εύκολες στη σύνταξη και στη δημιουργία προγραμμάτων και το περιβάλλον ανάπτυξης τους βοηθάει τον προγραμματιστή. Matlab, C, C#, Java, κ.α.
- Οι high level γλώσσες τείνουν να μιμηθούν την ανθρώπινη γλώσσα.

Γλώσσα προγραμματισμού

- Οι Scripting γλώσσες (ουσιαστικά πολύ υψηλού επιπέδου) είναι γενικού περιεχομένου προγραμματισμού. Στην ουσία χρησιμοποιούν απευθείας βιβλιοθήκες ή εντολές του λειτουργικού.

Παραδείγματα: JS (Javascript) ή Python.

Η python θεωρείται ως μια advanced scripting language.

Τι είναι το Λειτουργικό Σύστημα (Operating System - OS)

- Το λογισμικό που διαχειρίζεται το hardware και τις εφαρμογές.
- Παρέχει διεπαφή χρήστη (GUI ή Command Line).
- Παραδείγματα: Windows OS, macOS OS, Linux OS.

GUI – Διεπαφή χρήστη

- Το GUI (Graphical User Interface) – διεπαφή χρήστη (interface), είναι ουσιαστικά τα ψηφιακά εργαλεία για να επικοινωνεί ο άνθρωπος με μια μηχανή ή μια μηχανή με μια άλλη μηχανή.

Παραδείγματα:

- Εικονικό κουμπί σε μια εφαρμογή μπορεί να δώσει μια σειρά από οδηγίες σε μια μηχανή.
- Φόρμα συμπλήρωσης δεδομένων

GUI – Διεπαφή χρήστη

- Μια γραφική παράσταση που τη δημιουργεί ένα υπολογιστής.
- Ένα μήνυμα σφάλματος του υπολογιστή
- Ή ακόμη πιο πολύπλοκα συστήματα, όπως φωνητικές οδηγίες, touch screen, κλπ.

- Παρόμοια αν η επικοινωνία είναι μεταξύ μηχανών είναι ένα API, που μπορεί μια μηχανή να μιλάει με μια άλλη ακόμη και αν δεν βρίσκονται στον ίδιο χώρο (μέσω πρωτοκόλλων επικοινωνίας, Wifi, internet, Bluetooth) ή ακόμη και αν βρίσκονται σε διαφορετικό OS (cross-platform).

Κύριες Λειτουργίες του OS

- Διαχείριση μνήμης (memory management).
- Διαχείριση επεξεργαστή (CPU scheduling).
- Διαχείριση αρχείων και δικτύου.
- Ασφάλεια και άδειες χρήσης.

Το Σύστημα Αρχείων (File System)

- Όλα τα δεδομένα είναι αποθηκευμένα σε αρχεία.
- Δομή φακέλων διαφέρει ανά OS (π.χ. C:\Users vs /home/).
- Ο προγραμματιστής πρέπει να γνωρίζει πού γράφει/διαβάζει δεδομένα.

Διαδικασίες (Processes)

- Κάθε πρόγραμμα που εκτελείται είναι ένα process.
- Το OS διαχειρίζεται CPU (επεξεργαστής υπολογιστή) και μνήμη για κάθε process.
- Multitasking: εκτέλεση πολλών processes παράλληλα σε threads/cores.

Ασφάλεια στο OS (System Security)

- Περιορισμός πρόσβασης μέσω χρηστών και δικαιωμάτων (user, admin, root).
- Προστασία αρχείων και διεργασιών.
- Ενημερώσεις ασφαλείας.

Απόδοση και Ταχύτητα (Performance)

- Το OS κατανέμει πόρους (resoursess) = (CPU, μνήμη RAM) στα processes.
- Εργαλεία: Task Manager (Windows), top (Linux).
- Πολλά processes = μειωμένη απόδοση.

Internet

Λίγα λόγια για το internet και τη σημασία του στους υπολογιστές/mobile/IoT και γενικότερα μηχανές:

- Παγκόσμιο δίκτυο υπολογιστών που επικοινωνούν μέσω πρωτοκόλλων (protocols).
- Κάθε υπολογιστής (ή συνδεδεμένο device) έχει μοναδική IP διεύθυνση.
- DNS: Μετατρέπει ονόματα (domains) σε IP.

Browser / Websites

1. Ο χρήστης γράφει μια διεύθυνση (URL).
2. Ο browser ζητά IP από DNS.
3. Δημιουργείται σύνδεση HTTP/HTTPS.
4. Ο server επιστρέφει HTML, CSS, JavaScript.

<u>Σημείωση:</u> Το ίδιο συμβαίνει και σε εφαρμογές κινητών (mobile apps): χρησιμοποιούν HTTP/HTTPS για να επικοινωνούν με servers.

Browser

Browser:

Είναι μια εφαρμογή για αυτόματη και προκαθορισμένη επικοινωνία με άλλους υπολογιστές στο internet ή σε τοπικά δίκτυα σε γραφικό περιβάλλον.

Η γλώσσα που χρησιμοποιεί για το γραφικό περιβάλλον είναι η HTML.

Παραδείγματα: Chrome, Firefox, Safari

Mobiles & Smart Devices

- Τα κινητά έχουν λειτουργικό σύστημα (mobile OS) – Android, iOS.
- Τρέχουν εφαρμογές (apps) όπως οι υπολογιστές τρέχουν προγράμματα.
- Χρησιμοποιούν HTTP/HTTPS, APIs,
- Wi-Fi/bluetooth/4G/5G για επικοινωνία.

<u>*Παράδειγμα:*</u>

εφαρμογή καιρού ↔ server ↔ API ↔ απεικόνιση στο κινητό.

Internet of Things (IoT)

IoT = Δίκτυο “έξυπνων” συσκευών που επικοινωνούν μέσω Internet.

<u>Παραδείγματα:</u> αισθητήρες, ρολόγια, βιομηχανικές μηχανές, οικιακές συσκευές.

Χρησιμοποιούν:

- Ελαφριά λειτουργικά (embedded OS)
- Πρωτόκολλα επικοινωνίας (HTTP, MQTT, TCPIP)
- Cloud για αποθήκευση & ανάλυση δεδομένων
- Ζητήματα: Ασφάλεια, καθυστέρηση (latency), ενέργεια.

Εφαρμογές

Mobile devices και Smart Devices,
όπως και IoT συστήματα μπορούν να
Προγραμματιστούν.

Παραδείγματα εφαρμογών:
Μπορούμε να λαμβάνουμε στο κινητό μας πληροφορίες από IoT δίκτυο, όπως κατανάλωση ρεύματος, κατάσταση διακοπτών On/Off, θερμοκρασία, υγρασία, κλπ.
Από το κινητό μας μπορούμε να αλλάξουμε τις συνθήκες λειτουργίας, όπως να αλλάξουμε τη θερμοκρασία ενός θερμοστάτη A/C ή να κλείσουμε/ανοίξουμε μια συσκευή.

Πρωτόκολλα επικοινωνίας

- Είναι ένα προκαθορισμένο σύνολο κανόνων που καθορίζει πως επικοινωνούν μηχανές (ψηφιακά) με σκοπό να ανταλλάξουν δεδομένα μεταξύ τους

- Στόχος τους είναι η διασφάλιση μεταφοράς των δεδομένων και γνωστοποίηση των δυο μερών της επικοινωνίας για το αποτέλεσμα.

Παραδείγματα: Http, Https, Tcp/ip, Wifi, Bluetooth, κλπ.

HTTP και HTTPS

- HTTP: HyperText Transfer Protocol – βασική μορφή επικοινωνίας στο Web.
- HTTPS: HTTP + ασφάλεια μέσω SSL/TLS (κρυπτογράφηση).
- Κρατά τα δεδομένα ασφαλή από υποκλοπή.

Ταχύτητα και Καθυστέρηση (Speed & Latency)

- Bandwidth: Πόσα δεδομένα μεταφέρονται ανά δευτερόλεπτο.
- Latency: Χρόνος που χρειάζεται για να φτάσουν τα δεδομένα.
- Ping: Μέτρηση καθυστέρησης επικοινωνίας.

Client και Server

- Client: Ζητά υπηρεσίες (browser, εφαρμογή).

Στα επόμενα μαθήματα το matlab online που χρησιμοποιούμε από το browser μας (chrome, firefox, etc) είναι ο Client.

- Server: Παρέχει υπηρεσίες (web server, API).

Στα επόμενα μαθήματα το matlab online παρέχεται ως υπηρεσία από το mathworks.

Ασφάλεια στις Τηλεπικοινωνίες

- Κίνδυνοι: υποκλοπή, αλλοίωση δεδομένων, πλαστοπροσωπία.
- Μέτρα: Κρυπτογράφηση, authentication, πιστοποιητικά SSL, firewall, VPN.
- Ο προγραμματιστής πρέπει να προστατεύει τα δεδομένα χρήστη.

APIs – Επικοινωνία μεταξύ προγραμμάτων

- API (Application Programming Interface): ορίζει τρόπους επικοινωνίας εφαρμογών.
- Παράδειγμα: εφαρμογή καιρού που παίρνει δεδομένα από API μέσω HTTP.
- Ανταλλαγή δεδομένων με διάφορες δομές διάταξης δεδομένων JSON/XML, ή binary μορφή.