

Τ.Ε.Ι ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΤΗΛΕΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ :

ΑΝΑΠΤΥΞΗ CGI ΓΛΩΣΣΑΣ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ (INTERPRETER)

ΩΣ WEB SERVER EXTENSION

ΕΙΣΗΓΗΤΗΣ : ΓΛΑΒΑΣ ΕΥΡΙΠΙΔΗΣ

ΣΠΟΥΔΑΣΤΗΣ : ΠΑΠΠΑΣ ΑΓΓΕΛΟΣ

ΑΡΤΑ 2004

ΠΕΡΙΕΧΟΜΕΝΑ

1. Η ΕΞΕΛΙΞΗ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

- 1.1. Γλώσσες Μηχανής**
- 1.2. Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου**
- 1.3. Γλώσσες υψηλού επιπέδου**
- 1.4. Πλεονεκτήματα των γλωσσών υψηλού επιπέδου**
- 1.5. Γλώσσες 4ης γενιάς**
- 1.6. Ταξινόμηση γλωσσών προγραμματισμού**

2. INTERNET ΚΑΙ WORLD WIDE WEB

- 2.1. Η εξέλιξη του Internet**
- 2.2. Οι υπηρεσίες του Internet**
- 2.3. Το World Wide Web (WWW) – Ο Παγκόσμιος Ιστός**
- 2.4. Η αρχιτεκτονική πελάτη – εξυπηρετητή (client – server) και το WWW**
- 2.5. Uniform Resource Locator (URL) – Ομοιόμορφος Εντοπιστής Πόρου**
- 2.6. Οι World Wide Web Browsers**
- 2.7. Ορισμός της HTML**
- 2.8. Τμήματα ενός HTML εγγράφου**

3. ΙΣΤΟΣΕΛΙΔΕΣ ΔΥΝΑΜΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ

- 3.1. Η εξέλιξη των ιστοσελίδων δυναμικού περιεχομένου**
- 3.2. Το CGI (Common Gateway Interface – Κοινή Διασύνδεση Πύλης)**
- 3.3. Η λειτουργία του CGI**
- 3.4. Η απλή αλληλεπίδραση Εξερευνητή – Εξυπηρετητή (Browser – Server)**
- 3.5. Η εκτέλεση των CGI προγραμμάτων στον εξυπηρετητή**
- 3.6. Η εκτέλεση ενός CGI προγράμματος μέσω μιας HTML φόρμας**
- 3.7. Μέθοδοι κλήσης προς τον εξυπηρετητή**

4. Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ScriptC

- 4.1. Τεχνολογία και περιβάλλον ανάπτυξης**
- 4.2. Διαδικασία εκτέλεσης ενός αρχείου**
- 4.3. Χαρακτηριστικά της γλώσσας**
- 4.4. Τελεστές και προτεραιότητα**
- 4.5. Εντολές**
- 4.6. Built-In Συναρτήσεις**
- 4.7. Συναρτήσεις**
- 4.8. Χειρισμός εξαιρέσεων κατά την διάρκεια εκτέλεσης**
- 4.9. Συνδεσιμότητα με βάσεις δεδομένων**

5. ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Η ΕΞΕΛΙΞΗ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

1.1 Γλώσσες Μηχανής

Ο ηλεκτρονικός υπολογιστής (H/Y) άνοιξε το δρόμο για την αυτοματοποίηση εργασιών που επεξεργάζονται δεδομένα (**data processes**), πράγμα που ήταν το όνειρο πολλών γενεών ανθρώπων. Όπως κατασκευάζεται σήμερα, ο H/Y μπορεί να αυτοματοποιεί την εκτέλεση τέτοιων εργασιών αν του δοθεί μια περιγραφή τους, που ονομάζεται πρόγραμμα, σε μια κατάλληλη γλώσσα, που ονομάζεται γλώσσα προγραμματισμού. Αρχικά για να μπορέσει ο υπολογιστής να εκτελέσει μία οποιαδήποτε λειτουργία, έπρεπε να δοθούν κατευθείαν οι κατάλληλες ακολουθίες από 0 και 1, δηλαδή εντολές σε μορφή κατανοητή από τον υπολογιστή αλλά πολύ δύσκολα κατανοητές από τον άνθρωπο. Ο τρόπος αυτός ήταν επίπονος και ελάχιστοι μπορούσαν να τον υλοποιήσουν, αφού απαιτούσε βαθιά γνώση του υλικού και της αρχιτεκτονικής του υπολογιστή. Οι εντολές που δέχεται ένας υπολογιστής μετατρέπονται σε ακολουθίες που αποτελούνται από 0 και 1, γίνονται δηλαδή εντολές σε **γλώσσα μηχανής** όπως ονομάζονται, έτσι ώστε να γίνουν κατανοητές από τον υπολογιστή και στη συνέχεια να μπορέσει να τις εκτελέσει. Η μόνη γλώσσα προγραμματισμού που μπορεί να εκτελεί τυφλά ο H/Y είναι η γλώσσα μηχανής του, η οποία όμως είναι πολύ χαμηλού επιπέδου και επομένως δύσχρηστη. Για να εκτελεστεί ένα πρόγραμμα πρέπει να γραφεί σ' αυτή τη γλώσσα, αυτό όμως είναι ένα εξαιρετικά δύσκολο εγχείρημα. Τόσο δύσκολο που πρακτικά η συγγραφή και συντήρηση μεγάλων προγραμμάτων σε γλώσσα μηχανής είναι σχεδόν αδύνατη.

1.2 Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου

Από τα πρώτα χρόνια άρχισαν να γίνονται προσπάθειες για τη δημιουργία μιας συμβολικής γλώσσας, η οποία ενώ θα έχει έννοια για τον άνθρωπο θα μετατρέπεται εσωτερικά από τους υπολογιστές στις αντίστοιχες ακολουθίες από 0 και 1. Για παράδειγμα η λέξη ADD (πρόσθεσε) ακολουθούμενη από δυο αριθμούς, είναι κατανοητή από τον άνθρωπο και απομνημονεύεται σχετικά εύκολα. Η εντολή αυτή θα μεταφραστεί από τον υπολογιστή σε μία ακολουθία δυαδικών ψηφίων και στη

συνέχεια μπορεί να εκτελεστεί. Το έργο της μετάφρασης το αναλαμβάνει ένα ειδικό πρόγραμμα, ο **συμβολομεταφραστής (assembler)**.

1.3 Γλώσσες υψηλού επιπέδου

Οι ανεπάρκειες των συμβολικών γλωσσών και η προσπάθεια για καλύτερη επικοινωνία ανθρώπου μηχανής οδήγησαν στα τέλη της δεκαετίας του 50 στην εμφάνιση των πρώτων γλωσσών προγραμματισμού υψηλού επιπέδου.

Το 1957 η **IBM** ανέπτυξε την πρώτη γλώσσα υψηλού επιπέδου τη **FORTRAN**. Το όνομα FORTRAN προέρχεται από τις λέξεις FORmula TRANslation που σημαίνουν μετάφραση τύπων. Η FORTRAN αναπτύχθηκε ως γλώσσα κατάλληλη για την επίλυση μαθηματικών και επιστημονικών προβλημάτων.

Το 1960 αναπτύχθηκε μία άλλη γλώσσα, σταθμός στον προγραμματισμό η γλώσσα **COBOL**. Η COBOL όπως δηλώνει και το όνομα της (COmmon Business Oriented Language-Κοινή γλώσσα προσανατολισμένη στις επιχειρήσεις) είναι κατάλληλη για ανάπτυξη εμπορικών εφαρμογών, και γενικότερα διαχειριστικών εφαρμογών, τομέας όπου η FORTRAN υστερούσε.

Μια από τις σημαντικότερες γλώσσα προγραμματισμού με ελάχιστη πρακτική εφαρμογή αλλά που επηρέασε ιδιαίτερα τον προγραμματισμό και τις επόμενες γλώσσες είναι η **ALGOL** (Algorithmic Language-Αλγοριθμική γλώσσα).

Στο χώρο της Τεχνητής Νοημοσύνης αναπτύχθηκαν δυο γλώσσες διαφορετικές από τις άλλες. Στα μέσα του 60 αναπτύχθηκε η **LISP** (LISt Processor- Επεξεργαστής λίστας), γλώσσα η οποία προσανατολίζεται σε χειρισμό λιστών από τα σύμβολα και η **PROLOG** (PROgramming LOGic- Λογικός Προγραμματισμός) στις αρχές του 70.

Δύο σημαντικότερες γλώσσες γενικού σκοπού, οι οποίες αναπτύχθηκαν τη δεκαετία του 60 αλλά χρησιμοποιούνται πάρα πολύ στις ημέρες μας, είναι η **BASIC** και η **PASCAL**.

Η γλώσσα προγραμματισμού **BASIC** (Beginner's All Purpose Symbolic Code-Συμβολικός Κώδικας Εντολών Γενικής Χρήσης για Αρχάριους) αρχικά αναπτύχθηκε, όπως δηλώνει και το όνομα της, ως γλώσσα για την εκπαίδευση αρχαρίων στον προγραμματισμό. Σχεδιάστηκε για να γράφονται σύντομα προγράμματα, τα οποία εκτελούνται με τη βοήθεια διερμηνευτή (interpreter).

Η γλώσσα **PASCAL** έφερε μεγάλες αλλαγές στον προγραμματισμό. Παρουσιάστηκε το 1970 και στηρίχθηκε πάνω στην ALGOL. Είναι μια γλώσσα γενικής χρήσης, η οποία είναι κατάλληλη τόσο για την εκπαίδευση όσο και τη δημιουργία ισχυρών προγραμμάτων κάθε τύπου. Χαρακτηριστικό της γλώσσας είναι η καταλληλότητα για τη δημιουργία δομημένων προγραμμάτων.

Μία ακόμη γλώσσα που γνώρισε μεγάλη διάδοση είναι η **γλώσσα C**. Η C χρησιμοποιήθηκε για την ανάπτυξη του λειτουργικού συστήματος Unix. Γλώσσα με ισχυρά χαρακτηριστικά, μερικά από αυτά κοινά με την Pascal κατάλληλη για την ανάπτυξη δομημένων εφαρμογών αλλά και με πολλές δυνατότητες γλώσσας χαμηλού επιπέδου. Η C εξελίχτηκε στη γλώσσα C++, που είναι αντικειμενοστραφής.

Τα τελευταία χρόνια χρησιμοποιείται ιδιαίτερα, ειδικά για προγραμματισμό στο Διαδίκτυο (Internet), η **JAVA**. Η JAVA είναι μια αντικειμενοστραφής γλώσσα που αναπτύχθηκε από την εταιρία SUN με σκοπό την ανάπτυξη εφαρμογών, που θα εκτελούνται σε κατακευματισμένα περιβάλλοντα, δηλαδή σε διαφορετικούς υπολογιστές οι οποίοι είναι συνδεδεμένοι με το Διαδίκτυο.

1.4 Πλεονεκτήματα των γλωσσών υψηλού επιπέδου

Στα πλεονεκτήματα των γλωσσών προγραμματισμού υψηλού επιπέδου σε σχέση με τις συμβολικές μπορούν να αναφερθούν:

- Ο φυσικότερος και πιο «ανθρώπινος» τρόπος έκφρασης των προβλημάτων. Τα προγράμματα σε γλώσσα υψηλού επιπέδου είναι πιο κοντά στα προβλήματα που επιλύουν.
- Η ανεξαρτησία από τον τύπο του υπολογιστή. Προγράμματα σε μία γλώσσα υψηλού επιπέδου μπορούν να εκτελεστούν σε οποιονδήποτε υπολογιστή με ελάχιστες ή καθόλου μετατροπές. Η δυνατότητα της μεταφερσιμότητας των προγραμμάτων είναι σημαντικό προσόν.
- Η ευκολία της εκμάθησης και εκπαίδευσης ως απόρροια των προηγούμενων.
- Η διόρθωση λαθών και η συντήρηση προγραμμάτων σε γλώσσα υψηλού επιπέδου είναι πολύ ευκολότερο έργο.
- Συνολικά οι γλώσσες υψηλού επιπέδου ελάττωσαν σημαντικά το χρόνο και το κόστος παραγωγής νέων προγραμμάτων, αφού λιγότεροι προγραμματιστές μπορούν σε μικρότερο χρόνο να αναπτύξουν προγράμματα που χρησιμοποιούνται σε περισσότερους υπολογιστές..

1.5 Γλώσσες 4ης γενιάς

Στις γλώσσες αυτές ο χρήστης ενός υπολογιστή έχει τη δυνατότητα σχετικά εύκολα να υποβάλει ερωτήσεις στο σύστημα ή να αναπτύσσει εφαρμογές που ανακτούν πληροφορίες από βάσεις δεδομένων και να καθορίζει τον ακριβή τρόπο εμφάνισης αυτών των πληροφοριών. Μια τέτοια γλώσσα είναι η **SQL** τα αρχικά της οποίας σημαίνουν Structured Query Language.

1.6 Ταξινόμηση γλωσσών προγραμματισμού

Όλες οι γλώσσες προγραμματισμού που έχουν αναπτυχθεί μέχρι σήμερα, αντιπροσωπεύουν διάφορες ιδέες πάνω στον προγραμματισμό και η κάθε μία είναι συνήθως καλύτερα προσαρμοσμένη σε ορισμένες κατηγορίες προβλημάτων. Η μεγάλη πλειοψηφία των γλωσσών ανήκει στην κατηγορία των διαδικασιακών (procedural) γλωσσών. Είναι γνωστές επίσης και ως αλγοριθμικές γλώσσες, γιατί είναι σχεδιασμένες για να επιτρέπουν την υλοποίηση αλγορίθμων. Άλλες κατηγορίες γλωσσών υψηλού επιπέδου είναι:

- Αντικειμενοστραφείς γλώσσες (object-oriented languages).

- Συναρτησιακές γλώσσες (functional languages) π.χ LISP.
- Μη διαδικασιακές γλώσσες (non procedural languages) π.χ PROLOG. Χαρακτηρίζονται επίσης ως γλώσσες πολύ υψηλού επιπέδου.
- Γλώσσες ερωταπαντήσεων (query languages) π.χ SQL.

Μια άλλη ταξινόμηση που μπορεί να προκύψει με βάση την περιοχή χρήσης. Με αυτό το κριτήριο διακρίνουμε:

- Γλώσσες γενικής χρήσης. Θεωρητικά κάθε γλώσσα γενικής χρήσης μπορεί να χρησιμοποιηθεί για την επίλυση οποιουδήποτε προβλήματος. Στην πράξη ωστόσο κάθε γλώσσα έχει σχεδιαστεί για να ανταποκρίνεται καλύτερα σε ορισμένη κατηγορία προβλημάτων
- Γλώσσες επιστημονικής κατεύθυνσης (science oriented languages) π.χ FORTRAN
- Γλώσσες εμπορικής κατεύθυνσης (business-oriented language) π.χ COBOL

Ας σημειωθεί ότι ορισμένες γλώσσες τα καταφέρνουν εξίσου καλά και στους δυο προηγούμενους τομείς π.χ BASIC, Pascal

- Γλώσσες προγραμματισμού συστημάτων (system programming languages) π.χ C.
- Γλώσσες τεχνητής νοημοσύνης (artificial intelligence languages) π.χ LISP, PROLOG
- Γλώσσες ειδικής χρήσης. Πρόκειται για γλώσσες που χρησιμοποιούνται σε ειδικές περιοχές εφαρμογών όπως π.χ στα γραφικά με υπολογιστή, στη ρομποτική, στη σχεδίαση ολοκληρωμένων κυκλωμάτων, στα Συστήματα Διοίκησης Βάσεων Δεδομένων, στην εκπαίδευση μέσω υπολογιστή κ.α.

Η μετάβαση από την συγγραφή προγραμμάτων για υπολογιστές από γλώσσα μηχανής στην χρήση γλωσσών υψηλού επιπέδου δεν έγινε αμέσως. Αρχικά οι γλώσσες υψηλού επιπέδου δεν προσέφεραν πολλές δυνατότητες και η πραγματική τους διαφορά από την συμβολική γλώσσα (**assembly**) δεν ήταν και πολύ μεγάλη. Η ανάγκη για την συγγραφή πιο δομημένων προγραμμάτων επέφερε την προσθήκη ενός βασικότερου πλέον χαρακτηριστικού στις γλώσσες υψηλού επιπέδου: Την δυνατότητα δημιουργίας υποπρογραμμάτων (ρουτινών) μέσα στο ίδιο το πρόγραμμα, όπου θα μπορούσαν να υπάρχουν αλγόριθμοι παραμετροποιήσιμοι και ο

προγραμματιστής να μπορεί να ζητά την εκτέλεση αυτών των ρουτινών όταν αυτό ήταν απαραίτητο, χωρίς να ξαναγράφει τον αλγόριθμο. Το στάδιο αυτό ονομάστηκε **δομημένος προγραμματισμός** και το κύριο χαρακτηριστικό του ήταν η δυνατότητα επαναχρησιμοποίησης τμημάτων προγραμμάτων, η δημιουργία δηλαδή βιβλιοθηκών. Με τον καιρό οι ανάγκες για λογισμικό έγιναν πιο απαιτητικές και έτσι δημιουργήθηκε ένα νέο χαρακτηριστικό το οποίο υιοθετούν μέχρι και σήμερα οι γλώσσες προγραμματισμού: Τον σχεδιασμό και την χρήση αντικειμένων για την διεκπεραίωση των προγραμμάτων. Η δυνατότητα αυτή έρχεται να δώσει λύση σε πολύ πιο σύνθετα και μεγαλύτερα προβλήματα και η ανάπτυξη λογισμικού γίνεται πιο δομημένη και ιεραρχημένη από ποτέ. Το στάδιο αυτό ονομάζεται **αντικειμενοστραφής προγραμματισμός**. Η τεχνική αυτή δεν έχει αλλάξει μέχρι και σήμερα.

2. INTERNET KAI WORLD WIDE WEB

2.1. Η εξέλιξη του Internet

Το Internet αποτελείται από μερικά εκατομύρια διάσπαρτων σε όλον τον κόσμο υπολογιστών διασυνδεδεμένων μεταξύ τους με υψηλής χωρητικότητας γραμμές ή και δορυφορικές συνδέσεις. Ξεκίνησε στα τέλη της δεκαετίας του '60 ως ένα πείραμα του Αμερικάνικου στρατού, ο οποίος ήθελε να φτιάξει ένα ισχυρότατο τηλεπικοινωνιακό δίκτυο, το οποίο δε θα είχε κέντρο, αλλά κάθε κόμβος θα αποτελούσε ένα αυτόνομο κέντρο. Το δίκτυο άρχισε να λειτουργεί πειραματικά το 1969 με εννιά κόμβους, που συνέδεε αντίστοιχο αριθμό ερευνητικών κέντρων, και γρήγορα εξαπλώθηκε σε όλη την Αμερικάνικη ήπειρο. Το 1973 το επονομαζόμενο **ARPANET** απέκτησε κόμβους στη Βρετανία και τη Νορβηγία. Παράλληλα όμως οι χρήστες του, στη συντριπτική τους πλειοψηφία επιστήμονες, άρχισαν να βρίσκουν άλλες χρήσεις του δικτύου πέρα από την αρχική του. Μία από αυτές ήταν η ανταλλαγή απόψεων για θέματα άσχετα της επιστήμης μέσω του ηλεκτρονικού ταχυδρομείου. Η πρακτική αυτή άρχισε να εξαπλώνεται και το δίκτυο να αποκτά όλο και πιο “κοσμικό χαρακτήρα”. Από το 1983 το ARPANET γίνεται σιγά σιγά Internet και παίρνει τη μορφή με την οποία είναι πλέον γνωστό σε όλους. Το παγκόσμιο

δίκτυο τείνει να γίνει το Μέσο των Μέσων Επικοινωνίας. Μπορεί κανείς να διαβάσει εφημερίδες, περιοδικά και βιβλία μέσα από αυτό. Μπορεί να δει video, να στείλει και να πάρει ηλεκτρονικά γράμματα. Μπορεί επίσης να δει και να παραγγείλει διάφορα προϊόντα. Μπορεί ακόμα να τηλεφωνήσει σε οποιοδήποτε μέρος του κόσμου. Όλες αυτές οι λειτουργίες γίνονταν μέχρι σήμερα από διαφορετικά μέσα, εργαλεία και φορείς. Σήμερα ένας υπολογιστής με πρόσβαση στο Internet μπορεί να τις υποκαταστήσει μερικώς και ίσως στο μέλλον να τις αποκαταστήσει πλήρως, μιας και η απόδοση σε όλες τις λειτουργίες βελτιώνεται καθημερινά..

2.2. Οι υπηρεσίες του Internet

Οι υπηρεσίες που προσφέρει το Internet στους χρήστες του έχουν προς το παρόν διαχωρισθεί σε έντεκα κατηγορίες, τις ακόλουθες:

- **Ηλεκτρονικό Ταχυδρομείο (Electronic mail):** Μια υπηρεσία που διαφέρει ελάχιστα από τη συμβατική διακίνηση επιστολών. Το μόνο που είναι απαραίτητο είναι μια ηλεκτρονική διεύθυνση – μοναδική για κάθε χρήστη συνδεδεμένο στο Internet – και έτσι μπορεί να γίνει η αποστολή και λήψη μηνυμάτων σε ελάχιστα δευτερόλεπτα καλύπτοντας τεράστιες χιλιομετρικές αποστάσεις.
- **Usenet Newsgroups:** Ένα είδος ηλεκτρονικής διάσκεψης με περισσότερα από 10.000 θέματα, που περιλαμβάνει groups αφιερωμένα στην επιστήμη, στη διασκέδαση, στον πολιτισμό, στη φιλοσοφία κ.ο.κ. Εδώ ο καθένας μπορεί να διαβάσει ένα οποιοδήποτε θέμα και να προσθέσει τα δικά του σχόλια..
- **Mailing lists:** Λίστες χρηστών στους οποίους στέλνονται ομαδικά μηνύματα σχετικά με κάποιο συγκεκριμένο θέμα..
- **Πρωτόκολλο Μεταφοράς Αρχείων (FTP):** Ο πλέον διαδεδομένος τρόπος μεταφοράς αρχείων στο Internet. Ο χρήστης, καλώντας ένα από τα χιλιάδες sites ανά τον κόσμο που υποστηρίζει ο anonymous ftp, είναι σε θέση να κατεβάσει στον υπολογιστή Terabytes αρχείων.
- **Archie:** Μια τεράστια βάση δεδομένων όπου καταχωρούνται τα περιεχόμενα των μεγαλύτερων και δημοφιλέστερων FTP sites και η οποία επιτρέπει σε οποιονδήποτε χρήστη να εκτελέσει αναζητήσεις με βάση κάποια έκφραση αναζήτησης.

- **Telnet:** Μια από τις ουσιαστικότερες υπηρεσίες του διαδικτύου, που επιτρέπει τη σύνδεση με έναν απομακρυσμένο υπολογιστή τον οποίο ο χρήστης μπορεί να χρησιμοποιήσει να έχει συνδεθεί άμεσα με αυτόν. Έτσι έχει απευθείας πρόσβαση σε εκατομμύρια αρχεία και άμεση υπολογιστική ισχύ.
- **Internet Relay Chat (IRC):** Ένα εργαλείο που επιτρέπει τη συνομιλία των ανθρώπων από όλο τον κόσμο για ανταλλαγή απόψεων, αλλά κυρίως για διασκέδαση.
- **Gopher:** Ένα client – server σύστημα χαρακτηριστικό για τη μοναδικότητά του να επιτρέπει στο χρήστη την περιήγηση μέσα στο Internet με τη βοήθεια φιλικών menu. Έτσι, ο χρήστης έχει τη δυνατότητα να συναντήσει πάρα πολλά θέματα και να διαβάσει, να μεταφέρει και να αποθηκεύσει όποιο αρχείο επιθυμεί.
- **Veronica:** Ένα πρόγραμμα το οποίο δεν είναι standalone, αλλά συναντάται ως επιλογή σε πολλούς από τους gopher servers. Χρησιμοποιείται για την εύρεση της απαιτούμενης πληροφορίας σε οποιονδήποτε Gopher server και για τη διαπίστωση του αν αυτή είναι αποθηκευμένη.
- **WAIS (Wide Areas Information Servers):** Ένα ακόμα εργαλείο για αναζήτηση του πραγματικού κειμένου ενός αρχείου ή πλήθους αρχείων. Λειτουργεί με επαναληπτικές αφαιρέσεις παραγράφων κειμένου από αρχεία λιστών μέχρις ότου να πλησιάσει αρκετά στην επιλογή του χρήστη.
- **World Wide Web (WWW):** Η πιο διαδεδομένη και φιλική προς το χρήστη υπηρεσία του δικτύου, η οποία χρησιμοποιεί hypertext (υπερκείμενα), που είναι συνδεδεμένα μέσω links (συνδεσμών) με τα μεγαλύτερα sites του εξωτερικού και έτσι μπορούν να κάνουν άμεση τη ροή πληροφοριών ανάμεσα σε απομακρυσμένους υπολογιστές.

2.3. Το World Wide Web (WWW) – Ο Παγκόσμιος Ιστός

Είναι σίγουρο ότι αν δεν υπήρχε το World Wide Web (στο εξής θα αναφέρεται ως WWW ή παγκόσμιος Ιστός), το Internet δε θα γνώριζε ποτέ την τεράστια ανάπτυξη που έχει τα τελευταία χρόνια. Αυτή τη στιγμή υπάρχει ένα τεράστιο ποσό πληροφορίας άμεσα προσπελάσιμο μέσω αυτού. Η δύναμη του WWW έγκειται στο γεγονός ότι η άντληση πληροφοριών και η περιήγηση στο δίκτυο επιτυγχάνεται με

ένα απλό ‘πάτημα’ του ποντικιού. Επίσης, τα εργαλεία αναζήτησης πληροφοριών, που είναι μέρος του WWW, μετά από την επιλογή των κατάλληλων λέξεων κλειδιά (keywords) από το χρήστη κάνουν την άντληση πληροφοριών ακόμη πιο ανθρωποκεντρική. Και όλα τα παραπάνω πραγματοποιούνται μέσα από εντυπωσιακά και υψηλής αισθητικής πολλές φορές γραφικά περιβάλλοντα που κάνουν το μέσο χρήστη να μην αισθάνεται φόβο να περιηγηθεί στο δίκτυο. Η κεντρική ιδέα βασίζεται στο ότι οι πληροφορίες είναι δομημένες ή ενσωματωμένες σε hypertext κείμενα (υπερκείμενα). Τα hypertext κείμενα είναι απλά αρχεία κειμένου (δηλαδή αρχεία ASCII) που περιέχουν εκτός από απλό κείμενο και hypertext links, δηλαδή συνδέσμους που οδηγούν σε άλλα αρχεία hypertext κειμένου τα οποία με τη σειρά τους οδηγούν μέσω άλλων hypertext links σε άλλα hypertext κείμενα, κ.ο.κ. Γίνεται λοιπόν εύκολα κατανοητό γιατί το παγκόσμιο πρότυπο αυτό της οργάνωσης πληροφοριών ονομάστηκε World Wide Web, δηλαδή Παγκόσμιος Ιστός. Τα hypertext αρχεία που είναι δομικό στοιχείο του WWW δημιουργούνται μέσω κατάλληλης γλώσσας μορφοποίησης κειμένου, που ονομάζεται **HTML** (Hypertext Markup Language – Γλώσσα Χαρακτηρισμού Σήμανσης Υπερκειμένου). Το μεγαλύτερο πλεονέκτημα του WWW είναι ότι βασιζόμενο στο πρωτόκολλο HTTP και έξυπνες εφαρμογές Server και Client, πέρα από απλά κείμενα, μπορεί να παρέχει εικόνες, ήχο ακόμα και video. Οι multimedia δυνατότητες λοιπόν, παρά τις δυσκολίες που υπάρχουν λόγω των χαμηλών ταχυτήτων των δικτύων ευρείας περιοχής, το κάνουν ιδανικό και για άλλες εφαρμογές που απαιτούν παροχή σύνθετων πληροφοριών όπως π.χ. στο χώρο της εκπαίδευσης, του τουρισμού κ.ά.

2.4. Η αρχιτεκτονική πελάτη – εξυπηρετητή (client – server) και το WWW

Το WWW είναι στην πραγματικότητα ένας συνδυασμός – μια συνεργασία – δύο προγραμμάτων που έχουν αρχιτεκτονική πελάτη – εξυπηρετητή. Το πρόγραμμα που είναι πελάτης (client) κάτω από τις οδηγίες του χρήστη ζητάει από τον εξυπηρετητή (server) κάποιο υπερκείμενο (hypertext), δηλαδή κάποιο URL (Uniform Resource Locator – Μοναδιαία Θέση Πόρου). Ο server που είναι υπεύθυνος για το URL που ζητήθηκε, μεταδίδει την όλη πληροφορία στο client πρόγραμμα μέσω του δικτύου και το client πρόγραμμα είναι υπεύθυνο για την απεικόνιση της πληροφορίας αυτής.

Η όλη αυτή διαδικασία ακολουθεί κάποιος συγκεκριμένους κανόνες, ένα πρωτοκόλλο. Είναι σύμφωνη με το Hypertext Transfer Protocol (HTTP). Το client πρόγραμμα είναι συνήθως εγκατεστημένο στον προσωπικό υπολογιστή του χρήστη και λέγεται Web Browser ή Web Client ενώ το πρόγραμμα που παίζει το ρόλο του server είναι εγκατεστημένο στους διάφορους υπολογιστές του δικτύου και λέγεται Web Server ή Httpd (Hypertext Transfer Protocol Daemon). Υπάρχουν περιπτώσεις που ο Web Client και ο Web Server είναι εγκατεστημένοι στο ίδιο μηχάνημα.

2.5. Uniform Resource Locator (URL) – Ομοιόμορφος Εντοπιστής Πόρου

Το URL είναι ένα πρότυπο που χρησιμοποιείται στο WWW για τη σύνταξη της διεύθυνσης κάθε αντικειμένου στο Internet. Κάθε hypertext link σε ένα hypertext κείμενο αντιστοιχεί σε κάποιο URL και κάθε αντικείμενο – πληροφορία στο Internet έχει τουλάχιστον ένα URL. Μπορεί να είναι κάποιο απλό αρχείο hypertext στατική πληροφορία) ή ακόμα και το αποτέλεσμα μιας αναζήτησης σε κάποια βάση δεδομένων (δυναμική πληροφορία). Ακολουθώντας το πρότυπο αυτό, το WWW ‘ενοποιεί’ (στην πραγματικότητα οργανώνει, συνδέει) όλες τις άλλες υπηρεσίες του Internet όπως Gopher, Mail, Ftp, Telnet και News. Το URL περιέχει πληροφορίες για την Internet διεύθυνση του υπολογιστή που παρέχει την πληροφορία, τον τύπο της υπηρεσίας (δηλαδή το είδος του πρωτοκόλλου) από την οποία εξυπηρετείται η πληροφορία (π.χ. Http, gopher κ.ο.κ.), τη θέση του αντικειμένου και το Internet port. Η γενική σύνταξη κάποιου URL έχει τη γενική μορφή:

service://host:port/file_location, όπου

- service → Κάποιο από τα γνωστά πρωτόκολλα – προγράμματα του Internet όπως http, ftp, gopher, mail, telnet, news, wais και mail.
- host → Η Internet διεύθυνση του υπολογιστή που είναι υπεύθυνο για το συγκεκριμένο URL.
- port → Η Internet πόρτα. Είναι υποχρεωτικό εάν το URL δεν αντιστοιχεί στην προκαθορισμένη (default) πόρτα σύμφωνα με το πρωτόκολλό του (συνήθως υπάρχει σε telnet URLs).
- file_location → Η θέση στην οποία βρίσκεται το συγκεκριμένο αρχείο.

Στον παρακάτω πίνακα παρατίθενται παραδείγματα όλων των ειδών URLs:

Είδος URL	Γενική Μορφή Σύνταξης	Παράδειγμα
http URL	http://host:port/file_location	http://www.uom.gr/
gopher URL	gopher://host:port/file_location	gopher://gopher.forthnet.gr:70/
ftp URL	ftp://host:port/file_location	ftp://ftp.uom.gr/
news URL	news:news.group	news:comp.os.linux.misc
mail URL	mailto:login_name@host	mailto:noc@uom.gr

Είναι προφανές ότι τα mail και τα news URLs δεν ακολουθούν όσον αφορά τη σύνταξή τους τη γενική μορφή σύνταξης. Θα πρέπει επίσης να σημειωθεί ότι υπάρχουν σπάνιες περιπτώσεις που hypertext κείμενα γίνονται αντικείμενο διαχείρισης και από το ftp.

2.6. Οι World Wide Web Browsers

Οι World Wide Web Browsers είναι λογισμικά ειδικού τύπου που αναγνωρίζουν και απεικονίζουν έγγραφα γραμμένα στη γλώσσα HTML. Επίσης μπορούν να απεικονίζουν HTML φόρμες οι οποίες επιτρέπουν την εισαγωγή δεδομένων από τους χρήστες. Οι browsers χρησιμοποιώντας αυτές τις φόρμες μπορούν να συλλέγουν και να απεικονίζουν πληροφορίες. Οι WWW Browsers μπορούν να προσπελάσουν πολλά υπάρχοντα συστήματα δεδομένων, μέσω υπαρχόντων πρωτοκόλλων (FTP, NNTP) ή μέσω του HTTP και μιας πύλης (gateway). Μ' αυτόν τον τρόπο, η κρίσιμη μάζα της πληροφορίας επεκτείνεται γρήγορα και η αυξανόμενη χρήση του συστήματος, έχει ως αποτέλεσμα την παροχή όλο και περισσότερης πληροφορίας. Το πρωτόκολλο HTTP επεκτείνεται συνεχώς για να αποκτήσει ακόμα περισσότερες δυνατότητες. Όπου υπάρχουν χρησιμοποιούνται κάποια πρότυπα, όπως είναι το **SGML** (Standard Generalized Markup Language) για το μοντέλο του Hypertext ή το **MIME** (Multipurpose Internet Mail Extensions) για την καταχώρηση των τύπων των αρχείων.

2.7. Ορισμός της HTML

Είναι μια γλώσσα κατανοητή από όλους του υπολογιστές παγκοσμίως, η οποία χρησιμοποιείται για 'δημοσίευση' (κοινοποίηση) πληροφορίας σε παγκόσμιο επίπεδο. Η HTML δίνει τη δυνατότητα στους συγγραφείς των ιστοσελίδων να:

- Δημοσιεύουν ηλεκτρονικά έγγραφα με επικεφαλίδες, κείμενο, πίνακες, λίστες, φωτογραφίες, κτλ.
- Ανακτούν ηλεκτρονική πληροφορία μέσω συνδέσμων υπερκειμένων (hypertext links) με το πάτημα ενός πλήκτρου.
- Σχεδιάζουν φόρμες για να διαχειρίζονται συναλλαγές με απομακρυσμένες υπηρεσίες, για να βρίσκουν κάποια πληροφορία, για να κάνουν κρατήσεις, για να παραγγείλουν ένα προϊόν, κτλ.
- Συμπεριλαμβάνουν λογιστικά φύλλα, clips ήχου και εικόνες καθώς και ποικίλες άλλες εφαρμογές απευθείας στα έγγραφά τους.

2.8. Τμήματα ενός HTML εγγράφου

Ένα έγγραφο HTML αποτελείται από 3 μέρη:

- a) Από μια γραμμή που περιέχει πληροφορία σχετικά με την έκδοση της HTML.
- b) Από μια επεξηγηματική επικεφαλίδα (η οποία βρίσκεται μέσα στο HEAD). Το HEAD περιλαμβάνει πληροφορία σχετικά με το συγκεκριμένο κείμενο, όπως είναι ο τίτλος, λέξεις κλειδιά που χρειάζονται στις μηχανές αναζήτησης και άλλα δεδομένα που δε θεωρούνται περιεχόμενο του εγγράφου. Συγκεκριμένα, συντάσσεται ως εξής: `<HEAD>...</HEAD>`.
- c) Το κυρίως μέρος (σώμα), το οποίο περιλαμβάνει το πραγματικό περιεχόμενο του εγγράφου. Αποτελείται είτε από το στοιχείο BODY (π.χ. `<BODY>...</BODY>`), είτε από το στοιχείο FRAMESET.

Το κυρίως μέρος του εγγράφου περιέχει το περιεχόμενο του εγγράφου, το οποίο μπορεί να παρουσιαστεί με ποικίλους τρόπους, όπως π.χ. κείμενο, εικόνες, χρώματα και γραφικά. Το FRAMESET καθορίζει το 'στήσιμο' (σελιδοποίηση) του βασικού παραθύρου σε μορφή ορθογωνίων σχημάτων.

3. ΙΣΤΟΣΕΛΙΔΕΣ ΔΥΝΑΜΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ

3.1. Η εξέλιξη των ιστοσελιδών δυναμικού περιεχομένου

Από την στιγμή της εμφάνισης της υπηρεσίας WWW δημιουργήθηκε η ανάγκη για αλληλεπίδραση των ιστοσελιδών με τους χρήστες. Αρχικά τα περιεχόμενα των σελιδών ήταν στατικά. Ο χρήστης μπορούσε να δει τις πληροφορίες που περιείχε μια σελίδα, τα οποία όμως για να αλλάξουν απαιτούσαν από τον υπεύθυνο του site να ξαναγράψει την σελίδα με ανανεωμένο περιεχόμενο. Αυτό δεν ήταν καθόλου πρακτικό, αφού κατά κύριο λόγο το κάθε site χρειάζεται ανανέωση μόνο στα περιεχόμενά του και όχι στο εικαστικό του μέρος. Έπρεπε δηλαδή να βρεθεί ένας τρόπος κατά τον οποίο τα περιεχόμενα των ιστοσελιδών θα γινόταν δυναμικά, ανακτώμενα π.χ. από ένα σύστημα βάσεως δεδομένων, όπου θα ήταν πιο εύκολο για τον συντηρητή του site να τα μεταβάλει όποτε αυτό είναι αναγκαίο. Τέτοια παραδείγματα είναι site όπου δημοσιεύουν ειδήσεις ή τα προϊόντα ενός καταστήματος. Είναι προφανές ότι τα δεδομένα αυτά αλλάζουν διαρκώς και έτσι οι σελίδες που τα φιλοξενούν θα πρέπει να είναι σε θέση να τα ανακτούν όποτε αυτά αλλάζουν, χωρίς να είναι απαραίτητη καμία αλλαγή στα ίδια τα αρχεία των ιστοσελιδών. Επιπλέον δεν δινόταν η δυνατότητα στον χρήστη να αλληλεπιδρά με ένα Web Site. Για παράδειγμα να μπορεί ο χρήστης να αποστέλει στοιχεία του (π.χ. για την δημιουργία μιας online ατζέντας, παραμετροποιήσιμη για κάθε χρήστη) ή να αιτείται υπηρεσίες (όπως αγοραπωλησία προϊόντων). Έτσι δημιουργήθηκαν Web Servers με δυνατότητες επέκτασης των δυνατοτήτων τους. Αρχικά αυτό γινόταν μόνο με την βοήθεια script, που έγραφαν προγραμματιστές και έκαναν λειτουργίες που αφορούσαν την συγκεκριμένη σελίδα για την οποία γράφτηκαν (π.χ. αποθήκευση στοιχείων που έστειλε ένας χρήστης σε μία βάση δεδομένων). Αυτή η τεχνολογία ονομάζεται **Common Gateway Interface** και θα την αναλύσουμε πιο κάτω. Στην συνέχεια τα scripts εξελίχθηκαν σε γλώσσες προγραμματισμού, όπου η κάθε σελίδα πλέον περιέχει κώδικα που εκτελεί ο Web Server πριν αποστείλει την σελίδα στον χρήστη. Έτσι οι δυνατότητες των σημερινών ιστοσελιδών είναι πραγματικά απεριόριστες και μπορούν να δώσουν λύση σε οποιαδήποτε ανάγκη παρουσιάζεται. Αυτό άνοιξε το δρόμο για πραγματικά δυναμικό περιεχόμενο στα site και

επεξεργασία των αιτήσεων ενός χρήστη γρήγορα και εύκολα. Πλέον τα site μπορούν να παραμετροποιούνται κατά την επιθυμία ενός χρήστη, να ανανεώνονται συνεχώς και ανοίγουν νέους ορίζοντες για το ηλεκτρονικό εμπόριο και επιχειρείν. Παραδείγματα τέτοιων γλωσσών προγραμματισμού είναι η PHP και η εφάμιλλη ASP, καθώς και οι πιο εξελιγμένες JSP και ASP .NET.

3.2. Το CGI (Common Gateway Interface – Κοινή Διασύνδεση Πύλης)

Στο ακόλουθο τμήμα γίνεται εκτενής αναφορά στο CGI. Ας ξεκινήσουμε από το τι σημαίνουν τα αρχικά του :

- **Common:** Υποδηλώνει ότι το CGI μπορεί να χρησιμοποιηθεί από πολλές γλώσσες και να αλληλεπιδράσει με πολλά διαφορετικά συστήματα.
- **Gateway:** Υποδηλώνει ότι η δύναμη του CGI δεν βρίσκεται στο τί μπορεί να κάνει αυτό από μόνο του, αλλά στην ενδεχόμενη πρόσβαση που προσφέρει σε άλλα συστήματα (πόρους του HTTP server), όπως βάσεις δεδομένων, φόρμες για feedback clickable ή image maps. Είναι δηλαδή μια ‘πύλη’ ανάμεσα στο Web και στα συστήματα αυτά. Είναι συνήθως κάποιο πρόγραμμα το οποίο διαχειρίζεται κλήσεις για πληροφορία και επιστρέφει το ζητούμενο έγγραφο ή δημιουργεί ένα έγγραφο στη στιγμή.
- **Interface:** Σημαίνει ότι κάποιος μπορεί να γράψει προγράμματα τα οποία το χρησιμοποιούν.

Τα CGI scripts μπορούν να γραφούν σε γλώσσες όπως οι: Perl, C/C++, TCL, Python, Pascal, Basic ή ακόμα και Unix shell scripts. Επειδή τα CGI προγράμματα είναι εκτελέσιμα, η όλη διαδικασία εκτέλεσης ενός τέτοιου προγράμματος ισοδυναμεί με το να έχει την πρόσβαση ο οποιοσδήποτε να εκτελέσει ένα πρόγραμμα στο σύστημα (server), το οποίο προφανώς δεν είναι καθόλου ασφαλές. Γι’ αυτό το λόγο, τοποθετούνται συνήθως τα CGI προγράμματα σε έναν ειδικό κατάλογο (το αποκαλούμενο cgi-bin), έτσι ώστε ο Web server να ξέρει ότι πρέπει να τα εκτελέσει και όχι απλά να τα εμφανίσει στον browser. Αυτός ο κατάλογος είναι συνήθως υπό την απόλυτη εποπτεία του webmaster, ο οποίος δεν επιτρέπει στο μέσο χρήστη να δημιουργεί CGI προγράμματα. Στη συνέχεια δίνονται μερικά στοιχεία απαραίτητα

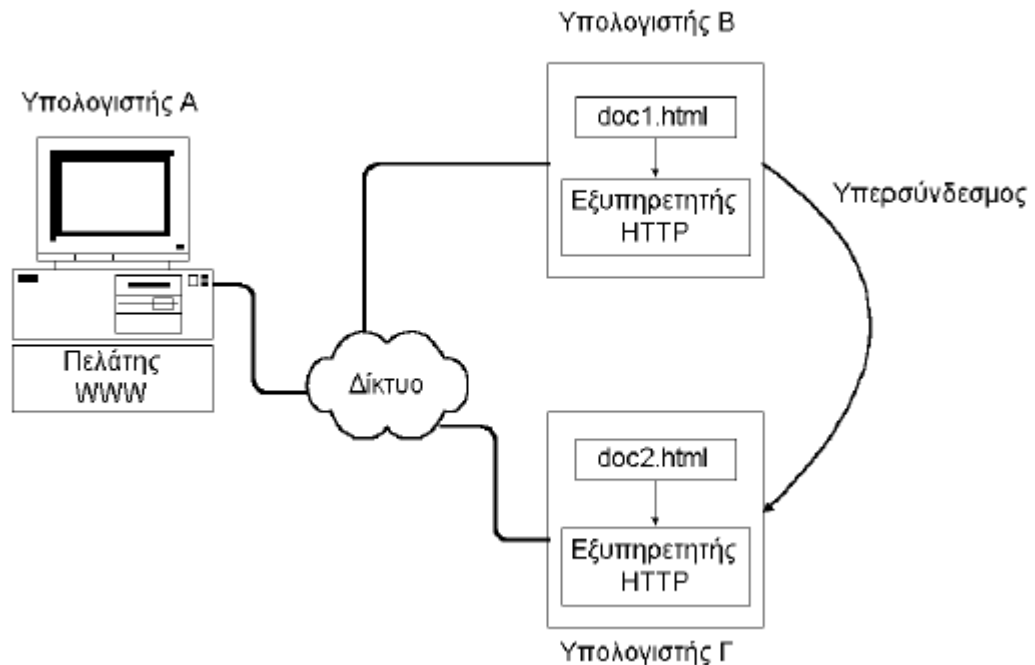
για την κατανόηση της όλης διαδικασίας που απαιτείται για τη διασύνδεση μιας βάσης δεδομένων στο Internet μέσω της μεθόδου του CGI.

3.3. Η λειτουργία του CGI

Οι World Wide Web browsers, όπως άλλωστε προαναφέρθηκε, απεικονίζουν έγγραφα γραμμένα σε γλώσσα HTML. Επίσης μπορούν να απεικονίζουν HTML φόρμες οι οποίες επιτρέπουν την εισαγωγή δεδομένων από τους χρήστες. Οι browsers χρησιμοποιώντας αυτές τις φόρμες μπορούν να συλλέγουν καθώς και να απεικονίζουν πληροφορίες. Όταν η πληροφορία συλλέγεται από τον browser, στέλνεται σ' έναν HTTP server, ο οποίος καθορίζεται από την HTML форма και επεξεργάζεται την πληροφορία. Αυτή η λειτουργία επιτελούν τα CGI scripts. Παρακάτω θα δείξουμε τους τρόπους με τους οποίους μια φόρμα, ένας client browser, ένας server και το HTTP Protocol συνεργάζονται επιτυχώς. Για να γίνει κατανοητή αυτή η πολύπλοκη αλληλεπίδραση πρέπει πρώτα να κατανοηθεί πώς ένας client και ένας server μπορούν να συνεργαστούν για τη μεταφορά ενός απλού HTML εγγράφου. Αυτή είναι η “φυσιολογική” δραστηριότητα του Web, ή συνήθους λειτουργία του. Μετά από αυτό πρέπει να γίνει κατανοητό πώς εκτελούνται τα scripts σε περιβάλλον δικτύου χωρίς τη μεσολάβηση φορμών. Μόλις κατανοηθούν αυτές οι δύο εργασίες, η διασύνδεση φορμών είναι ξεκάθαρη.

3.4. Η απλή αλληλεπίδραση Εξερευνητή – Εξυπηρετητή (Browser – Server)

Κατά τη διάρκεια ανταλλαγής ενός κανονικού εγγράφου ένας WWW client (π.χ. Netscape Navigator, Internet Explorer, Mosaic) ζητάει ένα έγγραφο από τον WWW server και το απεικονίζει σε μια συσκευή απεικόνισης του χρήστη. Εάν αυτό το έγγραφο περιέχει σύνδεσμο (link) σε άλλο έγγραφο και ο χρήστης το ενεργοποιήσει, τότε ο WWW client θα το φέρει και θα το απεικονίσει. Το παρακάτω διάγραμμα δείχνει την αλληλεπίδραση ενός WWW πελάτη ο οποίος τρέχει σε ένα προσωπικό υπολογιστή (Υπολογιστής Α), με δύο άλλους εξυπηρετητές: έναν HTTP ο οποίος τρέχει στον Υπολογιστή Β και έναν εξυπηρετητή HTTP που τρέχει στον Υπολογιστή Γ.

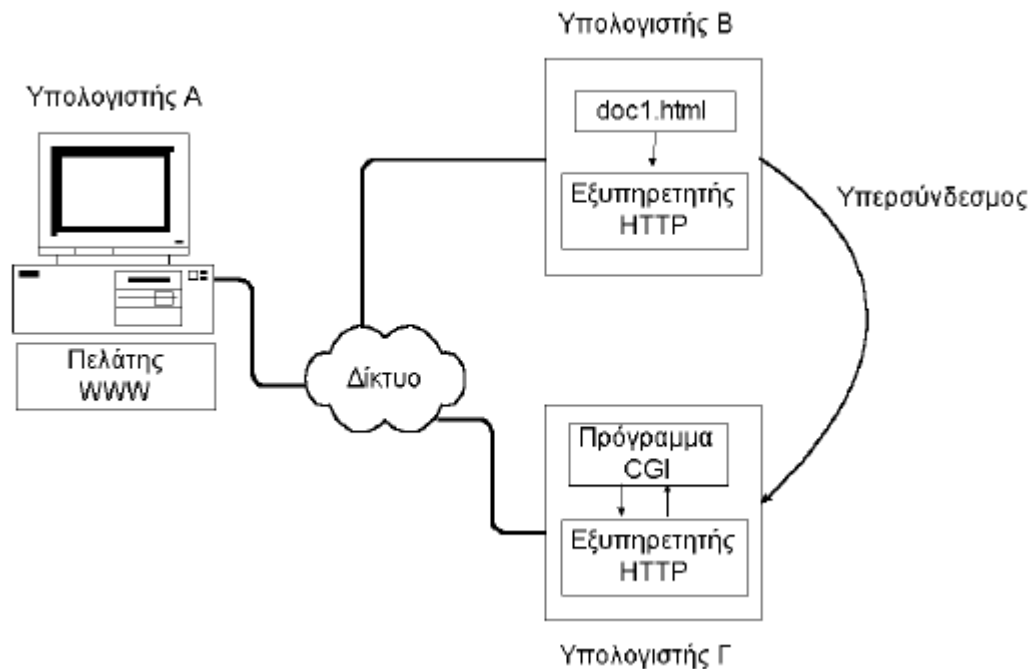


Ο WWW πελάτης, ο οποίος εκτελείται στον Υπολογιστή Α, παίρνει ένα έγγραφο το οποίο είναι αποθηκευμένο με το όνομα doc1.html στον εξυπηρετητή HTTP, ο οποίος εκτελείται στον Υπολογιστή Β. Αυτό το έγγραφο περιέχει έναν υπερσύνδεσμο σε ένα άλλο έγγραφο, το οποίο βρίσκεται αποθηκευμένο στον Υπολογιστή Γ με το όνομα doc2.html. Το URL του μπορεί να μοιάζει με το ακόλουθο: `http://ΥπολογιστήςΓ.domain/ doc2.html`. Εάν ο χρήστης ενεργοποιήσει αυτόν τον υπερσύνδεσμο, τότε ο πελάτης ανακτά το αρχείο από τον εξυπηρετητή HTTP ο οποίος εκτελείται στον Υπολογιστή Γ και το απεικονίζει στην οθόνη η οποία είναι συνδεδεμένη με τον Υπολογιστή Α. Το πρωτόκολλο HTTP είναι αυτό που θα καθορίζει την επικοινωνία μεταξύ του πελάτη και ενός εξυπηρετητή HTTP.

3.5. Η εκτέλεση των CGI προγραμμάτων στον εξυπηρετητή

Ένα HTTP URL μπορεί να ξεχωρίσει αν ένα αρχείο περιέχει ένα πρόγραμμα από ένα HTML έγγραφο. Αυτό το πρόγραμμα μπορεί να εκτελεστεί όταν ο χρήστης ενεργοποιήσει τον υπερσύνδεσμο στον οποίο περιέχεται το URL. Το παρακάτω διάγραμμα, δείχνει ένα υπερκείμενο στον Υπολογιστή Β με ένα υπερσύνδεσμο σε ένα αρχείο στον υπολογιστή Γ, το οποίο περιέχει το CGI πρόγραμμα το οποίο θα εκτελεστεί αν ο χρήστης ενεργοποιήσει τον υπερσύνδεσμο. Αυτό ο υπερσύνδεσμος

είναι ένας κανονικός http: υπερσύνδεσμος, αλλά το αρχείο είναι αποθηκευμένο με τέτοιο τρόπο ώστε ο εξυπηρετητής HTTP στον Υπολογιστή Γ να μπορεί να αποφανθεί για το ότι το αρχείο περιέχει ένα πρόγραμμα που πρόκειται να εκτελεστεί και όχι ένα έγγραφο το οποίο πρόκειται να σταλεί στον πελάτη όπως συνήθως.

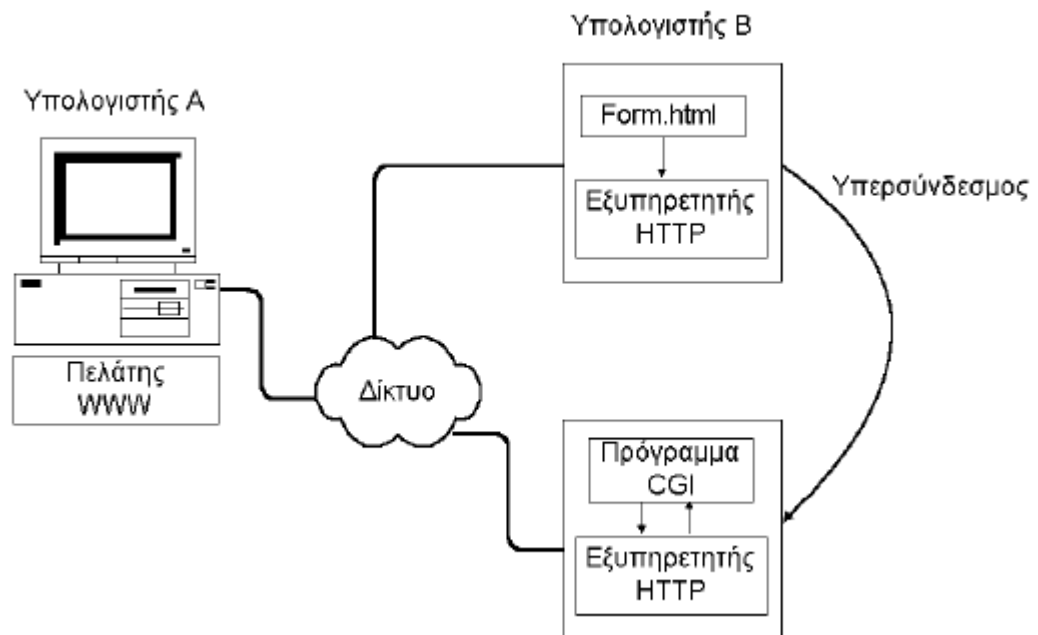


Όταν το πρόγραμμα εκτελείται, προετοιμάζει ένα HTML έγγραφο που περιέχει δυναμική πληροφορία και στέλνει αυτό το έγγραφο στον πελάτη, ο οποίος τα απεικονίζει όπως ακριβώς θα έκανε με κάθε άλλο HTML έγγραφο.

3.6. Η εκτέλεση ενός CGI προγράμματος μέσω μιας HTML φόρμας

Η ικανότητα επεξεργασίας φορμών με δεδομένα στο Internet απαιτεί αλλαγές σε HTML, Web πελάτες και Web εξυπηρετητές (τελικά ακόμα και στο HTTP). Ένα σύνολο ετικετών προστέθηκαν στην HTML, με σκοπό να κατευθύνουν έναν WWW πελάτη στην απεικόνιση μιας φόρμας προς συμπλήρωση από τον χρήστη και μετά να προωθήσουν τα συγκεντρωμένα δεδομένα σε ένα HTTP πελάτη, ο οποίος περιγράφεται στη φόρμα. Οι εξυπηρετητές έχουν υποστεί μετατροπές ώστε να μπορούν να ξεκινήσουν αμέσως το CGI πρόγραμμα που καθορίζεται στη φόρμα και να προωθήσουν τα συγκεντρωμένα δεδομένα στο πρόγραμμα, το οποίο μπορεί να

ετοιμάσει μια απάντηση (πιθανότατα με το να ανατρέξει σε μια υπάρχουσα βάση δεδομένων) και να επιστρέψει ένα WWW έγγραφο στο χρήστη.



Το σχήμα δείχνει τα διάφορα στοιχεία της επεξεργασίας. Σε αυτό το διάγραμμα ο Web πελάτης, ο οποίος τρέχει στον Υπολογιστή Α, παίρνει μια φόρμα από κάποιο Web εξυπηρετητή που τρέχει στον Υπολογιστή Β. Εμφανίζει τη φόρμα, ο χρήστης εισάγει δεδομένα και ο πελάτης στέλνει τις εισαχθείσες πληροφορίες στον HTTP εξυπηρετητή ο οποίος εκτελείται στον Υπολογιστή Γ. Εκεί, τα δεδομένα μεταβιβάζονται σε ένα CGI πρόγραμμα το οποίο προετοιμάζει ένα έγγραφο και το στέλνει στον πελάτη στον υπολογιστή Α. Ο πελάτης τότε εμφανίζει αυτό το έγγραφο.

3.7. Μέθοδοι κλήσης προς τον εξυπηρετητή

Προκειμένου να εκτελεστεί ένα CGI πρόγραμμα γίνεται μια κλήση στον εξυπηρετητή με τη χρήση μιας μεθόδου, η οποία καθορίζει τον τρόπο με τον οποίο το πρόγραμμα θα λάβει τα δεδομένα. Υπάρχουν τρεις μέθοδοι, η POST, η GET και HEAD, για τις οποίες γίνεται εκτενής αναφορά ευθύς αμέσως:

- **Η μέθοδος POST.** Όταν χρησιμοποιείται η μέθοδος POST ο Web εξυπηρετητής μεταδίδει τα δεδομένα στο CGI πρόγραμμα μέσα από το stdin (standard input).

Συγκεκριμένα, τα δεδομένα στέλνονται ως ξεχωριστό message body το οποίο ακολουθεί την κλήση που στέλνει ο πελάτης στον εξυπηρετητή. Το ξεχωριστό αυτό μήνυμα με το οποίο στέλνονται τα δεδομένα πηγαίνει στο CGI script σε μορφή κανονικού input. Ο εξυπηρετητής δε μαρκάρει το τέλος των δεδομένων με ένα χαρακτήρα EOF κι έτσι το πρόγραμμα πρέπει να χρησιμοποιήσει την τιμή της μεταβλητής περιβάλλοντος CONTENT_LENGTH για να διαβάσει σωστά το stdin. Η POST χρησιμοποιείται συνήθως όταν τα δεδομένα που αποστέλλονται θα αλλάξουν κάποια από τα δεδομένα στον Web εξυπηρετητή ή όταν αποστέλλονται μεγάλες ποσότητες δεδομένων στο CGI πρόγραμμα (συνήθως από 1024 bytes που είναι το ανώτατο όριο μήκους ενός URL).

- **Η μέθοδος GET.** Με αυτή τη μέθοδο τα δεδομένα περνιούνται μέσα στο query string του URL. Αν προσθέσει κανείς ένα ερωτηματικό (?) στο URL του script, τότε όλοι οι χαρακτήρες που ακολουθούν το ερωτηματικό θα περάσουν στο script στη μεταβλητή περιβάλλοντος που ονομάζεται QUERY_STRING. Αν το script ονομάζεται script.cgi και ο χρήστης κάνει ένα υπερσύνδεσμο στο script αυτό (ή γράψει στον εξερευνητή) τότε το string “name1=value1&name2=value2” θα είναι το QUERY_STRING. Το script, μπορεί να διαβάσει αυτήν τη μεταβλητή και να την επεξεργαστεί όπως θέλει ο χρήστης. Με τη χρήση του QUERY_STRING στο URL ενός υπερσυνδέσμου, μπορεί κανείς να εισάγει δεδομένα σε ένα script χωρίς να έχει φόρμα. Αυτό είναι βολικό αν κανείς θέλει να κάνει υπερσύνδεσμο σε ένα script, ή να εισάγει μικρή ποσότητα δεδομένων, χωρίς όμως να χρειαστεί να κατασκευάσει μια αναλυτική φόρμα ή να ζητά από το χρήστη να πατήσει ένα κουμπί υποβολής της φόρμας (τύπου “submit”). Επίσης η GET μέθοδος χρησιμοποιείται όταν ο χρήστης θέλει να πάρει δεδομένα από τον εξυπηρετητή και δεν πρόκειται να τα τροποποιήσει.:
`http://www.your_site.gr/script.cgi?name1=value1&name2=value2`
- **Η μέθοδος HEAD.** Είναι παρόμοια με τη μέθοδο GET με τη μόνη διαφορά ότι με τη μέθοδο αυτή στέλνονται από τον Web εξυπηρετητής στον εξερευνητή μόνο οι HTTP επικεφαλίδες και όχι πραγματικά δεδομένα. Γενικά, η GET θα έπρεπε να χρησιμοποιείται όταν η πρόσβαση σε ένα URL δεν αλλάζει την κατάσταση της βάσης δεδομένων (για παράδειγμα με την πρόσθεση ή αφαίρεση πληροφοριών) και η POST πρέπει να χρησιμοποιείται όταν μια πρόσβαση θα προκαλέσει αλλαγή. Ένα

πλεονέκτημα της GET είναι ολόκληρη η υποβολή της φόρμας μπορεί να ενσωματωθεί σε ένα URL, όπως ένας υπερσύνδεσμος. Υπάρχει ένας πρακτικός περιορισμός για την GET όμως, γιατί πολλά συστήματα έχουν όρια στο μήκος της GET request που μπορούν να χειριστούν. Όταν το συνολικό μήκος μιας request (URL+παράμετροι) πλησιάζει ή υπερβαίνει το 1 Kb, είναι προτιμότερο να χρησιμοποιήσει κανείς τη μέθοδο POST.

4. Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ScriptC

Αντικείμενο αυτής της διπλωματικής εργασίας είναι η ανάπτυξη μιας γλώσσας προγραμματισμού αντίστοιχης των PHP και ASP η οποία ονομάστηκε ScriptC. Είναι μία server side script γλώσσα η οποία δίνει την δυνατότητα δημιουργίας δυναμικών ιστοσελιδών. Τα χαρακτηριστικά της γλώσσας είναι αυτά των δύο προαναφερθέντων γλωσσών τα οποία και θα παρουσιάσουμε αναλυτικότερα :

4.1. Τεχνολογία και περιβάλλον ανάπτυξης

Η ScriptC δημιουργήθηκε σε C++ και συγκεκριμένα χρησιμοποιώντας το περιβάλλον Borland C++ Builder 5.0 για Windows. Είναι μια CGI εφαρμογή η οποία εγκαθίσταται στον Web Server. Όταν κληθεί (από τον Server) εκτελεί το δοθέν αρχείο σαν Interpreter, παράγοντας στην έξοδο το αποτέλεσμα κώδικα και HTML που βρέθηκε μέσα στο αρχείο. Τονίζεται στο σημείο αυτό ότι η ScriptC είναι ένας interpreter όπως ακριβώς οι PHP και ASP και δεν παράγει ενδιάμεσα αρχεία όπως λόγω χάρη η JSP και ASP .NET.

4.2. Διαδικασία εκτέλεσης ενός αρχείου

Όταν ένας πελάτης ζητήσει από τον Web Server ένα αρχείο, το οποίο έχει καθοριστεί να εκτελείται από την ScriptC, τότε ο Web Server εκκινεί την εφαρμογή. Η ScriptC κάνει parse το αρχείο και ελέγχει για την συντακτική ακεραιότητα του κώδικα που βρίσκεται μέσα στο αρχείο. Επειδή κώδικας και HTML βρίσκονται μαζί στο ίδιο αρχείο, χρησιμοποιούνται τα σύμβολα <% και %> για να σηματοδοτήσουν την

έναρξη και το τέλος ενός μπλοκ server side εντολών. Προγραμματιστές εξοικειωμένοι με PHP ή ASP γνωρίζουν την χρήση αντίστοιχων συμβόλων στις γλώσσες αυτές για τον σκοπό αυτό. Σε περίπτωση που ανιχνευθούν συντακτικά λάθη, η διαδικασία εκτέλεσης ακυρώνεται και αποστέλλεται μία πληροφοριακή σελίδα για τα λάθη που προέκυψαν. Σε αντίθετη περίπτωση, εκτελείται ο κώδικας και αφού ολοκληρωθεί η εκτέλεση αποστέλλεται το αποτέλεσμα της στον πελάτη (μέσω του Web Sever). Αναλυτικότερα η εκτέλεση χωρίζεται σε δύο φάσεις :

- **Φάση ανάλυσης :** Κατά την φάση της ανάλυσης, γίνεται συντακτικός έλεγχος στον κώδικα του προγράμματος. Τυχόν συντακτικά λάθη αναφέρονται όταν ολοκληρωθεί η διαδικασία. Στην διάρκεια της ανάλυσης, διατηρείται μία δομή δεδομένων οργανωμένη σαν δυαδικό δέντρο, κοινώς γνωστή ως **Symbol Table**. Σκοπός αυτής της δομής δεδομένων είναι να εισάγει και να διατηρεί μεταβλητές, συναρτήσεις, strings και αριθμοί που συναντώνται κατά την ανάλυση. Έτσι όταν η γλώσσα συναντήσει έναν identifier, ελέγχει αν βρίσκεται ήδη στο Symbol Table. Αν όχι τον προσθέτει. Στον symbol table αποθηκεύονται πληροφορίες για κάθε στοιχείο. Για μεταβλητές αποθηκεύεται το όνομά τους, για συναρτήσεις το όνομα τους καθώς και ο αριθμός των παραμέτρων που δέχονται ενώ για strings και αριθμούς η τιμή και ο τύπος δεδομένων τους. Μία σημαντική διαδικασία που λαμβάνει χώρα κατά την ανάλυση είναι η δημιουργία ενδιάμεσου κώδικα ο οποίος χρησιμοποιείται στην εκτέλεση. Ο ενδιάμεσος κώδικας είναι μία σύντομη αναπαράσταση του κώδικα, όπου υπάρχουν επιπλέον πληροφορίες (π.χ. για το τί εντολή ακολουθεί, που πρέπει να μεταφερθεί η εκτέλεση λόγω διακλαδώσεων κλπ.).
- **Φάση Εκτέλεσης :** Εάν η φάση ανάλυσης ολοκληρωθεί χωρίς λάθη εκκινείται η φάση εκτέλεσης. Στην φάση αυτή γίνεται εκτέλεση του προγράμματος μέσω του ενδιάμεσου κώδικα. Στην φάση αυτή, η σημασία του Symbol Table είναι επίσης κρίσιμη. Ρόλος του είναι να κρατά την τρέχουσα τιμή της κάθε μεταβλητής του προγράμματος. Επιπλέον στην φάση αυτή δημιουργείται και χρησιμοποιείται μία στοίβα. Σκοπός της είναι να παρακολουθεί την εκτέλεση των συναρτήσεων και την σειρά κλήσεως αυτών. Κάθε συνάρτηση που δημιουργεί ο προγραμματιστής, έχει δικό της ενδιάμεσο κώδικα και symbol table. Έτσι η γλώσσα ξεκινά την εκτέλεση χρησιμοποιώντας τον ενδιάμεσο κώδικα του κυρίως προγράμματος. Καθώς γίνονται

κλήσεις σε συναρτήσεις, ο ενδιάμεσος κώδικας της κάθε συνάρτησης σπρώχνεται πάνω στην στοίβα. Έτσι ο δείκτης της στοίβας δείχνει πάντα στον ενδιάμεσο κώδικα της συνάρτησης που εκτελείται εκείνη την στιγμή. Η γλώσσα εκτελεί πάντα εντολές από τον ενδιάμεσο που βρίσκεται στην κορυφή της στοίβας. Όταν τελειώσει η εκτέλεση μιας συνάρτησης, ο ενδιάμεσος κώδικας αυτής αφαιρείται (pop) από της κορυφή της στοίβας, έτσι η εκτέλεση συνεχίζεται στην συνάρτηση που είχε κληθεί ακριβώς προηγουμένως.

4.3. Χαρακτηριστικά της γλώσσας

Η ScriptC υιοθετεί το συντακτικό της γλώσσας προγραμματισμού C με περιορισμούς που επιβάλλει η φύση της, καθιστώντας την ένα υποσύνολο της C με ορισμένες διαφοροποιήσεις. Η ScriptC δεν έχει δηλώσεις μεταβλητών και κατά συνέπεια αυστηρούς τύπους δεδομένων. Μία μεταβλητή αναγνωρίζεται κατ'ευθείαν την πρώτη στιγμή που χρησιμοποιείται. Οι τύποι δεδομένων που υποστηρίζονται είναι integer, double, string, connection (σε βάση δεδομένων) και recordset. Στις περιπτώσεις όπου χρειάζονται Boolean τιμές, μία μη μηδενική αριθμητική τιμή θεωρείται true ενώ μηδενική false. Η γλώσσα δεν υλοποιεί scope και επομένως μεταβλητές που αναφέρονται για πρώτη φορά μέσα σε ένα block συνεχίζουν να υφίστανται και πέρα από αυτό. Η γλώσσα αναλαμβάνει να κάνει αυτόματα το cast από τύπο τον ένα δεδομένων στον άλλο. Αξίζει να σημειωθεί ότι μια μεταβλητή μπορεί να αλλάξει τύπο δεδομένων οποτεδήποτε. Π.χ. αν αναθέσουμε σε μια μεταβλητή μια ακέραια τιμή, αργότερα μπορούμε να αναθέσουμε σε αυτή ένα string ή ένα double κλπ. Σημειώνεται ότι η γλώσσα ScriptC είναι case sensitive, δηλ. κάνει διαχώριση πεζών και κεφαλαίων. Παρατίθεται ένα παράδειγμα μιας σελίδας γραμμένης σε ScriptC :

```

<HTML>
  <HEAD>
    <TITLE>Simple Script</TITLE>
  </HEAD>
  <BODY>
    <%
i = 0; /* Εδώ ουσιαστικά δηλώνουμε και αρχικοποιούμε την μεταβλητή i */
    while (i<10)
    {
        /* Προσθέτουμε δυναμικά HTML κώδικα στο αρχείο */
        ResponseWrite("<input type=text value=" . i . ">");
        i++;
    }
    %>
  </BODY>
</HTML>

```

Το παραπάνω παράδειγμα προσθέτει - «γράφει» ανάμεσα από τα tags <body> και </body> δέκα φορές την συμβολοσειρά :

<input type=text value=x>

όπου x η εκάστοτε τιμή του i. Βλέπουμε πόσο εύκολο είναι να προσθέσει κανείς δυναμικά HTML μέσα στην σελίδα, καθώς και πως ο server side κώδικας, που κάνει parse και εκτελεί η γλώσσα, συνυπάρχει με το στατικό HTML της σελίδας. Το αποτέλεσμα που θα δει ένας πελάτης που θα αιτηθεί την σελίδα αυτή από τον Web Server προφανώς **δεν** θα περιέχει τον server side κώδικα, παρα μόνο το HTML και το **αποτέλεσμα** της εκτέλεσης του κώδικα (αν αυτό υφίσταται). Δηλαδή αν ο πελάτης κοιτάξει το HTML της σελίδας αυτής στον browser του θα δει :

```

<HTML>
  <HEAD>
    <TITLE>Simple Script</TITLE>
  </HEAD>
  <BODY>
    <input type=text value=0>
    <input type=text value=1>
    <input type=text value=2>
    <input type=text value=3>
    <input type=text value=4>
    <input type=text value=5>
    <input type=text value=6>
    <input type=text value=7>
    <input type=text value=8>
    <input type=text value=9>
  </BODY>
</HTML>

```

4.4. Τελεστές και προτεραιότητα

Η γλώσσα διαθέτει τους παρακάτω τελεστές:

- Αριθμητικοί
 - o + (μοναδιαίος και δυαδικός)
 - o - (μοναδιαίος και δυαδικός)
 - o *
 - o /
 - o %
 - o ++ (pre/postfix)
 - o -- (pre/postfix)
- Ανάθεσης
 - o =
 - o +=
 - o -=
 - o *=
 - o /=
 - o %=
 - o <<=
 - o >>=

- o `&=`
- o `^=`
- o `|=`
- o `.=`
- Bitwise
 - o `&`
 - o `|`
 - o `^`
 - o `~`
 - o `>>`
 - o `<<`
- Λογικοί
 - o `&&`
 - o `||`
 - o `!`
- Ισότητα
 - o `==`
 - o `!=`
- Συσχετιστικοί
 - o `>`
 - o `<`
 - o `>=`
 - o `<=`
- Διάφοροι
 - o `(`
 - o `)`
 - o `,`
 - o `.`

Ειδικά για τον τελεστή «.» (τελεία) αναφέρεται ότι χρησιμοποιείται για την συνένωση δύο string expressions. Σημαντικό είναι το γεγονός ότι βάσει των τελεστών γίνεται και η εκτέλεση της κάθε πρότασης.

Π.χ. η πρόταση : $x = 1 + "4"$;

Αναθέτει στην μεταβλητή x την τιμή 5. Αυτό γιατί γίνεται χρήση του τελεστή $+$ και γίνεται αποτίμηση της αριθμητικής τιμής του κάθε όρου της πρότασης. Στην περίπτωση : $x = 1 . 5$; η μεταβλητή x θα πάρει την τιμή **"15"** ως συμβολοσειρά και πάλι λόγω χρήσης του συγκεκριμένου τελεστή, ο οποίος μετατρέπει την τιμή του κάθε όρου σε συμβολοσειρά.

Ο παρακάτω πίνακας δείχνει την προτεραιότητα αποτίμησης των τελεστών. Τελεστές με την ίδια προτεραιότητα αποτιμούνται σύμφωνα με την φορά εκτέλεσης, όπως αυτοί βρίσκονται στην πρόταση.

Προτεραιότητα	Τελεστές	Φορά εκτέλεσης
1	()	Αριστερά προς δεξιά
2	(prefix) ++,--	Δεξιά προς αριστερά
3	(μοναδιαίοι) +, -, !, ~	Δεξιά προς αριστερά
4	*, /, %	Αριστερά προς δεξιά
5	+, -, .	Αριστερά προς δεξιά
6	<<, >>	Αριστερά προς δεξιά
7	<, <=, >, >=	Αριστερά προς δεξιά
8	=, !=	Αριστερά προς δεξιά
9	&, , ^	Αριστερά προς δεξιά
10	&&,	Αριστερά προς δεξιά
11	=, *=, /=, +=, -=, %=, &=, =, ^=, <<=, >>=, . =	Δεξιά προς αριστερά

4.5. Εντολές

Η γλώσσα υποστηρίζει τις παρακάτω εντολές:

- **if** (<expression>) <statement-list>
- **if** (<expression>) <statement-list> **else** <statement-list>
- **for** (<expression>;<expression>;<expression>) <statement-list>
- **while** (<expression>) <statement-list>
- **do** { <statement-list> } **while** (<expression>)
- **switch** (<expression>) <case-branch> : <statement-list> [**default** :] <statement-list>
- **return** [<expression>]
- **break**

4.6. Built-In Συναρτήσεις

Η γλώσσα διαθέτει τις εξής συναρτήσεις ενσωματωμένες :

- **BeginTrans** (connection)
 - ο Δέχεται σαν παράμετρο μία μεταβλητή στην οποία ήδη έχει ανατεθεί μία σύνδεση σε κάποια βάση δεδομένων, η οποία (σύνδεση) είναι ανοιχτή.
 - ο Ξεκινεί ένα transaction σε αυτή την σύνδεση.
 - ο Επιστρέφει void.
- **ClearErrors** ()
 - ο Δεν δέχεται παραμέτρους
 - ο Μηδενίζει τις μεταβλητές τις γλώσσας που χειρίζονται run time errors.
 - ο Επιστρέφει void
- **CloseConnection** (connection)
 - ο Δέχεται σαν παράμετρο μία μεταβλητή στην οποία ήδη έχει ανατεθεί μία σύνδεση σε κάποια βάση δεδομένων, η οποία (σύνδεση) είναι ανοιχτή.
 - ο Κλείνει την σύνδεση αυτή.
 - ο Επιστρέφει void.
- **ColumnCount** (recordset)
 - ο Δέχεται μία μεταβλητή στην οποία έχει ανατεθεί ένα recordset το οποίο ανακτήθηκε από μια ΒΔ μέσω μιας σύνδεσης.

- Επιστρέφει το πλήθος των στηλών του συγκεκριμένου recordset.
- Επιστρέφει int.
- **ColumnName** (recordset, int)
 - Η πρώτη παράμετρος είναι ένα recordset, ενώ η δεύτερη είναι ένας ακέραιος – index για την στήλη της οποίας το όνομα πρόκειται να επιστραφεί.
 - Επιστρέφει το όνομα της στήλης μέσα στο recordset η οποία βρίσκεται στην θέση που καθορίζει η δεύτερη παράμετρος.
 - Επιστρέφει string
- **CommitTrans** (connection)
 - Δέχεται σαν παράμετρο μία μεταβλητή στην οποία ήδη έχει ανατεθεί μία σύνδεση σε κάποια βάση δεδομένων, η οποία (σύνδεση) είναι ανοιχτή και βρίσκεται ήδη σε transaction.
 - Επικυρώνει το τρέχον transaction.
 - Επιστρέφει void.
- **EOF** (recordset)
 - Δέχεται μία μεταβλητή στην οποία έχει ανατεθεί ένα recordset το οποίο ανακτήθηκε από μια ΒΔ μέσω μιας σύνδεσης.
 - Εξετάζει αν το recordset έχει φτάσει στο τέλος των εγγραφών που περιέχει.
 - Επιστρέφει bool.
- **ErrorMessage** ()
 - Δεν δέχεται παραμέτρους.
 - Επιστρέφει το μήνυμα του τελευταίου run time λάθους που συνέβη.
 - Επιστρέφει string.
- **ExecNonQuery** (string, connection)
 - Η πρώτη παράμετρος είναι μια SQL εντολή, η οποία δεν επιστρέφει result set. Η δεύτερη παράμετρος είναι μια ανοιχτή σύνδεση σε ΒΔ.
 - Εκτελεί μια εντολή SQL που δεν επιστρέφει result set (INSERT, UPDATE, DELETE κλπ.) και επιστρέφει τον αριθμό των εγγραφών που επηρεάστηκαν από την εντολή.
 - Επιστρέφει int.

- **ExecRecordset** (string, connection)
 - Η πρώτη παράμετρος είναι μια SQL εντολή, η οποία επιστρέφει result set. Η δεύτερη παράμετρος είναι μια ανοιχτή σύνδεση σε ΒΔ.
 - Εκτελεί μια εντολή SQL που επιστρέφει result set (SELECT κλπ.) και επιστρέφει ένα recordset που περιέχει αυτό το result set.
 - Επιστρέφει recordset.
- **FetchRow** (recordset)
 - Δέχεται μία μεταβλητή στην οποία έχει ανατεθεί ένα recordset το οποίο ανακτήθηκε από μια ΒΔ μέσω μιας σύνδεσης.
 - Προχωρά το recordset κατά μία εγγραφή.
 - Επιστρέφει void.
- **GetRecordsetItem** (recordset, string)
 - Δέχεται μία μεταβλητή στην οποία έχει ανατεθεί ένα recordset το οποίο ανακτήθηκε από μια ΒΔ μέσω μιας σύνδεσης καθώς και ένα string που περιέχει το όνομα της στήλης της οποίας η τιμή πρόκειται να επιστραφεί.
 - Επιστρέφει την τιμή της στήλης που ορίζει η δεύτερη παράμετρος, στην συγκεκριμένη εγγραφή όπου βρίσκεται ο κέρσορας του recordset εκείνη την στιγμή.
 - Επιστρέφει object.
- **HasErrors** ()
 - Δεν δέχεται παραμέτρους
 - Επιστρέφει λογική τιμή ανάλογα με το αν έχουν συμβεί λάθη κατά την διάρκεια εκτέλεσης του προγράμματος.
 - Επιστρέφει bool.
- **Length** (object)
 - Παίρνει σαν παράμετρο είτε ένα recordset είτε ένα string.
 - Εάν η παράμετρος είναι recordset, επιστρέφει το πλήθος των εγγραφών που περιέχει το recordset. Εάν η παράμετρος είναι string επιστρέφει το μήκος του string.
 - Επιστρέφει int.

- **OpenConnection** (string)
 - Δέχεται παράμετρο μία μεταβλητή τύπου string που περιέχει ένα connection string με πληροφορίες για σύνδεση σε μία ΒΔ.
 - Ανοίγει μία σύνδεση με μία ΒΔ.
 - Επιστρέφει connection.
- **QueryString** ([string])
 - Δέχεται προαιρετικά ένα string με το όνομα το πεδίου από το Query String που πρόκειται να ανακτηθεί.
 - Εάν υπάρχει παράμετρος, επιστρέφεται η τιμή του πεδίου που υποδεικνύει αυτή, αλλιώς επιστρέφεται ολόκληρο το Query String.
 - Επιστρέφει string.
- **RequestForm** ([string])
 - Δέχεται προαιρετικά ένα string με το όνομα το πεδίου από την μέθοδο POST, που έχει προηγηθεί, που πρόκειται να ανακτηθεί.
 - Εάν υπάρχει παράμετρος, επιστρέφεται η τιμή του πεδίου που υποδεικνύει αυτή, αλλιώς επιστρέφεται ολόκληρο το POST.
 - Επιστρέφει string.
- **ResponseRedirect** (string)
 - Δέχεται παράμετρο ένα string που περιέχει ένα URL.
 - Ανακατευθύνει τον πελάτη στο URL που καθορίζει η παράμετρος.
 - Επιστρέφει void.
- **ResponseWrite** (string)
 - Δέχεται παράμετρο ένα string.
 - Προσθέτει στην έξοδο του προγράμματος την παράμετρο.
 - Επιστρέφει void.
- **RollbackTrans** (connection)
 - Δέχεται σαν παράμετρο μία μεταβλητή στην οποία ήδη έχει ανατεθεί μία σύνδεση σε κάποια βάση δεδομένων, η οποία (σύνδεση) είναι ανοιχτή και βρίσκεται σε transaction.
 - Αναιρεί το transaction.
 - Επιστρέφει void.

- **SuppressErrors** (bool)
 - ο Δέχεται μία λογική τιμή σαν παράμετρο (0 – 1).
 - ο Εάν η παράμετρος είναι 0, η γλώσσα σταματά την διαδικασία εκτέλεσης όταν συμβεί κάποια εξαίρεση (run time error). Εάν είναι μή μηδενική, τότε η γλώσσα δεν σταματά την εκτέλεση όταν συμβαίνουν εξαιρέσεις.

4.7. Συναρτήσεις

Η γλώσσα υποστηρίζει συναρτήσεις που καθορίζονται από τον προγραμματιστή. Μία συνάρτηση αναγνωρίζεται είτε αφού η διαδικασία της ανάλυσης (parsing) συναντήσει την επικεφαλίδα της συνάρτησης ή την υλοποίηση αυτής. Το συντακτικό για τις συναρτήσεις καθορίζεται ως εξής :

Επικεφαλίδα συνάρτησης :

function function_name (<parameter-list>)

Υλοποίηση συνάρτησης :

function function_name (<parameter-list>) {<statement-list>}

Παράδειγμα

```
<%
  /*Header*/
  function samplefunction(a,b);

  /* Implementation*/
  function samplefunction(a,b)
  {
    return a + b;
  }
%>
```

Η επικεφαλίδα μιας συνάρτησης δεν είναι υποχρεωτικό να υπάρχει. Αντίθετα η υλοποίηση είναι υποχρεωτική. Μια συνάρτηση μπορεί να επιστρέφει μία τιμή μέσω της εντολής **return** το οποίο ωστόσο δεν είναι υποχρεωτικό.

4.8. Χειρισμός εξαιρέσεων κατά την διάρκεια εκτέλεσης

Ο χειρισμός λαθών κατά την διάρκεια εκτέλεσης γίνεται μέσω της ρουτίνας SuppressErrors. Εάν δεν γίνει χρήση αυτής, η εκτέλεση διακόπτεται και αποστέλλεται επεξηγηματική σελίδα στον πελάτη με πληροφορίες για την εξαίρεση

που συνέβη. Αν σε κάποιο σημείο ο προγραμματιστής θέλει να χειριστεί ο ίδιος τα λάθη που συμβαίνουν, πρέπει να χρησιμοποιήσει την SuppressErrors περνώντας της μη μηδενική(true) τιμή. Από το σημείο που θα γίνει αυτό, μέχρι να ξαναγίνει χρήση της συνάρτησης με μηδενική τιμή ή μέχρι το τέλος του προγράμματος, η εκτέλεση συνεχίζεται κανονικά παρόλα τα λάθη που συμβαίνουν. Μέσω της συνάρτησης HasErrors, ο προγραμματιστής μπορεί να γνωρίζει αν συνέβη κάποιο λάθος, ενώ μέσω της συνάρτησης ErrorMessage μπορεί να ανακτήσει το μήνυμα του τελευταίου λάθους.

4.9. Συνδεσιμότητα με βάσεις δεδομένων

Μέσω της ScriptC δίνεται η δυνατότητα σύνδεσης σε μία βάση δεδομένων. Η δημιουργία μιας σύνδεσης γίνεται με κλήση της συνάρτησης OpenConnection. Στην συνάρτηση αυτή πρέπει να δοθεί ένα Connection String όπως αυτό καθορίζεται από τα **ActiveX Data Objects (ADO)**, καθώς γίνεται χρήση αυτών εσωτερικά από την γλώσσα. Επιπλέον υπάρχουν συναρτήσεις για εκτέλεση εντολών SQL και χειρισμό transactions όπως αναφέρεται πιο πάνω.

5. ΒΙΒΛΙΟΓΡΑΦΙΑ

- The Art of Computer Programming - Donald E. Knuth
- Writing Compilers and Interpreters – Ronald Mak
- Microsoft Developer Network (MSDN)
- The Common Gateway Interface - <http://hoohoo.ncsa.uiuc.edu/cgi/overview.html>
- World Wide Web Consortium - <http://www.w3.org/>