

**Τμήμα
Μηχανικών
Πληροφορικής τ.ε.**

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

Πτυχιακή Εργασία

Σχεδιασμός και κατασκευή μετεωρολογικού σταθμού με αισθητήρες, συλλογή δεδομένων και απομακρυσμένη διαχείρισή του.

(Θεωρία και πράξη)

Κατσώνης Αντώνιος

A.M.: 1420

Επιβλέπων καθηγητής: Χριστοδούλου Σωτήριος

Αντίρριο 21/04/2016

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο 21/04/2016

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1.

2.

3.

Περίληψη

Στα πλαίσια της παρούσας πτυχιακής εργασίας αναπτύχθηκε μια εφαρμογή που αφορά την μελέτη και κατασκευή ενός πρωτότυπου συστήματος με αισθητήρες για την περιοδική αποστολή δεδομένων όπως εικόνες, θερμοκρασία, υγρασία και άλλες μετρήσεις από το σημείο που το τοποθετούμε προς κάποιον κεντρικό server. Το σύστημα αυτό είναι ανεξάρτητο από το δίκτυο της Δ.Ε.Η. και από τον πάροχο ευρυζωνικών υπηρεσιών όπως το ADSL Internet του δικτύου σταθερής τηλεφωνίας. Επίσης ο τοπικός έλεγχος και η συντήρηση γίνονται σπάνια καθώς δημιουργήθηκαν σενάρια για δυνατότητα απομακρυσμένου ελέγχου.

Αυτό το ολοκληρωμένο σύστημα αποτελείται από ένα πολυεστερικό κουτί με προστασία IP66, φωτοβολταϊκό panel, μπαταρία, τροφοδοτικό, camera, compact weather station και από έναν Linux server ο οποίος είναι συνδεδεμένος με ένα 2G/3G modem. Η εφαρμογή ολοκληρώνεται με την αποστολή των δεδομένων σε κάποιον server ώστε να απεικονίζονται σε HTML χρησιμοποιώντας γλώσσες προγραμματισμού όπως JavaScript, PHP για να τα βλέπει ο διαχειριστής ή οι επισκέπτες.

Μαθήματα που διδάχθηκα στη σχολή και με βοήθησαν στη συγκεκριμένη υλοποίηση:

Πρωτεύοντα:

Προγραμματισμός Διαδικτύου

Ηλεκτρονική

Τηλεπικοινωνιακά Συστήματα

Λειτουργικά συστήματα

Δευτερεύοντα:

Δίκτυα

Δίκτυα αισθητήρων

Δίκτυα κινητών επικοινωνιών

Abstract

This thesis is based on an application which was developed and concerns the study and construction of a prototype system with sensors for the periodical data transfer such as images, temperature, humidity and other measurements, from a fixed point to a main server. This system is independent from electricity and ADSL Internet suppliers. Also on the spot managing and maintenance is seldom, as there are scripts for controlling remotely.

This complete system consists of a polyester box with IP66 protection, solar panel, battery, dc supplier, camera, compact weather station and a Linux server which is connected to a 2G/3G modem.

The application is completed with the data transferred to a server, so it is displayed on HTML, by using programming languages such as JavaScript, PHP so that it can be viewed by the administrator or visitors.

Subjects I was taught at the college have help me in this implementation:

Primary:

Web programming

Electronics

Telecommunication systems

Operating systems

Secondary:

Networks

Sensor networks

Mobile communication networks

Ευχαριστίες

Θέλω να ευχαριστήσω τον καθηγητή μου κ. Σωτήρη Χριστοδούλου για την επίβλεψη και καθοδήγηση της παρούσας πτυχιακής εργασίας.

Επίσης να ευχαριστήσω και τους καθηγητές του ΤΕΙ που κατά την διάρκεια της φοίτησής μου, μου έδωσαν τις απαραίτητες γνώσεις για να μπορέσω ανταποκριθώ στη συγκεκριμένη υλοποίηση.

Λέξεις κλειδιά

Weather station, Raspberry Pi, Camera, microSD, Raspbian, Linux, GPIO, 3G modem, Solar panel, Battery, Relay, Fan, Server, Cron, Shell script, udev rules, Weewx, Gedit, HTML, XML, JavaScript, PHP.

Πίνακας περιεχομένων

Περίληψη	3
Abstract.....	4
Ευχαριστίες.....	5
Λέξεις κλειδιά.....	6
Πίνακας περιεχομένων.....	7
Εισαγωγή	11
1. Σύλλογή και περιγραφή εξοπλισμού	11
1.1 Μετεωρολογικός σταθμός	12
1.2 Πολυεστερικό κουτί με προστασία IP66.....	13
1.3 Πολυεστερικό κουτί με διάφανη πόρτα και προστασία IP66.....	14
1.4 Φωτοβολταϊκό στοιχείο.....	15
1.5 Ρυθμιστής φόρτισης.....	16
1.6 Μπαταρία 12V.....	17
1.7 Ασφαλειοδιακόπτης.....	18
1.8 Διακόπτης επιλογής 3 θέσεων με κλειδί.....	19
1.9 Διακόπτης επιλογής 2 θέσεων	20
1.10 Μαγνητικός διακόπτης 2 επαφών.....	21
1.11 Ανεμιστήρας	22
1.12 Τροφοδοτικό μετατροπής συνεχόμενης τάσης	23
1.13 GSM Alarm Dialer.....	24
1.14 Raspberry Pi 2.....	25
1.15 Κάρτα μνήμης για το Raspberry	27
1.16 Κάμερα.....	28
1.17 Mobile Broadband Modem	29
1.18 Ρελέ ανεμιστήρα.....	30
1.19 Magnetic Led Strip	31
2. Τεχνολογίες ιστού.....	32
2.1 HTML	32
2.1.1 Προέλευση	33
2.1.2 Πρώτες προδιαγραφές	33
2.1.3 Σήμανση	35
2.1.4 Στοιχεία.....	36

2.1.5 Παραδείγματα στοιχείων.....	37
2.1.6 Ιδιότητες	40
2.1.7 Αναφορές οντοτήτων χαρακτήρων	42
2.1.8 Τύποι δεδομένων	43
2.1.9 Δήλωση τύπου εγγράφου	43
2.1.10 Σημασιολογική	44
2.1.11 Μεταφορά της HTML	45
2.1.12 Δημιουργία με προγράμματα WYSIWYG.....	47
2.2.1 Η Βασική Σύνταξη των CSS	49
2.2.2 Εφαρμογή των CSS σε ένα HTML έγγραφο.....	50
2.3 XML	53
2.3.1 Χαρακτήρας Unicode.....	54
2.3.2 Επεξεργαστής και Εφαρμογή.....	54
2.3.3 Σήμανση και Περιεχόμενο	54
2.3.4 Ετικέτα	54
2.3.5 Στοιχείο	55
2.3.6 Χαρακτηριστικό	55
2.3.7 Δήλωση XML	55
2.3.8 Παράδειγμα	55
2.3.9 Χαρακτήρες και διαφυγή	56
2.4 JavaScript	56
2.4.1 Ιστορία	57
2.4.2 Μοντέλο εκτέλεσης	58
2.4.3 Javascript και Java	59
2.4.4 Δείγμα κώδικα Javascript	59
2.5 PHP	60
2.5.1 Επεκτάσεις αρχείων και διακομιστές.....	61
2.5.2 Ιστορία της PHP.....	61
3. Υλοποίηση συστήματος.....	62
3.1 Προγραμματισμός GSM Alarm Dialer και SIM Card	65
3.2 Προετοιμασία και προγραμματισμός Raspberry	65
3.2.1 Εγκατάσταση Raspbian.....	66
3.2.2 Εγκατάσταση πακέτων για Raspbian	66
3.2.2.1 Gedit	67

3.2.2.2 PPP	67
3.2.2.3 Weewx	67
3.2.3 Δημιουργία αρχείου shell script	70
3.2.4 Εκτέλεση αρχείων κατά την εκκίνηση	71
3.2.5 Περιοδική εκτέλεση αρχείων με τον χρονοπρογραμματιστή Cron	71
3.2.6 Raspberry Time Keeping Failure	72
3.2.7 Udev Rules	74
3.2.8 Ακροδέκτης GPIO	76
3.2.9 Σύνδεση ρελέ με τον ακροδέκτη του Raspberry	77
3.3 Προγραμματισμός Web Server	78
3.3.1 Client Side	78
3.3.1.2 Parsing the XML file	78
3.3.2 Server Side	79
3.3.2.1 Αρχείο PHP	79
3.3.2.1 Δημιουργία Under Construction ιστοσελίδας	79
Προβλήματα που αντιμετωπίστηκαν	80
Αναφορές / Links	82
AT Commands	83
Linux commands help	84
chat	84
cp	84
cut	87
date	89
expr	93
grep	95
kill	98
ls	98
mv	103
netstat	104
pppd	106
ps	107
reboot	107
rm	108
sed	109

shutdown	111
stat	112
tr	115
wc	117

Πίνακας εικόνων

Εικόνα 1 Weather Station.....	12
Εικόνα 2 Πολυεστερικό κουτί.....	13
Εικόνα 3 Πολυεστερικό κουτί με διάφανη πόρτα.....	14
Εικόνα 4 Solar Panel.....	15
Εικόνα 5 Solar Charge Controller.....	16
Εικόνα 6 Battery.....	17
Εικόνα 7 Fuse Disconnecter.....	18
Εικόνα 8 Selector Switch 3 positions.....	19
Εικόνα 9 Selector Switch 2 positions.....	20
Εικόνα 10 Safety Magnetic Switch.....	21
Εικόνα 11 Forced Ventilation.....	22
Εικόνα 12 DC to DC Converter.....	23
Εικόνα 13 GSM Alarm Dialer.....	24
Εικόνα 14 Raspberry Pi.....	25
Εικόνα 15 Raspberry Logo.....	26
Εικόνα 16 MicroSD Memory.....	27
Εικόνα 17 Camera Module.....	28
Εικόνα 18 GSM/WCDMA Modem.....	29
Εικόνα 19 Relay Module.....	30
Εικόνα 20 Led.....	31
Εικόνα 21 Raspbian Logo.....	66
Εικόνα 22 Isusb.....	75
Εικόνα 23 General purpose input/output (GPIO).....	76
Εικόνα 24 Transistor.....	77

Εισαγωγή

Τα περισσότερα συστήματα που υπάρχουν αυτή τη στιγμή στην αγορά λειτουργούν ως εξής:

Ένα DataLogger συλλέγει δεδομένα και τα στέλνει με δυο τρόπους:

Με sms προς κινητό τηλέφωνο κάθε μια ώρα χρησιμοποιώντας το δίκτυο κινητής τηλεφωνίας. (Μέγιστο επιτρεπόμενο όριο 160 χαρακτήρες / sms)

Με e-mail προς τον χρήστη κάθε μισή ώρα χρησιμοποιώντας το ADSL Internet.

Μειονέκτημα είναι ότι τα DataLoggers δεν υποστηρίζουν camera για τη λήψη εικόνων στη τοποθεσία που τα κάνουμε εγκατάσταση.

Ο βασικός σκοπός της πτυχιακής εργασίας είναι να λύσει αυτό το πρόβλημα και περιγράψει με ποιον τρόπο κατασκευάστηκε ένα αυτοματοποιημένο σύστημα που περιλαμβάνει αισθητήρες και camera και στέλνει τα δεδομένα (μετεωρολογικά ή εικόνες) σε πραγματικό χρόνο από οποιοδήποτε απομακρυσμένο σημείο. Στη προκειμένη περίπτωση το σύστημα είναι ανεξάρτητο. Αυτό σημαίνει:

Δεν είναι απαραίτητο να υπάρχει δίκτυο χαμηλής τάσης της Δ.Ε.Η.!

Δεν είναι απαραίτητο να υπάρχει δίκτυο σταθερής τηλεφωνίας για ADSL Internet!

Δεν απαιτούνται συχνές επισκέψεις!

Ο έλεγχος και η συντήρηση γίνονται σπάνια!

Τα δεδομένα στέλνονται σε διάστημα Update Interval: 5 λεπτά το ελάχιστο χρησιμοποιώντας το δίκτυο 2G/3G κινητής τηλεφωνίας!

Υπάρχει δυνατότητα απομακρυσμένης διαχείρισης όπως επανεκκίνηση, τερματισμός, από 2G δίκτυο σε 3G, άνοιγμα/κλείσιμο κάμερας και πολλά άλλα που θα δούμε παρακάτω.

1. Συλλογή και περιγραφή εξοπλισμού

Τα εξαρτήματα που χρησιμοποιήθηκαν για την συγκεκριμένη εφαρμογή έχουν υψηλό κόστος και ποιότητα κατασκευής λόγω ότι θα εγκατασταθεί σε δύσκολες συνθήκες. Υπάρχουν και

εναλλακτικές λύσεις για όλα τα εξαρτήματα με χαμηλότερο κόστος και ποιότητα κατασκευής. Αυτό το επιλέγει ο κάθε χρήστης ανάλογα με τις απαιτήσεις που θα έχει.

1.1 Μετεωρολογικός σταθμός



Εικόνα 1 Weather Station

Χαρακτηριστικά

Μοντέλο: DAVIS Vantage Vue

Χώρα προέλευσης: Αμερική

Μετεωρολογικός σταθμός με αισθητήρες μέτρησης θερμοκρασίας, υγρασίας, βαρομετρικής πίεσης, ταχύτητας - κατεύθυνσης ανέμου, χιλιοστά βροχής και ώρα ανατολής-δύσης του ηλίου. Συνδέεται με USB στο Raspberry.

1.2 Πολυεστερικό κουτί με προστασία IP66



Εικόνα 2 Πολυεστερικό κουτί

Χαρακτηριστικά

Μοντέλο: Schneider Electric NSYPLM43

Χώρα προέλευσης: Γαλλία

Πολυεστερικό κουτί για την τοποθέτηση των εξαρτημάτων με δείκτη στεγανότητας IP66. Κατάλληλο για δύσκολες συνθήκες, επίσης λόγω υλικού κατασκευής, αντέχει και στην υψηλή υπεριώδη ακτινοβολία του ηλίου. Μέσα σε αυτό τοποθετήθηκαν και συνδέθηκαν όλα τα εξαρτήματα.

1.3 Πολυεστερικό κουτί με διάφανη πόρτα και προστασία IP66



Εικόνα 3 Πολυεστερικό κουτί με διάφανη πόρτα

Χαρακτηριστικά

Μοντέλο: Schneider Electric NSYPLM3025T

Χώρα προέλευσης: Γαλλία

Κουτί μικρότερο σε διαστάσεις, επίσης με δείκτη στεγανότητας IP66 για την τοποθέτηση της μπαταρίας. Διάφανη πόρτα για την ομαλή λειτουργία της μπαταρίας τον χειμώνα σε πολύ χαμηλές θερμοκρασίες.

1.4 Φωτοβολταϊκό στοιχείο



Εικόνα 4 Solar Panel

Χαρακτηριστικά

Μοντέλο: Offgridtec 20W Mono Solarpanel 12V

Διαστάσεις: 430x345x250 mm

Χώρα προέλευσης: Γερμανία

Φωτοβολταϊκό στοιχείο για την λειτουργία του συστήματος και την φόρτιση της μπαταρίας την ημέρα. Η απόδοση εξαρτάται από την ηλιοφάνεια.

1.5 Ρυθμιστής φόρτισης



Εικόνα 5 Solar Charge Controller

Χαρακτηριστικά

Μοντέλο: Steca Solsum 6.6F

Χώρα προέλευσης: Γερμανία

Ο ρυθμιστής φόρτισης, που επίσης λέγεται αυτόματος-κόφτης-σταθεροποιητής, είναι απαραίτητο όργανο για να προστατεύει τις μπαταρίες από υπερβολική φόρτιση και πολλές φορές υπερβολική εκφόρτιση. Παρεμβάλλεται μεταξύ του φωτοβολταϊκού συλλέκτη και της μπαταρίας.

Όταν η μπαταρία δεν μπορεί και δεν πρέπει να δεχθεί περισσότερη ηλεκτρική φόρτιση, ο ρυθμιστής φόρτισης διακόπτει τη ροή του ρεύματος προς την μπαταρία και έτσι δεν υπάρχει κίνδυνος έκρηξης της μπαταρίας.

Επίσης, αν οι ηλεκτρικές συσκευές είναι συνδεδεμένες με το ρεύμα της μπαταρίας και εκείνη δεν πρέπει να φτάσει χαμηλά επίπεδα τάσης καθώς τα στοιχεία του θα υποστούν ζημιά, ο ρυθμιστής φόρτισης διακόπτει το ρεύμα προς τις συσκευές και προστατεύει την μπαταρία από αλλοίωση.

1.6 Μπαταρία 12V



Εικόνα 6 Battery

Χαρακτηριστικά

Μοντέλο: Winner Icarus 12V 21Ah

Χώρα προέλευσης: Γερμανία

Επαναφορτιζόμενη μπαταρία κλειστού τύπου για την λειτουργία του συστήματος τη νύχτα ή τις μέρες που έχει πολύ χαμηλή ηλιοφάνεια. Είναι High Rate Discharge και υψηλό Cycle Use.

1.7 Ασφαλειοδιακόπτης



Εικόνα 7 Fuse Disconnecter

Χαρακτηριστικά

Μοντέλο: Schneider Electric DF81

Χώρα προέλευσης: Γαλλία

Ασφαλειοδιακόπτης με τηκόμενη ασφάλεια έντασης 6Α. Χρησιμοποιούνται 2 τεμάχια το 1 προστατεύει το κύκλωμα του φωτοβολταϊκού και το άλλο το κύκλωμα της μπαταρίας.

1.8 Διακόπτης επιλογής 3 θέσεων με κλειδί



Εικόνα 8 Selector Switch 3 positions

Χαρακτηριστικά

Μοντέλο: Schneider Electric XB5AG03

Χώρα προέλευσης: Γαλλία

Διακόπτης επιλογής 3 θέσεων με κλειδί χωρίς επαναφορά. Το κλειδί βγαίνει σε όλες τις θέσεις. Ο διακόπτης αυτός στη πρώτη θέση το σύστημα είναι κλειστό, στη δεύτερη θέση φορτίζει η μπαταρία και στη τρίτη θέση φορτίζει η μπαταρία και το σύστημα είναι σε λειτουργία.

1.9 Διακόπτης επιλογής 2 θέσεων



Εικόνα 9 Selector Switch 2 positions

Χαρακτηριστικά

Μοντέλο: Schneider Electric XB5AJ21

Χώρα προέλευσης: Γαλλία

Διακόπτης επιλογής 2 θέσεων χωρίς επαναφορά. Ο διακόπτης αυτός στη πρώτη θέση κλείνει τον φωτισμό στο κουτί και στη δεύτερη θέση ανοίγει τον φωτισμός στο κουτί.

1.10 Μαγνητικός διακόπτης 2 επαφών



Εικόνα 10 Safety Magnetic Switch

Χαρακτηριστικά

Μοντέλο: Schneider Electric XCSDMC5902

Χώρα προέλευσης: Γαλλία

Μαγνητικός διακόπτης 2 επαφών, 4 καλωδίων που τοποθετήθηκε στη πόρτα του κουτιού και έχει 2 λειτουργίες. Η πρώτη λειτουργία όταν η πόρτα είναι κλειστή επιτρέπει την διέλευση του ηλεκτρικού ρεύματος η επαφή NO (Normally Open) στο καλώδιο που είναι συνδεδεμένο στο Input του ρελέ. Η δεύτερη λειτουργία όταν η πόρτα είναι ανοιχτή επιτρέπει την διέλευση του ηλεκτρικού ρεύματος η επαφή NC (Normally Closed) στο καλώδιο που είναι συνδεδεμένο στο Input του GSM Alarm Dialer.

Αποτέλεσμα. Όταν η πόρτα είναι κλειστή ο ανεμιστήρας λειτουργεί και το GSM Alarm Dialer βρίσκεται σε αναμονή. Όταν η πόρτα είναι ανοιχτή ο ανεμιστήρας δεν λειτουργεί και το GSM Alarm Dialer ειδοποιεί με τηλεφωνική κλήση για παραβίαση του συστήματος.

1.11 Ανεμιστήρας



Εικόνα 11 Forced Ventilation

Χαρακτηριστικά

Μοντέλο: Schneider Electric NSYCVF38M24DPF

Χώρα προέλευσης: Γαλλία

Ανεμιστήρας για την ψύξη του κουτιού και των εξαρτημάτων. Τοποθετήθηκε στο εμπρός μέρος της πόρτας και κάνει εξαγωγή αέρα. Περιέχει περσίδα και φίλτρο για να προστατεύει το κουτί εσωτερικά από νερό και σκόνη και έχει δείκτη στεγανότητας IP66. Λειτουργεί με ονομαστική τάση από 10 έως 27.6V DC. Η λειτουργία του εξαρτάται από την θερμοκρασία και την υγρασία του κουτιού, από την εξωτερική θερμοκρασία και υγρασία και από την θερμοκρασία του επεξεργαστή του Raspberry.

1.12 Τροφοδοτικό μετατροπής συνεχόμενης τάσης



Εικόνα 12 DC to DC Converter

Χαρακτηριστικά

Μοντέλο: MeanWell SD-15A-5

Χώρα προέλευσης: Κίνα

Η πλακέτα Raspberry Pi λειτουργεί με τάση +5V DC. Η τάση της μπαταρίας είναι +12V, αδύνατο να το συνδέσουμε απευθείας. Χρησιμοποιήθηκε τροφοδοτικό DC to DC Converter που δέχεται στην είσοδο τάση +12V DC και την μετατρέπει στην έξοδο σε +5V DC. Μέγιστη επιτρεπόμενη ισχύς 15W.

Χρησιμοποιούμε το πολύμετρο για την μέτρηση.

1.13 GSM Alarm Dialer



Εικόνα 13 GSM Alarm Dialer

Χαρακτηριστικά

Μοντέλο: Advent Controls A3

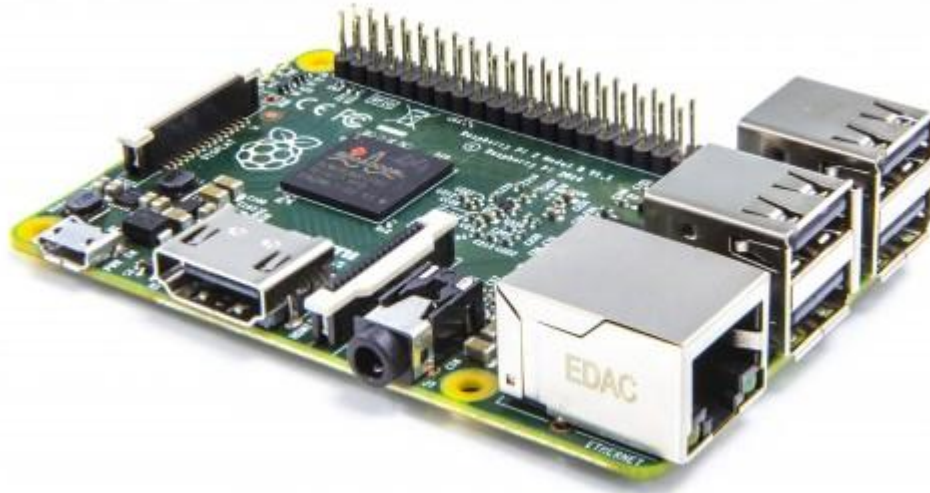
Χώρα προέλευσης: Αγγλία

Το GSM Alarm Dialer είναι μια συσκευή η οποία δέχεται κάρτα SIM και συνδέεται στο δίκτυο κινητής τηλεφωνίας. Ο Master User στέλνοντας SMS προς αυτή τη κάρτα προγραμματίζει τη συσκευή πως θα συμπεριφέρεται όταν δέχεται τηλεφωνική κλήση ή sms.

Το ρελέ που έχει ενσωματωμένο ελέγχεται με κλήση ή sms. Επίσης αν τα Inputs A και B γίνουν από LOW σε HIGH μας ειδοποιήσει με κλήση ή sms. Η διάρκεια είναι όσο επιθυμεί ο διαχειριστής.

Λειτουργεί με τάση 12V και συνδέθηκε πριν το τροφοδοτικό μετατροπής συνεχόμενης τάσης από 12V σε 5V.

1.14 Raspberry Pi 2



Εικόνα 14 Raspberry Pi

Χαρακτηριστικά

Μοντέλο: Raspberry Pi 2 Model B

Χώρα προέλευσης: Αγγλία

Το Raspberry Pi είναι ένας υπολογιστής με μέγεθος όσο μια πιστωτική κάρτα που συνδέεται στην οθόνη. Το λειτουργικό του σύστημα βασίζεται στα Linux. Είναι χρήσιμο για εργασίες οι οποίες χρειάζονται γραφικό περιβάλλον ή σύνδεση με το διαδίκτυο.

Η κατανάλωσή του αγγίζει τα 3W!

Τα χαρακτηριστικά του είναι τα εξής:

900MHz quad-core ARM Cortex-A7 CPU

1GB RAM

4 USB ports

40 GPIO pins

Full HDMI port

Ethernet port

Combined 3.5mm audio jack and composite video

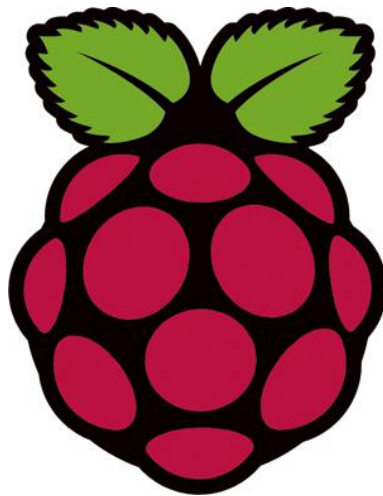
Camera interface (CSI)

Display interface (DSI)

Micro SD card slot

VideoCore IV 3D graphics core

Λειτουργικό σύστημα: Raspbian is the Foundation's official supported Operating System



Εικόνα 15 Raspberry Logo

1.15 Κάρτα μνήμης για το Raspberry



Εικόνα 16 MicroSD Memory

Χαρακτηριστικά

Μοντέλο: Kingston microSDHC 16GB UHS Speed

Χώρα προέλευσης: Ταϊβάν

Κάρτα μνήμης τύπου microSD χωρητικότητας 16GB Class 10 για το Raspberry. Στη κάρτα αυτή εγκαταστάθηκε το λειτουργικό σύστημα Raspbian και λειτουργεί ως σκληρός δίσκος. Επιλέγουμε αυτή τη χωρητικότητα για να υπάρχει αρκετός ελεύθερος χώρος στο σύστημα επειδή δημιουργούνται αρχεία και γραφήματα τα οποία κάποια θα στέλνονται περιοδικά σε κάποιον κεντρικό server και κάποια θα στέλνονται όταν τα αναζητήσει ο διαχειριστής μέσω της απομακρυσμένης διαχείρισης.

1.16 Κάμερα



Εικόνα 17 Camera Module

Χαρακτηριστικά

Μοντέλο: Raspberry Pi Camera

Χώρα προέλευσης: Αγγλία

Sensor: OmniVision OV5647

Still resolution: 5 Megapixels

Pixel size: 1.4 μm x 1.4 μm

Sensitivity: 680 mV/lux-sec

Κάμερα η οποία είναι επίσημη έκδοση του Raspberry για την λήψη εικόνων ή βίντεο.

1.17 Mobile Broadband Modem



Εικόνα 18 GSM/WCDMA Modem

Χαρακτηριστικά

Μοντέλο: HUAWEI E1750

Χώρα προέλευσης: Κίνα

2G/3G modem για σύνδεση στο Internet από το δίκτυο κινητής τηλεφωνίας. Επίσης λαμβάνει sms τα οποία χρησιμοποιούνται για απομακρυσμένη διαχείριση. Η επικοινωνία με το modem γίνεται με AT commands.

Modem Led Status:

Flashing Quickly > No Signal

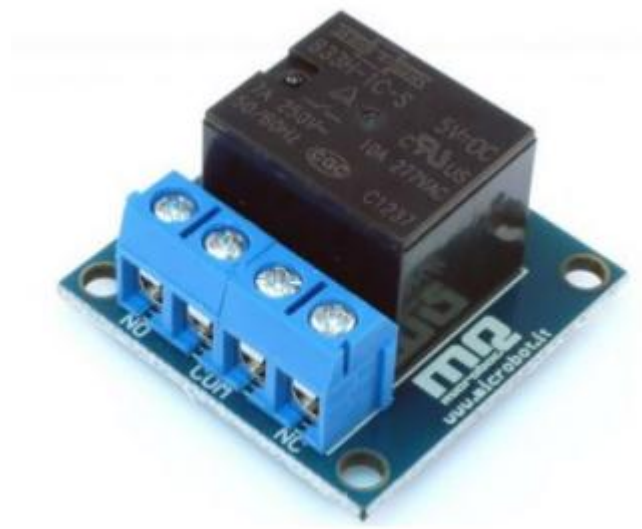
Green > GSM

Blue > WCDMA

Cyan > HSPA

Για παρακολούθηση των διαθέσιμων MB ή ανανέωσης πακέτων που προσφέρει η κάθε εταιρία φτιάχνουμε Account στο επίσημο site της εταιρίας που θα επιλέξουμε. Στην συγκεκριμένη υλοποίηση χρησιμοποιείται το δίκτυο κινητής τηλεφωνίας Vodafone.

1.18 Ρελέ ανεμιστήρα



Εικόνα 19 Relay Module

Χαρακτηριστικά

Μοντέλο: Microbot MR009-001.1

Χώρα προέλευσης: Ιταλία

Ρελέ το οποίο οπλίζει για να ανοίγει ο ανεμιστήρας. Το ρελέ αυτό συνδέθηκε επάνω στην πλακέτα του Raspberry και συγκεκριμένα σε κάποια GPIO. Λειτουργεί με τάση 5V και η τροφοδοσία ρεύματος για την λειτουργία του την δίνει επίσης το Raspberry. Το Raspberry παίρνει όλες τις απαραίτητες μετρήσεις και αν υπάρχει ανάγκη για εξαερισμό του συστήματος κάνει export κάποια από τις GPIO που ορίστηκε και οπλίζει το ρελέ.

1.19 Magnetic Led Strip



Εικόνα 20 Led

Χαρακτηριστικά

Μοντέλο: Bitfenix Alchemy 2.0

Χώρα προέλευσης: Κίνα

Led ταινία 12cm για το φωτισμό στο εσωτερικό του κουτιού. Έχει μαγνήτη στη πίσω πλευρά και στηρίζεται σε οποιαδήποτε μεταλλική επιφάνεια. Η κατανάλωση ρεύματος είναι χαμηλή και το χρώμα φωτισμού είναι ψυχρό λευκό. Χρησιμοποιήθηκαν 2 τεμάχια.

2. Τεχνολογίες ιστού

Για την ιστοσελίδα διαχείρισης και προβολής αποτελεσμάτων χρησιμοποιούνται τα εξής:

- HTML
- CSS
- XML
- JavaScript
- PHP

2.1 HTML

Η HTML (αρχικοποίηση του αγγλικού HyperText Markup Language, ελλ. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

2.1.1 Προέλευση

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990. Τον ίδιο χρόνο, ο Μπέρνερς Λι και ο μηχανικός συστημάτων πληροφορικής του CERN Robert Cailliau συνεργάστηκαν σε μια κοινή προσπάθεια εύρεσης χρηματοδότησης, αλλά το έργο δεν υιοθετήθηκε ποτέ επίσημα από το CERN. Στις προσωπικές του σημειώσεις από το 1990, ο Μπέρνερς Λι αριθμεί «μερικές από τις πολλές χρήσεις του υπερκειμένου», και αναφέρει πρώτα από όλες μια εγκυκλοπαίδεια.

2.1.2 Πρώτες προδιαγραφές

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο με το όνομα Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991.

Περιέγραφε τα 20 στοιχεία τα οποία αποτελούσαν τον αρχικό και σχετικά απλό σχεδιασμό της HTML. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν έντονα επηρεασμένες από την SGMLguid, μια μορφή δημιουργίας τεκμηρίωσης, φτιαγμένη στο CERN και βασισμένη στην SGML. Δεκατρία από εκείνα τα αρχικά στοιχεία υπάρχουν ακόμα σήμερα στην HTML 4.

Το ίδιο το πρότυπο SGML αναπαράγει μερικές από τις τεχνικές των τυπογράφων, αλλά εκτός από απλή μίμηση της τυπογραφίας προσθέτει γενικευμένη σήμανση βασισμένη σε στοιχεία, τα οποία μπορούν να εμφωλεύονται το ένα μέσα στο άλλο και να φέρουν ιδιότητες. Ακόμα, το SGML διαχωρίζει τη δομή από το περιεχόμενο, κατεύθυνση προς την οποία αργότερα κινήθηκε και η HTML, με τα CSS. Πολλά από τα στοιχεία κειμένου προέρχονται από την τεχνική αναφορά ISO TR 9537, Techniques for using SGML (τεχνικές χρήσης της SGML), η οποία με τη σειρά της καλύπτει τα χαρακτηριστικά των πρώιμων γλωσσών μορφοποίησης κειμένου που χρησιμοποιούνταν από τα TYPSET και RUNOFF, και είχαν αναπτυχθεί στις αρχές της δεκαετίας του 1960 για το λειτουργικό σύστημα CTSS.

Ο Μπέρνερς Λι θεώρησε την HTML ως μια υλοποίηση του SGML. Αυτό ορίστηκε και επίσημα από το Internet Engineering Task Force (IETF) με τη δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML, στα μέσα του 1993, η οποία περιλάμβανε και έναν Ορισμό τύπου εγγράφου (DTD, Document Type Definition) της SGML, ο οποίος όριζε την γραμματική. Αυτό το πρόχειρο έληξε μετά την πάροδο έξι μηνών, αλλά περιέχει κάτι αξιοσημείωτο: την αναγνώριση της ετικέτας του NCSA Mosaic για την ενσωμάτωση εικόνων μέσα στο κείμενο, η οποία αντικατοπτρίζει την φιλοσοφία του IETF για ενσωμάτωση επιτυχημένων πρωτότυπων μέσα στα πρότυπα. Κάτι παρόμοιο περιείχε και το ανταγωνιστικό πρόχειρο του Dave Raggett, «HTML+ (Hypertext Markup Format)», από τα τέλη του, το οποίο πρότεινε την προτυποποίηση μερικών ήδη υλοποιημένων δυνατοτήτων, όπως οι πίνακες και οι φόρμες.

Μετά που τα πρόχειρα HTML και HTML+ έληξαν, στις αρχές του 1994, το IETF δημιούργησε την Ομάδα Εργασίας για την HTML, η οποία το 1995 ολοκλήρωσε την «HTML 2.0», με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα

βασίζονταν οι μελλοντικές υλοποιήσεις.[10] Η HTML 2.0 δημοσιεύτηκε ως RFC 1866, και περιείχε ιδέες από τα πρόχειρα HTML και HTML+. Η αρίθμηση 2.0 σκόπευε απλά να ξεχωρίσει την νέα έκδοση από τα πρόχειρα που προηγήθηκαν. designation was intended to distinguish the new edition from previous drafts.

Η περαιτέρω ανάπτυξη κάτω από την επίβλεψη του IETF καθυστέρησε λόγω σύγκρουσης ενδιαφερόντων. Από το 1996 και μετά, οι προδιαγραφές της HTML τηρούνται, μαζί με ανάδραση από τους δημιουργούς λογισμικού, από το World Wide Web Consortium (W3C). Ωστόσο, το 2000 η HTML έγινε επίσης παγκόσμιο πρότυπο (ISO/IEC 15445:2000). Η τελευταία προδιαγραφή της HTML, η HTML 4.01 δημοσιεύτηκε από το W3C το 1999, και το 2001 δημοσιεύτηκαν επίσης και τα λάθη και οι παραλείψεις της (errata).

2.1.3 Σήμανση

Η σήμανση HTML αποτελείται από μερικά βασικά συστατικά, συμπεριλαμβανομένων των στοιχείων (και των ιδιοτήτων τους), της βασισμένους σε χαρακτήρες τύπους δεδομένων, της αναφορές χαρακτήρων και της αναφορές οντοτήτων. Ένα ξεχωριστό σημαντικό συστατικό είναι η δήλωση τύπου εγγράφου (document type declaration), η οποία ορίζει στον browser τον τρόπο εμφάνισης της σελίδας.

Στην HTML, το πρόγραμμα Hello world, ένα συνηθισμένο πρόγραμμα υπολογιστή που χρησιμεύει για τη σύγκριση γλωσσών προγραμματισμού, γλωσσών σεναρίων και γλωσσών σήμανσης, φτιάχνεται με 9 γραμμές κώδικα, παρότι οι νέες γραμμές είναι προαιρετικές στη HTML:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Hello HTML</title>
```

```
</head>

<body>

<p>Hello world</p>

</body>

</html>
```

Το κείμενο ανάμεσα στο `<html>` και το `</html>` περιγράφει την ιστοσελίδα, και το κείμενο μεταξύ του `<body>` και του `</body>` είναι το ορατό μέρος της. Το σημασμένο κείμενο `<title>Hello HTML</title>` καθορίζει τον τίτλο που θα εμφανίζεται στην μπάρα τίτλου του browser.

Το Document Type Declaration στον πιο πάνω κώδικα είναι για την HTML 5. Αν δεν συμπεριλαμβάνεται η δήλωση `<!doctype html>`, τότε μερικοί browser θα καταφύγουν στην λειτουργία quirks για την εμφάνιση της σελίδας.

2.1.4 Στοιχεία

Τα έγγραφα HTML αποτελούνται από στοιχεία HTML τα οποία στην πιο γενική μορφή της έχουν τρία συστατικά: ένα ζεύγος από ετικέτες, την «ετικέτα εκκίνησης» και την «ετικέτα τερματισμού», της ιδιότητες μέσα στην ετικέτα εκκίνησης, και τέλος το κείμενο ή το γραφικό περιεχόμενο μεταξύ των ετικετών, το οποίο μπορεί να περιλαμβάνει και άλλα στοιχεία εμφωλευμένα μέσα του. Το στοιχείο HTML μπορεί να είναι οτιδήποτε ανάμεσα της ετικέτες εκκίνησης και τερματισμού. Τέλος, κάθε ετικέτα περικλείεται σε σύμβολα «μεγαλύτερο από» και «μικρότερο από», δηλαδή `<` και `>`.

Επομένως, η γενική μορφή ενός στοιχείου HTML είναι: `<ετικέτα ιδιότητα1=»τιμή1» ιδιότητα2=»τιμή2»>περιεχόμενο</ετικέτα>`. Μερικά στοιχεία HTML περιγράφονται ως άδεια στοιχεία, έχουν τη μορφή `<ετικέτα ιδιότητα1=»τιμή1» ιδιότητα2=»τιμή2»>`, και δεν έχουν καθόλου περιεχόμενο. Το όνομα κάθε στοιχείου HTML είναι το ίδιο όνομα που χρησιμοποιείται της αντίστοιχες ετικέτες. Το όνομα της ετικέτας τερματισμού ξεκινά με μια

κάθετο «/», η οποία παραλείπεται στα άδεια στοιχεία. Τέλος, αν δεν αναφέρονται ρητά οι ιδιότητες ενός στοιχείου, τότε χρησιμοποιούνται οι προεπιλογές σε κάθε περίπτωση.

2.1.5 Παραδείγματα στοιχείων

Κεφαλίδα του εγγράφου HTML: <head>...</head>. Συνήθως περιέχει τον τίτλο, π.χ.:

```
<head>
```

```
<title>Ο τίτλος</title>
```

```
</head>
```

Επικεφαλίδες: οι επικεφαλίδες στην HTML ορίζονται με τις ετικέτες <h1> έως <h6> :

```
<h1>Επικεφαλίδα1</h1>
```

```
<h2>Επικεφαλίδα2</h2>
```

```
<h3>Επικεφαλίδα3</h3>
```

```
<h4>Επικεφαλίδα4</h4>
```

```
<h5>Επικεφαλίδα5</h5>
```

```
<h6>Επικεφαλίδα6</h6>
```

Παράγραφοι:

```
<p>Παράγραφος 1</p> <p>Παράγραφος 2</p>
```

Αλλαγή γραμμής: `
`. Η διαφορά ανάμεσα στο `
` και το `<p>` είναι ότι το «`br`» αλλάζει γραμμή χωρίς να αλλάζει την σημαντική δομή της σελίδας, ενώ το «`p`» τεμαχίζει τη σελίδα σε παραγράφους. Το «`br`» είναι ένα άδειο στοιχείο, δηλαδή δεν έχει περιεχόμενο, ούτε και χρειάζεται ετικέτα τερματισμού.

`<p>`Αυτή `
` είναι μία παράγραφος `
` με `
` αλλαγές γραμμής`</p>`

Της είναι της σύνδεσμος σε HTML: Για τη δημιουργία του χρησιμοποιείται η ετικέτα `<a>`. Η ιδιότητα `href` περιέχει τη σελίδα προορισμού του συνδέσμου.

``Της σύνδεσμος στη Βικιπαίδεια!``

Σχόλια:

`<!--Αυτό είναι ένα σχόλιο -->`

Τα σχόλια μπορούν να βοηθήσουν στην κατανόηση της σήμανσης. Δεν εμφανίζονται στην ιστοσελίδα.

Υπάρχουν διάφοροι τύποι στοιχείων στην HTML.

Η δομική σήμανση περιγράφει τον σκοπό του κειμένου

Για παράδειγμα, το `<h2>Golf</h2>` σημαίνει ότι η λέξη «Golf» θα είναι μια επικεφαλίδα δεύτερου επιπέδου. Η δομική σήμανση δεν εμπεριέχει συγκεκριμένο τρόπο εμφάνισης, αλλά

οι περισσότεροι φυλλομετρητές έχουν δικό της προκαθορισμένο τρόπο για τη μορφοποίηση όλων των στοιχείων. Το περιεχόμενο μπορεί να μορφοποιηθεί οπτικά με τη χρήση των CSS.

Η σήμανση οπτικής μορφοποίησης περιγράφει την εμφάνιση του κειμένου, άσχετα από τον σκοπό του.

Για παράδειγμα το `έντονο κείμενο` υποδεικνύει ότι οι συσκευές που θα εμφανίσουν το κείμενο θα το κάνουν έντονο, αλλά δεν λέει τι θα κάνουν με το κείμενο οι συσκευές που δεν μπορούν να το εμφανίσουν, της για παράδειγμα οι συσκευές φωνητικής ανάγνωσης σελίδων. Τόσο στην περίπτωση του `έντονου` όσο και του `<i>πλάγιου</i>`, υπάρχουν άλλα στοιχεία τα οποία μπορεί οπτικά να εμφανίζονται ίδια, αλλά είναι πιο σημαντικά, της το `σημαντικό κείμενο` και το `κείμενο με έμφαση` αντίστοιχα. Είναι ευκολότερο να ερμηνεύσει της φωνητικός φυλλομετρητής τα τελευταία δύο στοιχεία. Ωστόσο, δεν είναι ισότιμα με τα αντίστοιχα στοιχεία οπτικής μορφοποίησης. Για παράδειγμα ένα πρόγραμμα φωνητικής ανάγνωσης της σελίδας δεν θα πρέπει να προφέρει με έμφαση τον τίτλο της βιβλίου, αλλά όταν το ίδιο περιεχόμενο εμφανίζεται σε μια οθόνη τότε ο τίτλος θα είναι με πλάγια γράμματα. Τα περισσότερα στοιχεία οπτικής μορφοποίησης έχουν θεωρηθεί ξεπερασμένα μετά την προδιαγραφή HTML 4.0, και έχουν αντικατασταθεί από τα CSS.

Η σήμανση υπερκειμένου κάνει μερικά τμήματα της εγγράφου να συνδέουν με άλλα έγγραφα.

Το στοιχείο `anchor` δημιουργεί έναν υπερσύνδεσμο στο έγγραφο, και η ιδιότητα `href` ορίζει τον στόχο του συνδέσμου. Για παράδειγμα, η σήμανση HTML `Βικιπαίδεια` εμφανίζει το κείμενο «Βικιπαίδεια» ως υπερσύνδεσμο. Για την εμφάνιση μιας εικόνας ως συνδέσμου, μπορεί να εισαχθεί ένα στοιχείο `` ως περιεχόμενο του στοιχείου «a». Της και το «b», το «img» είναι άδειο στοιχείο. Έχει ιδιότητες αλλά δεν έχει περιεχόμενο, ούτε ετικέτα τερματισμού: ``.

2.1.6 Ιδιότητες

Οι περισσότερες ιδιότητες των στοιχείων είναι ζεύγη ονομάτων και τιμών, τα οποία διαχωρίζονται με ένα «=» και γράφονται μέσα στην ετικέτα εκκίνησης της στοιχείου, μετά το όνομα του στοιχείου. Η τιμή μπορεί να περικλείεται σε μονά ή διπλά εισαγωγικά, παρότι τιμές που αποτελούνται από συγκεκριμένους χαρακτήρες μπορούν να γράφονται χωρίς εισαγωγικά στην HTML, αλλά όχι στην XHTML. Το να μένουν οι τιμές των ιδιοτήτων χωρίς εισαγωγικά θεωρείται ανασφαλές. Εκτός από της ιδιότητες που γράφονται ως ζεύγη ονομάτων και τιμών, υπάρχουν και της οι οποίες επηρεάζουν το στοιχείο απλά με την παρουσία της μέσα στην ετικέτα εκκίνησης, της η ιδιότητα `ismap` του στοιχείου `img`.

Υπάρχουν και της κοινές ιδιότητες οι οποίες εμφανίζονται σε πολλά στοιχεία:

Η ιδιότητα `id` παρέχει ένα αναγνωριστικό για ένα στοιχείο το οποίο είναι μοναδικό σε ολόκληρο το έγγραφο. Χρησιμοποιείται για να ταυτοποιεί το στοιχείο ώστε τα CSS να μπορούν να αλλάξουν τον τρόπο που αυτό εμφανίζεται, καθώς και τα σενάρια μπορούν να αλλάξουν, να μετακινήσουν ή να διαγράψουν τα περιεχόμενα ή την εμφάνισή του. Ακόμα, αν ένα `id` προστεθεί στο URL μιας σελίδας, παρέχει ένα μοναδικό αναγνωριστικό για ένα τμήμα της σελίδας, για παράδειγμα: <http://el.wikipedia.org/wiki/HTML#Ιδιότητες>

Η ιδιότητα `class` παρέχει τη δυνατότητα ταξινόμησης παρόμοιων αντικειμένων στην ίδια κλάση. Μπορεί να χρησιμοποιηθεί για να να αποδώσει κάποια σημασία στο στοιχείο, ή για σκοπούς εμφάνισης. Για παράδειγμα, ένα έγγραφο HTML μπορεί να χρησιμοποιεί την επισήμανση `class=>notation` σε μερικά στοιχεία για να ξεχωρίσει από το υπόλοιπο κείμενο του εγγράφου. Κατά την εμφάνιση του εγγράφου, αυτά τα στοιχεία μπορεί –για παράδειγμα– να εμφανίζονται όλα μαζί στο τέλος της σελίδας ως υποσημειώσεις, άσχετα με την θέση που εμφανίζονται μέσα στον κώδικα. Της οι ιδιότητες `class` χρησιμοποιούνται σημασιολογικά στα `microformat`. Ένα στοιχείο μπορεί να έχει πολλαπλές κλάσεις, για παράδειγμα το `class=>notation important` βάζει το στοιχείο τόσο στην κλάση «`notation`» όσο και στην «`important`».

Η ιδιότητα `style` εφαρμόζει στυλ εμφάνισης σε συγκεκριμένα στοιχεία. Θεωρείται καλύτερη τακτική να χρησιμοποιούνται οι ιδιότητες `id` ή `class` ώστε να επιλέγεται το στοιχείο μέσα σε

ένα CSS, αλλά της φορές μπορεί να είναι πιο απλό να ανατεθούν style κατευθείαν στο στοιχείο.

Η ιδιότητα title προσθέτει μια εξήγηση στο στοιχείο στο οποίο εφαρμόζεται. Της περισσότερους φυλλομετρητές αυτή η ιδιότητα εμφανίζεται ως αναδυόμενο παράθυρο βοήθειας.

Η ιδιότητα lang ταυτοποιεί την φυσική γλώσσα των περιεχομένων του στοιχείου, η οποία μπορεί να είναι διαφορετική από το υπόλοιπο έγγραφο. Για παράδειγμα, μέσα σε ένα έγγραφο μπορεί να υπάρχει το κείμενο:

```
<p>Τι να κάνουμε; <span lang=»fr»>c'est la vie</span>, της λένε και οι Γάλλοι.</p>
```

Το στοιχείο abbr μπορεί να χρησιμοποιηθεί για να επιδείξει της από της πιο πάνω ιδιότητες:

```
<abbr id=»anId» class=»jargon» style=»color:purple;» title=»Hypertext Markup Language»>HTML</abbr>
```

Αυτό το παράδειγμα εμφανίζεται ως: HTML. Της περισσότερους φυλλομετρητές, όταν σταθεί ο δείκτης του ποντικιού πάνω από αυτό το στοιχείο, τότε θα εμφανιστεί ένα αναδυόμενο παράθυρο με το περιεχόμενο «Hypertext Markup Language».

Τα περισσότερα στοιχεία μπορούν της να δεχτούν την ιδιότητα dir η οποία καθορίζει την κατεύθυνση του κειμένου. Η τιμή «rtl» της ιδιότητας, δείχνει κείμενο που διαβάζεται από δεξιά της τα αριστερά, της για παράδειγμα στα Αραβικά, στα Περσικά ή στα Εβραϊκά.

2.1.7 Αναφορές οντοτήτων χαρακτήρων

Από την έκδοση 4.0 και μετά, στην HTML ορίζεται ένα σύνολο από 252 αναφορές οντοτήτων χαρακτήρων και ένα σύνολο από 1.114.050 αναφορές οντοτήτων αριθμών. Και τα δύο σύνολα επιτρέπουν τη γραφή μοναδικών χαρακτήρων ως σήμανση, αντί χρησιμοποιώντας της της της χαρακτήρες. Της χαρακτήρας και η αντίστοιχη σήμανση γι' αυτόν θεωρούνται ισοδύναμες οντότητες, και εμφανίζονται ίδια.

Αυτή η δυνατότητα διαφυγής χαρακτήρων με αυτόν τον τρόπο επιτρέπει την ερμηνεία χαρακτήρων της < και & (όταν γράφονται ως < Και &, αντίστοιχα) ως κείμενο και όχι ως σήμανση. Για παράδειγμα, ο χαρακτήρας < κανονικά σημαίνει την έναρξη μιας ετικέτας, και το & σημαίνει την έναρξη μιας αναφοράς οντότητας χαρακτήρα ή αριθμού. Η γραφή της ως & Ή & Ή & Επιτρέπει στο & να συμπεριλαμβάνεται μέσα στο περιεχόμενο της στοιχείου ή μέσα στην τιμή μιας ιδιότητας. Της, ο χαρακτήρας διπλών εισαγωγικών («), ο οποίος χρησιμοποιείται για να περιβάλλει την τιμή μιας ιδιότητας, πρέπει να γράφεται ως " Ή " Ή " Όταν εμφανίζεται μέσα στην ίδια την τιμή της ιδιότητας. Ομοίως, ο χαρακτήρας μονών εισαγωγικών ('), πρέπει να γράφεται ως ' Ή ' (αλλά όχι ως ', με εξαίρεση την γραφή σε XHTML) όταν εμφανίζεται μέσα στην ίδια την τιμή της ιδιότητας. Αν της δημιουργός εγγράφου HTML παραβλέψει να γράψει αυτά τα σύμβολα ως οντότητες χαρακτήρων ή αριθμών, τότε μερικοί browser μπορούν να είναι αρκετά επιεικείς και προσπαθούν να χρησιμοποιήσουν το παραπλήσιο περιεχόμενο για να μαντέψουν την πρόθεση του χρήστη. Το αποτέλεσμα δεν παύει να είναι άκυρη σήμανση, η οποία κάνει το έγγραφο λιγότερο προσιτό σε της browser οι οποίοι μπορεί να προσπαθήσουν να αναλύσουν το έγγραφο για της σκοπούς, της για παράδειγμα για την καταχώρηση του περιεχομένου του σε μηχανές αναζήτησης.

Η διαφυγή διευκολύνει της τη χρήση χαρακτήρων που είτε είναι δύσκολο να δακτυλογραφηθούν, είτε δεν συμπεριλαμβάνονται καν στην κωδικοποίηση χαρακτήρων του εγγράφου. Για παράδειγμα, ο τονισμένος χαρακτήρας e (δηλαδή το é), ο οποίος τυπικά συναντάται σε της ευρωπαϊκές γλώσσες, μπορεί να γραφεί σε οποιοδήποτε έγγραφο HTML ως η οντότητα é Ή ως η αριθμητική αναφορά é Ή é, χρησιμοποιώντας χαρακτήρες οι οποίοι είναι διαθέσιμοι σε όλα τα πληκτρολόγια και υποστηρίζονται σε της

της κωδικοποιήσεις χαρακτήρων. Οι κωδικοποιήσεις Unicode, της η UTF-8, είναι συμβατές με όλους της σύγχρονους browser και επιτρέπουν την πρόσβαση σε σχεδόν όλους της χαρακτήρες όλων των συστημάτων γραφής του κόσμου.

2.1.8 Τύποι δεδομένων

Στην HTML ορίζονται μερικοί τύποι δεδομένων για το περιεχόμενο των στοιχείων, της σενάρια εντολών ή stylesheet, και μια πληθώρα τύπων για της τιμές των ιδιοτήτων, συμπεριλαμβανομένων των ID, των name, των URI, διαφόρων αριθμών και μονάδων μήκους, γλωσσών, τύπων αρχείων πολυμέσων, χρωμάτων, κωδικοποιήσεων χαρακτήρων, ημερομηνιών κλπ.

2.1.9 Δήλωση τύπου εγγράφου

Τα έγγραφα HTML πρέπει να αρχίζουν με μια Δήλωση τύπου εγγράφου (Document Type Declaration, ανεπίσημα λέγεται και «doctype»). Αυτή η δήλωση βοηθά της browser να καταλάβουν πώς πρέπει να διαβάσουν το περιεχόμενο του εγγράφου και πώς να το παρουσιάσουν μετά, και ιδιαίτερα όταν χρησιμοποιείται το quirks mode.

Ο αρχικός σκοπός του doctype ήταν να επιτρέπει την ανάλυση και επιβεβαίωση των εγγράφων HTML από εργαλεία SGML τα οποία ήταν βασισμένα στο Document Type Definition (DTD). Το DTD στο οποίο αναφέρεται το DOCTYPE περιέχει μια γραμματική σε γλώσσα μηχανής, η οποία καθορίζει τι επιτρέπεται και τι απαγορεύεται να υπάρχει μέσα στο έγγραφο. Οι browser, από την άλλη, δεν υλοποιούν την HTML ως εφαρμογή της SGML και συνεπώς δεν διαβάζουν το DTD. Στην HTML5 δεν ορίζεται κανένα DTD, λόγω εγγενών τεχνολογικών περιορισμών. Έτσι το doctype `<!doctype html>` δεν αναφέρεται σε κανένα.

Ένα παράδειγμα doctype σε HTML 4 είναι:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Αυτή η δήλωση αναφέρεται στο DTD για την εκδοχή «strict» της HTML 4.01. Τα προγράμματα επιβεβαίωσης κώδικα τα οποία βασίζονται σε SGML, διαβάζουν το DTD με σκοπό να ερμηνεύσουν κατάλληλα το έγγραφο και να επαληθεύσουν την ορθότητά του. Της σύγχρονους browser, ένα έγκυρο doctype ενεργοποιεί τη λειτουργία κατά τα πρότυπα, αντί για το quirks mode.

Επιπλέον, στην HTML 4.01 παρέχονται και τα DTD Transitional και Frameset.

2.1.10 Σημασιολογική

Η Σημασιολογική HTML είναι της τρόπος γραφής της HTML με έμφαση στο νόημα της σημασμένης πληροφορίας παρά στην εμφάνισή της. Η HTML συμπεριλαμβάνει σημασιολογικά στοιχεία από την σύλληψή της, συμπεριλαμβάνει της και στοιχεία αποκλειστικά εμφανισιακά, της της ετικέτες , <i> και <center>. Υπάρχουν της και τα σημασιολογικά ουδέτερα span και div. Από τα τέλη της δεκαετίας του 1990, όταν τα CSS άρχισαν να δουλεύουν της περισσότερους browser, οι σχεδιαστές ιστοσελίδων ενθαρρύνονταν να αποφεύγουν τη χρήση των εμφανισιακών ετικετών της HTML και να προτιμούν να διαχωρίζουν το περιεχόμενο από την παρουσίαση.[53]

Σε μια συζήτηση για τον Σημασιολογικό ιστό, ο Τιμ Μπέρνερς Λι και άλλοι έδωσαν παραδείγματα τρόπων με της οποίους έξυπνα λογισμικά πελάτη ιστού θα μπορούν μια μέρα να περιηγούνται στον Ιστό και να βρίσκουν, να φιλτράρουν και να συσχετίζουν τα προηγουμένως άσχετα και αδημοσίευτα δεδομένα, της όφελος των ανθρώπων χρηστών. Δέκα χρόνια μετά, δεν είναι ακόμα κοινή η χρήση τέτοιων προγραμμάτων, αλλά της από της ιδέες του Web 2.0, των mashups και των ιστοσελίδων σύγκρισης τιμών προϊόντων μπορεί να πλησιάζουν στο σκεπτικό εκείνο. Η βασική διαφορά μεταξύ αυτών των εφαρμογών και των έξυπνων λογισμικών του Μπέρνερς Λι, είναι ότι η τρέχουσες μέθοδοι συγκέντρωσης

πληροφοριών συνήθως σχεδιάζεται από προγραμματιστές ιστού, οι οποίοι ήδη ξέρουν της τοποθεσίες ιστού και τα API των δεδομένων που θέλουν να συνενώσουν, να συγκρίνουν και να συνδυάσουν.

Της σημαντικός τύπος λογισμικοί που περιηγείται αυτόματα το διαδίκτυο και διαβάζει ιστοσελίδες, χωρίς προηγούμενη γνώση του περιεχομένου της, είναι και τα προγράμματα των μηχανών αναζήτησης. Αυτά τα λογισμικά εξαρτώνται από την σημασιολογική σαφήνεια των ιστοσελίδων που βρίσκουν, καθώς χρησιμοποιούν διάφορες τεχνικές και αλγόριθμους για να διαβάζουν και να ταξινομούν εκατομμύρια σελίδων καθημερινά, και να παρέχουν της επισκέπτες της τη δυνατότητα αναζήτησης, χωρίς την οποία ο Παγκόσμιος Ιστός θα είχε μόνο ένα μικρό κλάσμα της χρησιμότητάς του.

Οι σημασιολογικές δομές που ήδη υπάρχουν στην HTML, είναι σημαντικό να εφαρμόζονται καθολικά, για να βοηθούν το περιεχόμενο του δημοσιευμένου κειμένου να γίνεται καλύτερα αντιληπτό. Με αυτό τον τρόπο, τα λογισμικά των μηχανών αναζήτησης, αλλά και άλλα λογισμικά που φτιάχνουν mashup ή άλλα υβρίδια από το περιεχόμενο των ιστότοπων θα μπορούν καλύτερα να εκτιμούν τη σημασία του κειμένου που βρίσκουν στα έγγραφα HTML. Της το σκοπό αυτό, οι ετικέτες της HTML που εξυπηρετούσαν μόνο εμφανισιακό σκοπό θεωρούνται ξεπερασμένες στην XHTML, και απαγορεύονται στην HTML5.

Τέλος, η σημασιολογικά γραμμένη HTML βελτιώνει την προσιτότητα των εγγράφων στον ιστό. Για παράδειγμα, της browser για χρήστες με προβλήματα όρασης ή ακοής, είναι πιο εύκολη η σωστή απόδοση της δομής του εγγράφου με τρόπο άλλο από τον οπτικό, όταν αυτό είναι σωστά σημασιολογικά σημασμένο.

2.1.11 Μεταφορά της HTML

Τα αρχεία HTML, της συμβαίνει και με της υπόλοιπους τύπους αρχείων του υπολογιστή, μπορούν να μεταφερθούν με πολλούς τρόπους. Ωστόσο, λόγω της λειτουργίας της HTML, οι δύο πιο συνηθισμένοι τρόποι είναι μέσω HTTP από έναν εξυπηρετητή, ή μέσω e-mail.

HTTP

Κύριο λήμμα: HTTP

Ο Παγκόσμιος Ιστός αποτελείται κυρίως από αρχεία HTML τα οποία μεταφέρονται από εξυπηρετητές της browsers χρησιμοποιώντας το πρωτόκολλο HTTP. Ωστόσο, μια που το ίδιο πρωτόκολλο μπορεί επιπλέον να μεταφέρει εικόνες, ήχο και άλλο περιεχόμενο, χρειάζεται της τρόπος αναγνώρισης του τύπου του περιεχομένου που μεταφέρεται. Έτσι, μαζί με το ίδιο το αρχείο, μεταφέρονται και της επιπλέον πληροφορίες, ή μεταδεδομένα, μεταξύ αυτών και ο τύπος MIME (για παράδειγμα text/html ή application/xhtml+xml) καθώς και η κωδικοποίηση χαρακτήρων που χρησιμοποιείται.

Της σύγχρονους browser, ο τύπος MIME που συνοδεύει το αρχείο HTML μπορεί να επηρεάζει τον τρόπο που αυτό εμφανίζεται. Για παράδειγμα, ένα αρχείο το οποία συνοδεύεται από έναν τύπο XHTML MIME, αναμένεται να είναι γραμμένο σε γλώσσα σωστά διατυπωμένη, κατά τα πρότυπα της XML. Αν δεν είναι, τότε τα σφάλματα στον κώδικα μπορεί να αποτρέψουν την ορθή απεικόνιση του αρχείου. Επειδή η XHTML 1.0 και η XHTML 1.1 είναι πάντα συμβατές με την XML, το W3C υποστηρίζει ότι δεν παίζει ρόλο ποιος τύπος MIME θα χρησιμοποιηθεί.

HTML e-mail

Αρκετά προγράμματα ηλεκτρονικής αλληλογραφίας ενσωματώνουν δυνατότητες της HTML, ώστε να επιτρέπουν της χρήστες να χρησιμοποιούν την σήμανση της HTML στα μηνύματά της, για να στέλνουν κείμενο με χρώμα, με μορφοποίηση, ή και με ενσωματωμένες εικόνες και διαγράμματα. Ωστόσο, οι μέθοδοι αυτές προκαλούν μερικά προβλήματα, τα οποία προκαλούνται αφ' της από την έλλειψη κάποιου προτύπου για την περίληψη HTML σε e-mail (με αποτέλεσμα ο κάθε προγραμματιστής να ενεργεί κατά βούληση), και αφ' ετέρου από τη δυνατότητα εκμετάλλευσης αυτών των δυνατοτήτων από κακόβουλους χρήστες.

Ονόματα αρχείων

Ο πιο κοινός τύπος αρχείου για έγγραφα HTML είναι .html, της έχει επιβιώσει και η συντόμευση .htm, από μερικά παλαιότερα λειτουργικά συστήματα που δεν αναγνώριζαν επεκτάσεις αρχείων με περισσότερα από τρία γράμματα. Επιπλέον, στα Microsoft Windows χρησιμοποιείται ο τύπος .hta (από το HTML Application) ο οποίος δείχνει ότι το αρχείο συμπεριλαμβάνει HTML μαζί με κάποια δυναμικά στοιχεία που το κάνουν να εκτελείται ως εφαρμογή.

2.1.12 Δημιουργία με προγράμματα WYSIWYG

Ένα θέμα αντιπαράθεσης της κοινότητας των δημιουργών περιεχομένου για τον Ιστό είναι η χρήση προγραμμάτων WYSIWIG (What You See Is What You Get), δηλαδή προγραμμάτων στα οποία ο χρήστης σχεδιάζει οπτικά τη σελίδα της θα φαίνεται μετά στον browser του επισκέπτη. Αυτό κάνει τη γνώση της HTML προαιρετική για την κατασκευή μιας ιστοσελίδας. Αυτό το μοντέλο δημιουργίας με προγράμματα WYSIWIG έχει γνωρίσει έντονες κριτικές, κυρίως λόγω της κακής ποιότητας του κώδικα που δημιουργείται αυτόματα.

Παρόλα αυτά, οι επεξεργαστές ιστοσελίδων WYSIWYG είναι δημοφιλείς λόγω της ευκολίας που παρέχουν.

2.2 CSS

Τα Διαδοχικά Φύλλα Στυλ (CSS, Cascading Style Sheets) αποτελούν ένα πολύ καλό εργαλείο για να μπορούμε να αλλάζουμε την εμφάνιση και τη διάταξη (layout) των ιστοσελίδων μας. Μπορούν να μας γλυτώσουν από πολύ χρόνο και κόπο και μας δίνουν τη δυνατότητα να σχεδιάζουμε τις ιστοσελίδες μας με μια εντελώς καινούργια φιλοσοφία. Η κατανόηση των CSS απαιτεί να υπάρχει κάποια βασική εμπειρία με την HTML.

Για να δουλέψουμε με τα CSS δεν χρειάζεται να χρησιμοποιήσουμε κάποιο πρόγραμμα δημιουργίας ιστοσελίδων, όπως είναι τα γνωστά FrontPage, DreamWeaver ή και το Word, καθώς θα σταθούν εμπόδιο στην κατανόηση των CSS. Αυτό που χρειαζόμαστε είναι ένας απλός επεξεργαστής κειμένου, όπως είναι το Σημειωματάριο (Notepad) των Windows.

Μπορούμε να χρησιμοποιήσουμε όποιον φυλλομετρητή (browser) επιθυμούμε για να βλέπουμε πώς θα εμφανίζονται οι ιστοσελίδες που θα δημιουργούμε. Συνεπώς αυτό που

χρειαζόμαστε είναι ένας φυλλομετρητής (browser) και ένας απλός επεξεργαστής κειμένου (text editor).

Στο ερώτημα τι μπορούμε να κάνουμε με τα CSS, θα πρέπει να γνωρίζουμε ότι τα CSS είναι μια γλώσσα στυλ (style language) που ορίζουν τη διάταξη (layout) των HTML εγγράφων. Για παράδειγμα, τα CSS έχουν να κάνουν με γραμματοσειρές (fonts), με χρώματα (colours), με περιθώρια (margins), με εικόνες φόντου (background images) και με πολλά άλλα. Με την HTML θα δυσκολευτούμε να αλλάξουμε τη διάταξη των ιστοσελίδων μας, αλλά τα CSS προσφέρουν πολλές επιλογές και είναι πολύ πιο συγκεκριμένα στις λεπτομέρειες. Επιπλέον, υποστηρίζονται απ' όλους τους φυλλομετρητές.

Στο ερώτημα ποια είναι η διαφορά ανάμεσα στα CSS και την HTML, μπορούμε να πούμε ότι η HTML χρησιμοποιείται για να δομήσει το περιεχόμενο (content), ενώ τα CSS χρησιμοποιούνται για τη διαμόρφωση ή μορφοποίηση (formatting) του δομημένου περιεχομένου. Σύντομα θα γίνει σαφές το τι εννοούμε. Από το ξεκίνημά της, η HTML χρησιμοποιείτο μόνο για να μπορούμε να προσθέτουμε δομή στο κείμενο, όπως για να επισημαίνουμε ποια είναι κεφαλίδα (headline) ή ποια είναι παράγραφος (paragraph) με τα γνωστά tags (ετικέτες) της HTML, όπως είναι τα <h1> και <p>.

Όμως, καθώς η δημοτικότητα του Web αύξανε συνεχώς, οι σχεδιαστές των ιστοσελίδων άρχισαν να ψάχνουν για επιπλέον δυνατότητες, όπως προσθήκη διάταξης (layout) στα έγγραφα. Για να γίνει αυτό, οι φυλλομετρητές επινόησαν καινούργια HTML tags, όπως για παράδειγμα το , τα οποία διέφεραν από τα αρχικά HTML tags καθώς όριζαν τη διάταξη και όχι τη δομή μιας ιστοσελίδας.

Στο ερώτημα τι έχουν να μου προσφέρουν τα CSS, μπορούμε να πούμε ότι τα CSS αποτέλεσαν μια επανάσταση στον κόσμο του Web design και τα μεγάλα πλεονεκτήματά τους είναι τα εξής :

- Δυνατότητα για τον ταυτόχρονο έλεγχο της διάταξης (layout) πολλών εγγράφων από ένα μόνο φύλλο στυλ (style sheet).
- Πιο ακριβής έλεγχος της διάταξης (layout) των ιστοσελίδων.
- Εφαρμογή διαφορετικής διάταξης σε διαφορετικές εξόδους, όπως screen, print κ.ά.
- Αναρίθμητες προηγμένες και εξεζητημένες τεχνικές.

2.2.1 Η Βασική Σύνταξη των CSS

Θα δούμε τώρα πώς μπορούμε να δημιουργήσουμε το πρώτο μας φύλλο στυλ (style sheet). Θα πρέπει να έχουμε υπόψη μας ότι πολλές από τις ιδιότητες που χρησιμοποιούνται στα CSS είναι παρόμοιες μ' αυτές της HTML. Ας υποθέσουμε ότι θέλουμε να έχουμε ένα ωραίο κόκκινο χρώμα για το φόντο (background) μιας ιστοσελίδας. Με την HTML θα είχαμε γράψει τα εξής :

```
<body bgcolor="#FF0000">
```

Με τα CSS μπορούμε να επιτύχουμε το ίδιο αποτέλεσμα ως εξής :

```
body {  
  
background-color: #FF0000;  
  
}
```

Όπως μπορούμε να δούμε, οι κωδικοί είναι περίπου οι ίδιοι και για την HTML και για τα CSS. Από το παραπάνω παράδειγμα μπορούμε να δούμε και το βασικό μοντέλο των CSS :

```
selector ( επιλογέας ) {  
  
property ( ιδιότητα ): value (τιμή);  
  
}
```

selector	property	value
Σε ποια tags εφαρμόζεται	Μια ιδιότητα είναι το χρώμα	Η τιμή της ιδιότητας, όπως π.χ. κόκκινο
η ιδιότητα (π.χ. body)	φόντου (background color)	για το χρώμα φόντου

2.2.2 Εφαρμογή των CSS σε ένα HTML έγγραφο

Υπάρχουν τρεις τρόποι που μπορούμε να χρησιμοποιήσουμε για να εφαρμόσουμε ένα CSS σ' ένα HTML έγγραφο.

Μέθοδος 1 : In-line (attribute style)

Ο πρώτος τρόπος για να εφαρμόσουμε ένα CSS σ' ένα HTML κείμενο είναι να χρησιμοποιήσουμε την ιδιότητα (attribute) style της HTML. Το είδαμε ήδη προηγουμένως με το κόκκινο χρώμα φόντου και μπορεί να εφαρμοσθεί ως εξής:

```
<html>
<head>
<title>Παράδειγμα</title>
</head>
<body style="background-color: #ff0000;">
<p>Αυτή είναι μια κόκκινη σελίδα</p>
</body>
</html>
```

Μέθοδος 2 : Εσωτερικό-Internal (tag style)

Ένας άλλος τρόπος είναι να γράψουμε τον κώδικα για τα CSS με το HTML tag <style>, ως εξής :

```
<html>
<head>
<title>Παράδειγμα</title>
<style type="text/css">
```

```

body {

background-color: #ff0000;

}

</style>

</head>

<body>

<p>Αυτή είναι μια κόκκινη σελίδα</p>

</body>

</html>

```

Μέθοδος 3 : Εξωτερικό-External (link to a style sheet)

Ο τρίτος τρόπος, που είναι αυτός που συνιστάται να χρησιμοποιούμε, είναι ένας σύνδεσμος (link) προς ένα εξωτερικό αρχείο φύλλου στυλ (external style sheet). Αυτή η μέθοδος θα χρησιμοποιηθεί σ' όλα τα παραδείγματα που ακολουθούν. Ένα εξωτερικό αρχείο φύλλου στυλ (external style sheet) είναι απλά ένα αρχείο κειμένου (text file) που έχει επέκταση (extension) .css. Μπορούμε να το τοποθετήσουμε (upload) στον Web server, όπως όλα τα άλλα αρχεία.

Για παράδειγμα, ας υποθέσουμε ότι το αρχείο μας style sheet έχει όνομα style.css και βρίσκεται στον φάκελο style. Για να δημιουργήσουμε έναν σύνδεσμο (link) από το HTML έγγραφο, που είναι το default.htm, προς το αρχείο του style sheet, που είναι το style.css, θα πρέπει να γράψουμε τον εξής κώδικα :

```
<link rel="stylesheet" type="text/css" href="style/style.css" />
```

Η διαδρομή (path) προς το style sheet προσδιορίζεται με την ιδιότητα (attribute) href. Η παραπάνω γραμμή κώδικα πρέπει να τοποθετηθεί στο τμήμα header του HTML κώδικα, δηλ. ανάμεσα στα tags <head> και </head>, ως εξής :

```
<html>
```

```
<head>

<title>Κύρια Ιστοσελίδα</title>

<link rel="stylesheet" type="text/css" href="style/style.css" />

</head>

<body>
```

Ο παραπάνω σύνδεσμος (link) λέει στον φυλλομετρητή ότι θα πρέπει να χρησιμοποιήσει τη διάταξη (layout) που υπάρχει στο συγκεκριμένο CSS αρχείο όταν πρόκειται να εμφανίσει το περιεχόμενο του τρέχοντος HTML αρχείου.

Αυτό που είναι πολύ καλό με τη μέθοδο αυτή είναι ότι μπορούμε να συνδέσουμε πολλά HTML αρχεία με το ίδιο αρχείο style sheet. Μ' άλλα λόγια, ένα αρχείο CSS μπορεί να χρησιμοποιηθεί για να μπορούμε να ελέγχουμε το layout πολλών HTML εγγράφων ταυτόχρονα.

Για να το δούμε αυτό και στην πράξη, μπορούμε να δημιουργήσουμε δύο αρχεία, ένα HTML και ένα CSS, με το εξής περιεχόμενο :

```
default.htm

<html>

<head>

<title>Κύρια Ιστοσελίδα</title>

<link rel="stylesheet" type="text/css" href="style.css" />

</head>

<body>

<h1>Το Πρώτο μας Φύλλο Στυλ</h1>

</body>

</html>
```

```
style.css

body {

background-color: #ff0000;

}
```

Θα πρέπει να δημιουργήσουμε τα αρχεία και να τα αποθηκεύσουμε με τις σωστές επεκτάσεις .htm και .css και φυσικά να τα τοποθετήσουμε και στον ίδιο φάκελο. Μετά, μόλις ανοίξουμε την ιστοσελίδα default.htm με τον φυλλομετρητή μας, θα διαπιστώσουμε ότι θα έχει κόκκινο φόντο.

2.3 XML

Η XML (αγγλ. Αρκτ. Από το Extensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες της σχετικές προδιαγραφές ανοιχτών προτύπων.

Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για της της γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα της υπηρεσίες ιστού.

Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML.

Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για της περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork της εταιρίας Apple.

2.3.1 Χαρακτήρας Unicode

Εξ ορισμού, ένα κείμενο XML είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

2.3.2 Επεξεργαστής και Εφαρμογή

Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Είναι αναμενόμενο, ότι ένας επεξεργαστής δουλεύει για μία εφαρμογή. Υπάρχουν μερικές πολύ συγκεκριμένες απαιτήσεις, σχετικά με το τι μπορεί και τι δεν μπορεί να κάνει ένας επεξεργαστής XML, αλλά καμία, όσον αφορά στη συμπεριφορά της εφαρμογής. Ο επεξεργαστής (όπως ονοματίζεται από την προδιαγραφή), αναφέρεται συχνά, με τον αγγλικό όρο XML parser.

2.3.3 Σήμανση και Περιεχόμενο

Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML, αποτελούν είτε τη σήμανση είτε το περιεχόμενό του. Η σήμανση και το περιεχόμενο, μπορούν να επισημανθούν και να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά που συνιστούν τη σήμανση, είτε ξεκινούν με το χαρακτήρα «<» και καταλήγουν στο χαρακτήρα «>», είτε ξεκινούν με το χαρακτήρα «&» και καταλήγουν στο χαρακτήρα «;». Ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση, αποτελούν το περιεχόμενο ενός κειμένου XML.

2.3.4 Ετικέτα

Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα "<" και καταλήγει στο χαρακτήρα ">". Υπάρχουν τρία είδη ετικέτας: ετικέτες-αρχής, για παράδειγμα <section>, ετικέτες-τέλους, για παράδειγμα </section>, και ετικέτες-χωρίς-περιεχόμενο, για παράδειγμα <line-break/>.

2.3.5 Στοιχείο

Ένα λογικό απόσπασμα ενός κειμένου, που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας-αρχής και μιας ετικέτας-τέλους, συνιστούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων, που ονομάζονται στοιχεία-παιδιά. Ένα παράδειγμα ενός στοιχείου είναι το `<Greeting>Hello, world.</Greeting>`. Ένα άλλο είναι το `<line-break/>`.

2.3.6 Χαρακτηριστικό

Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι όνομα/τιμή, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο. Στο παράδειγμα παρακάτω, το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`: ``. Ένα άλλο παράδειγμα θα ήταν το `<step number=»3»>Connect A to B.</step>`, όπου το όνομα του χαρακτηριστικού είναι «number» και η τιμή του είναι «3».

2.3.7 Δήλωση XML

Τα κείμενα XML μπορούν να αρχίζουν, με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως στο ακόλουθο παράδειγμα:

```
<?xml version="1.0" encoding="UTF-8"?>
```

2.3.8 Παράδειγμα

Το παρακάτω είναι ένα μικρό, αλλά πλήρες κείμενο XML, που χρησιμοποιεί όλες τις παραπάνω έννοιες και στοιχεία.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<painting>

<caption>This is Raphael's "Foligno" Madonna, painted in
<date>1511</date>-<date>1512</date>.</caption>
</painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το κείμενο του παραδείγματος: τα `painting`, `img`, `caption`, και δύο `date`. Τα στοιχεία `date`, είναι παιδιά του στοιχείου `caption`, το οποίο είναι παιδί του στοιχείου-ρίζας `painting`. Το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`.

2.3.9 Χαρακτήρες και διαφυγή

Τα κείμενα XML αποτελούνται εξ ολοκλήρου από χαρακτήρες Unicode. Εκτός από ένα μικρό αριθμό, ειδικά εξαιρούμενων χαρακτήρων ελέγχου, κάθε χαρακτήρας που ορίζεται στο Unicode, μπορεί να εμφανίζεται στο περιεχόμενο ενός κειμένου XML. Το σύνολο των χαρακτήρων που μπορούν να εμφανίζονται στη σήμανση, αν και κάπως περιορισμένο, παραμένει μεγάλο.

Η XML παρέχει κάποιες διευκολύνσεις για την ταυτοποίηση της κωδικοποίησης των χαρακτήρων Unicode που απαρτίζουν ένα κείμενο και για την απεικόνιση χαρακτήρων που, για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν ευθέως.

2.4 JavaScript

Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript.

2.4.1 Ιστορία

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε βήτα (beta) εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator εκδοχή 2.0 τον Σεπτέμβριο του 1995. LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape εκδοχή 2.0B3.

Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης

στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφάρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. JScript πρόσθεσε νέους μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίοι βασίστηκαν στην java.util.Date τάξη της Java. JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996.

Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript.

Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (και μεταξύ άλλων λόγων). Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες (π.χ. server-side).

2.4.2 Μοντέλο εκτέλεσης

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη στη γλώσσα προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες.

Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού).

Παρά την ευρεία χρήση της Javascript για συγγραφή προγραμμάτων σε περιβάλλον φυλλομετρητή, από την αρχή χρησιμοποιήθηκε και για τη συγγραφή κώδικα από την πλευρά του διακομιστή, από την ίδια τη Netscape στο προϊόν LiveWire, με μικρή επιτυχία. Η χρήση της Javascript στο διακομιστή εμφανίζεται πάλι σήμερα, με τη διάδοση του Node.js, ενός μοντέλου προγραμματισμού βασισμένο στα γεγονότα (events).

2.4.3 Javascript και Java

Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Η χρήση της λέξης «Java» στο όνομα της γλώσσας έχει περισσότερη σχέση με το προφίλ του προϊόντος που έπρεπε να έχει και λιγότερο με κάποια πιθανή συμβατότητα ή άλλη στενή σχέση με τη Java. Ρόλο σε αυτήν τη σύγχυση έπαιξε και ότι η Java και η Javascript έχουν δεχτεί σημαντικές επιρροές από τη γλώσσα C, ειδικά στο συντακτικό, ενώ είναι και οι δύο αντικειμενοστρεφείς γλώσσες. Τονίζεται ότι ο σωστός τρόπος γραφής της είναι «Javascript» και όχι 'Java script' σαν δύο λέξεις, όπως λανθασμένα γράφεται ορισμένες φορές.

2.4.4 Δείγμα κώδικα Javascript

Ο κώδικας Javascript μιας σελίδας περικλείεται από τις ετικέτες της HTML `<script type="text/javascript">` και `</script>`.

Για παράδειγμα, ο ακόλουθος κώδικας Javascript εμφανίζει ένα πλαίσιο διαλόγου με το κείμενο "Γεια σου, κόσμε!":

```
<script type="text/javascript">  
alert('Γεια σου, κόσμε!');  
</script>
```

Αν ο κώδικας Javascript περιέχει περισσότερες από μία εντολές, αυτές θα πρέπει να διαχωριστούν μεταξύ τους με το χαρακτήρα του ελληνικού ερωτηματικού ';' (δηλαδή της λατινικής άνω τελείας). Η χρήση του χαρακτήρα αυτού για την τελευταία εντολή δεν είναι απαραίτητη. Η διαχώριση των εντολών στους νεότερους φυλλομετρητές (browsers) δεν είναι απαραίτητη.

Μια άλλη βασική εντολή, η window.prompt("μήνυμα προς το χρήστη"), ζητάει από το χρήστη να συμπληρώσει ένα κομμάτι μιας αίτησης απευθείας ώστε τα δεδομένα να χρησιμοποιηθούν σαν κείμενο:

```
<script>  
var FIRSTvariable = window.prompt( "PLEASE FILL IN YOUR NAME" )  
alert( "Your name is " + FIRSTvariable + "." )  
</script>
```

2.5 PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό

περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

2.5.1 Επεκτάσεις αρχείων και διακομιστές

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

2.5.2 Ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον

Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

3. Υλοποίηση συστήματος

Συνδέουμε όλα τα παρελκόμενα μεταξύ τους με αγωγό ο οποίος έχει διατομή 0.50mm και αντέχει φορτίο μέχρι 6A. Η διατομή αυτή προέκυψε από το πηλίκο τάσης και ισχύς που χρησιμοποιούν οι συσκευές καθώς και από την πρόσθετη παροχή που θα δώσουμε στο σύστημα σε περίπτωση που μελλοντικά, προσθέσουμε ή αλλάξουμε συσκευή με διαφορετική κατανάλωση ενέργειας.

Κατανάλωση συσκευών:

Raspberry: 3W

Modem: 1W

GSM Alarm Dialer: 1W

Κονσόλα: 1W

Ρελέ: 0.24W

Ανεμιστήρας: 3.7W

Σύνολο: 9.94W

Υπολογίζουμε την ένταση:

$$I = \frac{P}{V} \Rightarrow$$

$$I = \frac{9.94}{12} \Rightarrow$$

$$I = 0.83A$$

Άρα περισσεύουν στο σύστημα 5A περίπου σε περίπτωση που προσθέσουμε και άλλα εξαρτήματα μελλοντικά. Απαιτούνται 2 ασφαλειοθήκες με ασφάλειες μέγιστης έντασης 6A οι οποίες είναι συνδεδεμένες σε σειρά ή μια με την μπαταρία και η άλλη με το φωτοβολταϊκό.

Η διάρκεια της μπαταρίας χωρίς ηλιοφάνεια με ενδεχόμενο να δουλεύουν όλες οι συσκευές με αδιάκοπη λειτουργία είναι:

$$\text{Τάση μπαταρίας} = 12\text{V}$$

$$\text{Χωρητικότητα μπαταρίας} = 21\text{Ah}$$

$$\text{Κατανάλωση ενέργειας} = 9.94\text{Wh}$$

Ο ανεμιστήρας είναι συνδεδεμένος στην παροχή του φωτοβολταϊκού άρα:

$$9.94 - 3.7 = 6.24\text{W} \text{ είναι η κατανάλωση των συσκευών από την μπαταρία.}$$

Άρα σε 24 ώρες θα έχουμε:

$$24 * 6.24 = 149.76\text{Wh}$$

Η μπαταρία έχει χωρητικότητα 21Ah και αυτό σημαίνει:

$$21 * 12 = 252\text{Wh}$$

Σύμφωνα με τα παραπάνω αποτελέσματα:

$$252 - 149.76 = 102.24\text{Wh}$$

Συμπέρασμα:

Η μπαταρία καλύπτει όλο το σύστημα όταν λειτουργούν όλες οι συσκευές χωρίς ηλιοφάνεια για 32 ώρες και 52 λεπτά. Η χωρητικότητα αυτή αρκεί επειδή υπάρχει κάθε μέρα ηλιοφάνεια έστω και λίγη (με σύννεφα φτάνει και τα $150\text{W}/\text{m}^2$) και οι συσκευές δεν θα λειτουργούν ποτέ όλες μαζί για τόσες ώρες. Εκτιμάται το σύστημα κατά μέσο όρο να λειτουργεί 2 μέρες.

Υπολογισμός διατομής αγωγού μεταξύ φωτοβολταϊκού στοιχείου και μπαταρίας κατά την διάρκεια φόρτισης.

Σε μεγάλη ηλιοφάνεια η απόδοση του φωτοβολταϊκού είναι 12V και 20W.

Άρα όταν φορτίζει η μπαταρία η ένταση εκείνη τη στιγμή είναι:

$$I = \frac{P}{V} \Rightarrow$$

$$I = \frac{20}{12} \Rightarrow$$

$$I = 1.66A$$

Οπότε χρησιμοποιούμε τον ίδιο αγωγό και η ένταση που περισσεύει σημαίνει ότι έχουμε περιθώρια για φωτοβολταϊκό στοιχείο μεγαλύτερων διαστάσεων με περισσότερη ισχύ.

Οι συσκευές όταν λειτουργούν όλες μαζί η ένταση στο κύκλωμα είναι 0.83A. Μια τυχαία μέτρηση που έγινε με το πολύμετρο όταν το σύστημα λειτουργούσε σε κανονικά επίπεδα έδειξε ένταση 0.47A. Άρα μια εκτίμηση που μπορούμε να κάνουμε είναι ότι από τα 1.66A έντασης που υπάρχει στον αγωγό του φωτοβολταϊκού σε μέρες με πλήρη ηλιοφάνεια, τα 0.47A τα χρησιμοποιούν οι συσκευές και ότι περισσεύει, δηλαδή 1.19A φορτίζει την μπαταρία. Άρα:

$$\frac{21}{1.19} = 17.6$$

Η μπαταρία εξαντλημένη χρειάζεται 17.6 ώρες ηλιοφάνειας για να φορτίσει πλήρως, υπό ιδανικές συνθήκες.

Σε όλες τις συνδέσεις είναι υποχρεωτική η χρήση φορητού πολύμετρου για έλεγχο, μέτρηση και ανίχνευση ηλεκτρικών προβλημάτων στο κύκλωμα.

Όσο αναφορά τον προγραμματισμό στο στάδιο της ανάπτυξης, ο προγραμματιστής πρέπει να προσέχει τα συντακτικά και περισσότερο τα λογικά λάθη ώστε να ολοκληρωθεί με επιτυχία η διαδικασία.

Η διαδικασία που θα περιγραφεί στη συνέχεια έχει σχεδιαστεί με προσανατολισμό την ελάχιστη κατανάλωση ενέργειας και του όγκου της πληροφορίας που ανταλλάσσεται μέσω του δικτύου κινητής τηλεφωνίας.

3.1 Προγραμματισμός GSM Alarm Dialer και SIM Card

Τα βήματα που πρέπει να κάνουμε για να δώσουμε τις παραμέτρους στην κάρτα SIM του δικτύου κινητής τηλεφωνίας Vodafone πριν την τοποθετήσουμε στο GSM Alarm Dialer.

Στην οθόνη αναμονής πληκτρολογούμε #43# και απενεργοποιούμε την αναμονή κλήσεων.

Καλούμε το 1285 και ακολουθούμε τις ηχογραφημένες οδηγίες. Απενεργοποιούμε τον αυτόματο τηλεφωνητή και την ειδοποίηση κλήσεων.

Απενεργοποιούμε τον κωδικό εισαγωγής PIN 1.

Η συγκεκριμένη διαδικασία ισχύει και για την κάρτα SIM του μόντεμ.

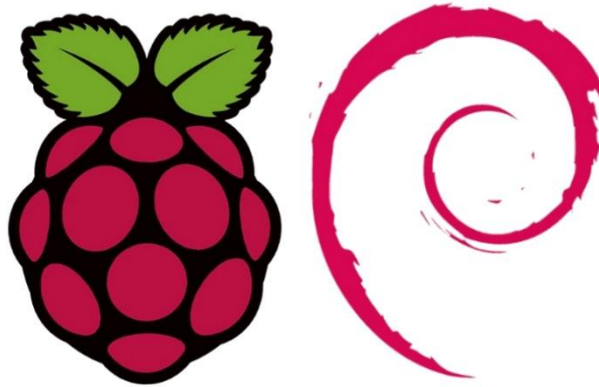
Η πλατφόρμα αυτή απαιτεί Master User, πιο συγκεκριμένα από ποιόν αριθμό κινητού δέχεται να εκτελεί τις παραμέτρους που της στέλνει με sms.

Στη δική μας περίπτωση το ρελέ αυτό οπλίζει και σβήνει το Raspberry με τηλεφωνική κλήση η μήνυμα και το Input A γίνεται HIGH όταν ανοίξει η πόρτα από το κουτί σε περίπτωση παραβίασης ειδοποιείται ο διαχειριστής.

3.2 Προετοιμασία και προγραμματισμός Raspberry

Το Raspberry δεν περιλαμβάνει τροφοδοσία, μνήμη ROM και λειτουργικό σύστημα. Η μνήμη που συνιστά ο κατασκευαστής είναι, ελάχιστη χωρητικότητα 4GB Class 10 η οποία φτάνει τα 10 MB/s. Τροφοδοτικό χρησιμοποιείται ήδη το DC to DC Converter το οποίο περιλαμβάνεται στα εξαρτήματα.

3.2.1 Εγκατάσταση Raspbian



Εικόνα 21 Raspbian Logo

Το Raspbian είναι ένα Debian-based λειτουργικό σύστημα του υπολογιστή για Raspberry Pi, που αναπτύχθηκε από μια μικρή ομάδα προγραμματιστών.

Κατεβάζουμε το πρόγραμμα από το https://downloads.raspberrypi.org/raspbian_latest

και το αρχείο το περνάμε στην MicroSD Card με πρόγραμμα όπως το Win32 Disk Imager.

Το πρόγραμμα θα φτιάξει 2 partition με αρχεία, το πρώτο FAT32 και το δεύτερο το γνωστό στα Linux Ext4. Όταν ολοκληρωθεί τοποθετούμε την κάρτα στο Raspberry. Κατά την εκκίνηση θα ξεκινήσει η εγκατάσταση του Raspbian με διάφορες παραμέτρους που πρέπει να δώσει ο χρήστης π.χ. όνομα χρήστη, περιβάλλον γραφικό ή μόνο κονσόλα, ενεργοποίηση/απενεργοποίηση κάμερας, ενεργοποίηση/απενεργοποίηση I2C κ.α.

3.2.2 Εγκατάσταση πακέτων για Raspbian

Για να εγκατασταθούν τα πακέτα τα οποία είναι απαραίτητα για να υλοποιήσουμε την εφαρμογή πληκτρολογούμε στον Terminal:

```
sudo apt-get update
```

```
sudo apt-get install gedit
```

```
sudo apt-get install ppp
```

3.2.2.1 Gedit

Το gedit είναι ένα πρόγραμμα επεξεργασίας κειμένου σχεδιασμένο για το γραφικό περιβάλλον εργασίας GNOME, του οποίου και αποτελεί τον επίσημο επεξεργαστή κειμένου. Είναι προεπιλεγμένο πρόγραμμα στην εγκατάσταση αρκετών διανομών Linux, όπως το Ubuntu και το Debian. Από το Δεκέμβριο του 2008 διατίθεται και για το λειτουργικό σύστημα Microsoft Windows. Το gedit είναι ελεύθερο λογισμικό, αδειοδοτημένο με την GNU General Public License (GNU GPL). Για να ανοίξουμε κάποιο αρχείο με τον Gedit αρκεί να πληκτρολογήσουμε:

```
sudo gedit /main.sh
```

3.2.2.2 PPP

Στα δίκτυα υπολογιστών, το PPP (Point-to-Point Protocol) είναι ένα πρωτόκολλο ζεύξης δεδομένων που χρησιμοποιείται για τη δημιουργία μια απευθείας σύνδεσης μεταξύ δύο κόμβων. Μπορεί να παρέχει έλεγχο ταυτότητας σύνδεσης, κρυπτογράφηση μετάδοσης και συμπίεση. Χρησιμοποιείται για dial-up πρόσβαση στο διαδίκτυο, δεδομένου ότι τα πακέτα IP δεν μπορούν να μεταδοθούν μέσω μιας γραμμής modem χωρίς κάποιο πρωτόκολλο ζεύξης δεδομένων.

Στη δική μας υλοποίηση με το πρωτόκολλο αυτό συνδέεται το modem στο Internet και ενημερώνει από μόνο του το Routing Table. Επίσης χρησιμοποιώντας την Chat που έχει ενσωματωμένη μπορούμε να επικοινωνούμε με το modem με AT Commands ώστε να μας δίνει τις πληροφορίες που χρειαζόμαστε. Για να δούμε το Routing Table πληκτρολογούμε:

```
sudo netstat -r -n
```

Κάποιες ρυθμίσεις όπως το APN του παρόχου είναι αποθηκευμένα στην κάρτα SIM.

3.2.2.3 Weewx

Το Weewx είναι ένα δωρεάν πρόγραμμα, ανοιχτού κώδικα λογισμικό, γραμμένο σε Python, το οποίο αλληλεπιδρά με το μετεωρολογικό σταθμό ώστε να παράγουν γραφικές

παραστάσεις, αρχεία αναφορών και αρχεία με καταλήξεις όπως HTML, XML κ.α. δίνοντας τα κατάλληλα weather tags.

Βήματα για την εγκατάσταση του Weewx:

```
sudo apt-get install python-imaging
```

```
sudo apt-get install python-usb
```

```
sudo apt-get install python-dev
```

<http://weewx.com/downloads/weewx-3.5.0.tar.gz>

```
tar xvfz weewx-3.4.0.tar.gz
```

```
cd weewx-3.4.0/
```

```
./setup.py build
```

```
sudo ./setup.py install
```

```
cd /home/weewx
```

```
sudo ./bin/weewxd weewx.conf
```

```
sudo tail -f /var/log/syslog
```

Βήματα για την απεγκατάσταση του Weewx:

```
sudo rm -r /home/weewx
```

```
sudo apt-get remove python-imaging
```

```
sudo apt-get remove python-usb
```

```
sudo apt-get remove python-dev
```

Ανοίγουμε το configuration file του Weewx:

```
sudo gedit /weewx/weewx.conf
```

για να ορίσουμε τους παρακάτω παραμέτρους:

Όνομα σταθμού

Υψόμετρο

Γεωγραφικό μήκος

Γεωγραφικό πλάτος

Τύπος σταθμού

USB θύρα

Μονάδες μέτρησης

Αρχείο/α εξαγωγής

Με το αρχείο που βρίσκεται στο:

```
/etc/init.d/weewx
```

το Weewx κάνει ενέργειες οι οποίες εξαρτώνται από τα arguments που δίνουμε.

```
sudo /etc/init.d/weewx start
```

 το πρόγραμμα ξεκινάει

```
sudo /etc/init.d/weewx stop
```

 το πρόγραμμα σταματάει

```
sudo /etc/init.d/weewx restart
```

 το πρόγραμμα κάνει επανεκκίνηση

Όταν το τρέξουμε για πρώτη φορά υπάρχει μεγάλη καθυστέρηση μέχρι να γίνει πλήρη συλλογή δεδομένων του ιστορικού της κονσόλας. Τα δεδομένα και τα γραφήματα δημιουργούνται μέσα στο φάκελο /weewx/public_html

3.2.3 Δημιουργία αρχείου shell script

Δημιουργούμε το βασικό αρχείο shell script π.χ. main.sh στο οποίο θα δηλώσουμε τύπο shell π.χ. sh ή bash. Για να το δηλώσουμε το ανοίγουμε με τον gedit και στη πρώτη γραμμή πληκτρολογούμε #!/bin/sh ή #!/bin/bash και πατάμε save. Είναι έτοιμο για να ξεκινήσουμε να γράφουμε τις εντολές. Για να καλέσει τις συναρτήσεις επειδή δεν δηλώνονται πρέπει να γίνει προσπέλαση ολόκληρου του shell. Μπορούμε να ξεκινήσουμε από το τέλος να διαβάζει τις συναρτήσεις. Για να έχουμε διαφορετικές εκτελέσεις σε κάθε ξεκίνημα φτιάχνουμε μια switch-case και την τρέχουμε με την τιμή που έχει το εξωτερικό argument. Όταν είμαστε έτοιμοι να το εκτελέσουμε είναι αδύνατο επειδή κάθε νέο αρχείο που δημιουργούν τα Linux δεν είναι executable. Για να το κάνουμε λοιπόν εκτελέσιμο ανοίγουμε τον Terminal και πληκτρολογούμε

```
sudo chmod +x main.sh
```

Το αρχείο είναι έτοιμο για εκτέλεση.

Στη παρούσα υλοποίηση το αρχείο αυτό έχει προγραμματιστεί ώστε να εκτελεί συγκεκριμένες ενέργειες για τη σωστή λειτουργία και την συντήρηση του συστήματος. Πιο συγκεκριμένα ελέγχει αν υπάρχει σύνδεση στο Internet κάθε μια ώρα, αν υπάρχει κάποιο πρόβλημα με το 3G modem, ισχύς σήματος modem, lac και cell modem, διαβάζει και εκτελεί εντολές από sms κάθε ένα λεπτό, αναγνώριση αριθμού αποστολέα του sms, δημιουργία αρχείου της τρέχουσας κατάστασης του συστήματος όταν το ζητήσει ο διαχειριστής, λήψης εικόνας από την κάμερα κάθε 5 λεπτά ή όταν το ζητήσει ο διαχειριστής, έλεγχος θερμοκρασίας του συστήματος και αυτόματη λειτουργία του ανεμιστήρα κάθε 1 λεπτό, περιοδική αποστολή δεδομένων στον κεντρικό server μέσω ftp κάθε 5 λεπτά, αλλαγή από 2G σε 3G και το αντίστροφο όταν το ζητήσει ο διαχειριστής.

Το script έχει προγραμματιστεί έτσι ώστε να δέχεται μέσω sms εντολές όπως:

Shutdown, reboot, status, fan auto, fan off, fan on, modem connect, modem disconnect, modem reset, modem auto, modem gsm, modem gsm/wcdma, modem wcdma, modem wcdma/gsm, camera auto, camera off, camera on, weewx stop, weewx start, weewx restart, charts, default command.

Η κάμερα έχει λήψη εικόνας με διαστάσεις 640 x 480 pixels και τύπο αρχείου .jpg. Σε αυτές τις διαστάσεις έχουμε καλή ποιότητα εικόνας και χαμηλό μέγεθος αρχείου.

3.2.4 Εκτέλεση αρχείων κατά την εκκίνηση

Στο αρχείο `rc.local` έχουμε τη δυνατότητα να βάζουμε εντολές και να εκτελούνται κατά την εκκίνηση του Raspbian.

```
sudo gedit /etc/rc.local
```

Στο τέλος κάθε εντολής βάζουμε το `&` για να εκτελεστεί σε `background`. Αν δεν την εκτελέσουμε σε `background` και η εντολή δεν εκτελεστεί σωστά το shell θα μείνει κολλημένο στο συγκεκριμένο `process id` και το Raspbian δεν θα ξεκινήσει.

3.2.5 Περιοδική εκτέλεση αρχείων με τον χρονοπρογραμματιστή Cron

Ο χρονοπρογραμματιστής Cron είναι το κατάλληλο πρόγραμμα για προγραμματισμό επαναλαμβανόμενων εργασιών.

Δίνοντας την εντολή `crontab -e` ανοίγει ένα παράθυρο που είναι το configuration file του Cron. Αυτό το αρχείο ρυθμίσεων καθορίζει τις εντολές shell να τρέξουν σε τακτά χρονικά διαστήματα σύμφωνα με το χρονοδιάγραμμα. Σε κάθε γραμμή ορίζουμε πρώτα τα πεδία με τις προδιαγραφές όπως λεπτά, ώρα, έτος, ημέρα εβδομάδας και μετά την εργασία που επιθυμούμε να εκτελεστεί. Όταν τα πεδία αυτά ταιριάζουν με την τρέχουσα ημερομηνία και ώρα τότε εκτελείται η εργασία.

```
# _____ min ( 0 – 59 )
# | _____ hour ( 0 – 23 )
# | | _____ day of month ( 1 – 31 )
# | | | _____ month ( 1 – 12 )
# | | | | _____ day of week ( 0 – 6 ) ( 0 to 6 are Sunday to Saturday )
# | | | | |
# * * * * * command to execute
```

Παραδείγματα:

Εκτέλεσε την παρακάτω εργασία κάθε 1 ώρα.

```
0 * * * * sudo /usr/local/sbin/main.sh
```

ή

```
@hourly sudo /usr/local/sbin/main.sh
```

Εκτέλεσε την παρακάτω εργασία κάθε 2 ώρες στα πρώτα 15 λεπτά.

```
15 */2 * * * sudo /usr/local/sbin/main.sh
```

Εκτέλεσε την παρακάτω εργασία στις 31 Αυγούστου στις 20:00 η ώρα.

```
0 20 31 8 * sudo /usr/local/sbin/main.sh
```

Εκτέλεσε την παρακάτω εργασία από τις 07:00 μέχρι τις 15:00 κάθε 20 λεπτά.

```
*/20 7-15 * * * sudo /usr/local/sbin/main.sh
```

Εκτέλεσε την παρακάτω εργασία στις 06:00 και στις 22:00.

```
0 6,22 * * * sudo /usr/local/sbin/main.sh
```

3.2.6 Raspberry Time Keeping Failure

Το Raspberry μετά από σβήσιμο ή επανεκκίνηση χάνει την ώρα, δεν έχει Real Time Clock. Δεν υπάρχει μπαταρία πάνω στην πλακέτα. Αποτέλεσμα κάποια προγράμματα να μη λειτουργούν σωστά. Για να λυθεί το πρόβλημα αυτό δημιουργούμε 1 αρχείο shell script με όνομα time.sh με τις παρακάτω εντολές:


```
#!/bin/bash

time=`date "+%s"`

while [ true ]

do

    [ "${time}" -lt "946684800" ] && break

    sleep 1

    time=`date "+%s"`

done

echo 'New Time Arrival!'
```

Τι πετύχαμε με αυτό.

Όταν ξεκινάει το Raspberry η ώρα βρίσκεται στο σημείο 01/01/1980 00:00 αλλά επειδή κατά την διάρκεια εγκατάστασης του Raspbian θέσαμε μια παράμετρο την TimeZone σε GMT+2 η ώρα που βλέπουμε είναι 01/01/1980 02:00.

Τη στιγμή που συνδέεται στο Internet ένας δαίμονας τρέχει στο Raspberry ονόματι NTP (Network Time Protocol). Αυτός ο δαίμονας στην εκκίνηση και ανά διαστήματα συγχρονίζει την ώρα από το Internet χρησιμοποιώντας την τοποθεσία που έχουμε ρυθμίσει στο Raspbian. Το πρόβλημα δημιουργείται όταν στην εκκίνηση ξεκινήσουν κάποια προγράμματα με την παλιά ώρα και μετά από λίγο μάθει ο NTP την καινούργια ώρα και μόλις γίνει ενημέρωση ταυτόχρονα παύουν να δουλεύουν.

Ένα από αυτά είναι το Weewx. Οπότε πριν την εντολή:

```
sudo /etc/init.d/weewx start
```

που είναι η εντολή για να ξεκινάει το Weewx μπορούμε να βάλουμε μέσα στο shell script να καλεί το παραπάνω shell script. Αυτό γίνεται ως εξής:

```
./time.sh
```

```
sudo /etc/init.d/weewx start
```

Η τελεία καλεί εξωτερικό shell script. Στις τελευταίες σελίδες υπάρχουν τα commands help του Linux. Στο date --help παρατηρούμε αυτό:

```
%s seconds since 1970-01-01 00:00:00 UTC
```

Η while κάνει επαναλήψεις όσο τα δευτερόλεπτα από την 01/01/1970 00:00 είναι λιγότερα από τα δευτερόλεπτα που έχουν περάσει το έτος 01/01/2000 00:00 από την 01/01/1970 00:00 δηλαδή 946684800. Ο συγκεκριμένος χρόνος ονομάζεται Epoch Time.

Όταν ο NTP συγχρονιστεί με την καινούργια ώρα το loop της while σπάει επειδή τα δευτερόλεπτα της τρέχουσας ώρας που έχουν περάσει από την 01/01/1970 00:00 είναι περισσότερα από 946684800. Αποτέλεσμα τα προγράμματα να ξεκινούν όταν συγχρονιστεί η σωστή ώρα!

3.2.7 Udev Rules

Κατά την εκκίνηση οι συσκευές φορτώνονται ασύγχρονα επειδή έχουν διαφορετικό setup και firmware η κάθε μία. Κάθε φορά που ξεκινάει το λειτουργικό οι USB συσκευές τυχαίνει να έχουν διαφορετικό όνομα και αυτό εξαρτάται από τον χρόνο που χρειάζεται η κάθε συσκευή να φορτωθεί. Αυτό έχει ως αποτέλεσμα κάποια προγράμματα όπως το Weewx να είναι αδύνατο να εντοπίσουν την συσκευή που ορίσαμε στο Configuration file σε κάποια συγκεκριμένη θύρα.

Το Udev είναι ένας διαχειριστής συσκευής για τον πυρήνα του Linux και επιτρέπει κανόνες που καθορίζουν τι όνομα δίνεται σε μια συσκευή, ανεξάρτητα σε ποια θύρα είναι συνδεδεμένη. Εντοπίζει συσκευές που βασίζονται στις ιδιότητές τους, όπως το Vendor ID (αναγνωριστικό προμηθευτή) και το Device ID (αναγνωριστικό συσκευής). Διαχειρίζεται Udev κυρίως αρχεία συσκευών στον κατάλογο /dev.

Για να δούμε το Vendor ID και Device ID μιας συσκευής στον Terminal δίνουμε την εντολή lsusb. Παρατηρούμε έναν αριθμό χωρισμένο με άνω-κάτω τελεία :.

Το αρχείο με τους κανόνες το φτιάχνουμε στη θέση:

```
/etc/udev/rules.d/
```

Παραδείγματα πως φτιάχνουμε το αρχείο Udev με τους δικούς μας κανόνες:

```
ACTION=="add",      ATTRS{idVendor}=="10c4",      ATTRS{idProduct}=="ea60",  
SYMLINK+="station"
```

2 κανόνες επειδή κάποιες συσκευές όπως το modem έχει περισσότερα από 1 USB Interface:

```
ACTION=="add",      ATTRS{idVendor}=="12d1",      ATTRS{idProduct}=="1001",  
ENV{ID_USB_INTERFACE_NUM}=="00", SYMLINK+="3gmodem"
```

```
ACTION=="add",      ATTRS{idVendor}=="12d1",      ATTRS{idProduct}=="1001",  
ENV{ID_USB_INTERFACE_NUM}=="02", SYMLINK+="smsmodem"
```

```
$ lsusb  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.  
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.  
Bus 001 Device 004: ID 046d:08ad Logitech, Inc. QuickCam Communi  
cate STX
```

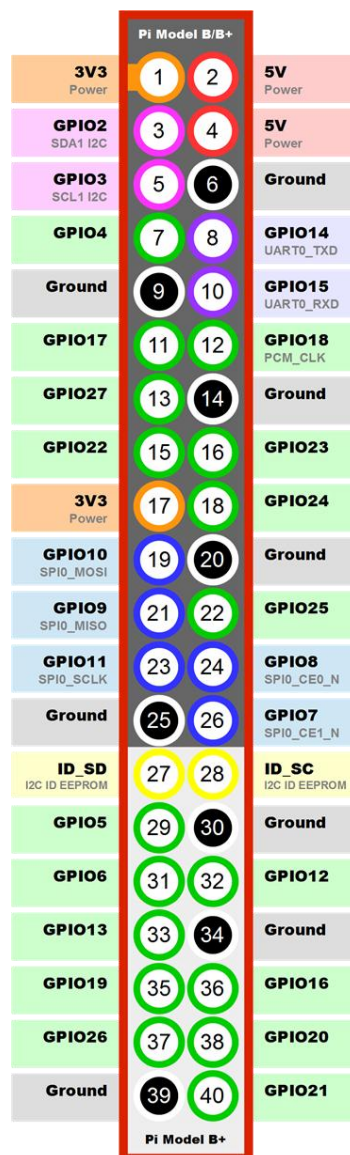
Εικόνα 22 lsusb

Για περισσότερες λεπτομέρειες μιας συσκευής στον Terminal δίνουμε την εντολή:

```
lsusb -vv
```

3.2.8 Ακροδέκτης GPIO

Ο GPIO (General Purpose Input/Output, γενικού σκοπού είσοδος/έξοδος) είναι ένας ακροδέκτης I/O ενός ολοκληρωμένου κυκλώματος που η συμπεριφορά του ακόμα και το αν θα λειτουργεί σαν είσοδος ή έξοδος ρυθμίζεται μέσω λογισμικού. Από προεπιλογή είναι απενεργοποιημένο και δεν έχει κάποια προκαθορισμένη χρήση.



Εικόνα 23 General purpose input/output (GPIO)

Το Raspberry αποτελείται από 40 ακροδέκτες I/O. Κατά την εκκίνηση του συστήματος θέτει αυτούς τους ακροδέκτες σε λογικό επίπεδο 0 εκτός αυτούς τους οποίους είναι Power και

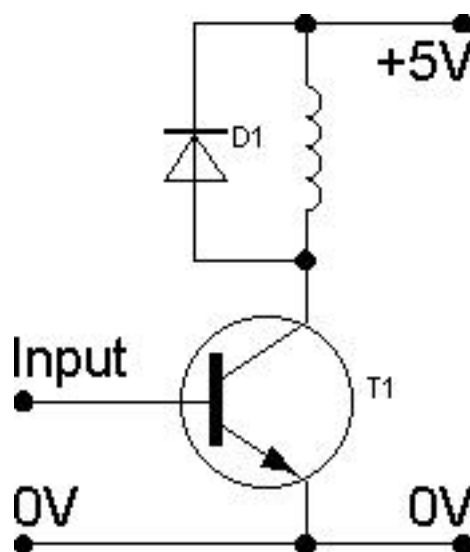
Ground. Αυτό σημαίνει ότι αν μετρηθούν θα διαπιστώσουμε ότι δεν υπάρχει τάση. Όταν ο ακροδέκτης GPIO25 βρίσκεται σε λογικό επίπεδο 1 τότε η τάση μεταξύ ακροδέκτη GPIO25 και ακροδέκτη Ground είναι 3.3V. Κάνουμε την μέτρηση χρησιμοποιώντας το πολύμετρο.

Για να θέσουμε κάποιον ακροδέκτη I/O σε λογικό επίπεδο 1 πρέπει από το λογισμικό να γίνει πρώτα export και να ορίσουμε το direction αν θα είναι output (τάση 0V ή 3.3V) ή input (διάβασε αν η τάση είναι 0V ή 3.3V).

3.2.9 Σύνδεση ρελέ με τον ακροδέκτη του Raspberry

Στην παρούσα υλοποίηση χρησιμοποιήθηκε ο ακροδέκτης GPIO25 ο οποίος συνδέθηκε στο In του ρελέ. Το ρελέ επίσης παίρνει τροφοδοσία από τους ακροδέκτες Power και Ground του Raspberry η οποία είναι 5V.

Το ρελέ για να οπλίσει το κανονίζει το τρανζίστορ που βρίσκεται πάνω στην ίδια πλακέτα. Το τρανζίστορ είναι διάταξη ημιαγωγών στερεάς κατάστασης, η οποία βρίσκει διάφορες εφαρμογές στην ηλεκτρονική, μερικές εκ των οποίων είναι η ενίσχυση, η σταθεροποίηση τάσης, η διαμόρφωση συχνότητας, η λειτουργία ως διακόπτης και ως μεταβλητή ωμική αντίσταση. Το τρανζίστορ μπορεί, ανάλογα με την τάση με την οποία πολώνεται, να ρυθμίζει την ροή του ηλεκτρικού ρεύματος που απορροφά από συνδεδεμένη πηγή τάσης. Τα τρανζίστορ κατασκευάζονται είτε ως ξεχωριστά ηλεκτρονικά εξαρτήματα είτε ως τμήματα κάποιου ολοκληρωμένου κυκλώματος.



Εικόνα 24 Transistor

Στη δική μας περίπτωση λειτουργεί ως διακόπτης όπως φαίνεται και στο παραπάνω σχήμα. Στο shell script κατασκευάστηκε συνάρτηση η οποία διαβάζει το αρχείο που έχει δημιουργήσει το Weewx το οποίο περιλαμβάνει την εσωτερική θερμοκρασία και υγρασία που διαβάζει η κονσόλα. Επίσης η συνάρτηση διαβάζει και την τρέχουσα θερμοκρασία του επεξεργαστή. Αυτό γίνεται με την παρακάτω εντολή:

```
local cpu=`/opt/vc/bin/vcgencmd measure_temp | sed -e "s/^temp=*(.*)'C.*$/1/g"`
```

Το Standard Output μετά το pipe το παίρνει η sed και με regular expressions διώχνει το temp=°C και κρατάει μόνο την τιμή. Ανάλογα με τις απαιτήσεις που έχει ο καθένας ορίζει βάση αυτών των παραμέτρων πότε βρίσκεται σε λογικό επίπεδο ο ακροδέκτης GPIO25. Όταν ο ακροδέκτης GPIO25 βρίσκεται σε λογικό επίπεδο 1 τότε η τάση είναι 3.3V και φτάνει στον ακροδέκτη του τρανζίστορ με αποτέλεσμα να κλείσει το κύκλωμα που συνδέεται το ρελέ. Στις επαφές του ρελέ είναι συνδεδεμένο το καλώδιο του ανεμιστήρα. Αφού σπλίσει και η πόρτα από το κουτί είναι κλειστή, ο ανεμιστήρας ξεκινά να λειτουργεί.

3.3 Προγραμματισμός Web Server

Βασικός σκοπός είναι η δημιουργία ιστοσελίδας η οποία θα περιέχει πίνακα με τα δεδομένα του αρχείου XML που δημιουργεί το Weewx καθώς και την εικόνα από την κάμερα. Τα δεδομένα θα απεικονίζονται στους επισκέπτες. Επίσης δημιουργία δεύτερης ιστοσελίδας η οποία θα περιέχει πίνακα με τα δεδομένα του αρχείου XML που αφορά την κατάσταση του συστήματος. Αυτό το αρχείο δημιουργείται από το shell script. Τα δεδομένα θα απεικονίζονται μόνο στον διαχειριστή που θα ελέγχει το σύστημα.

3.3.1 Client Side

3.3.1.2 Parsing the XML file

Υλοποιούμε το αρχείο HTML το οποίο απεικονίζει τα δεδομένα. Η πηγή των δεδομένων είναι το αρχείο XML. Για να γίνει το parsing ώστε τα δεδομένα να φαίνονται στην HTML και να ανανεώνονται χωρίς reload θα χρησιμοποιήσουμε JavaScript και μέσα σε αυτή Ajax.

Το Ajax (Asynchronous JavaScript and XML) είναι ένα σύνολο τεχνικών ανάπτυξης ιστού χρησιμοποιώντας πολλές τεχνολογίες Web στην πλευρά του πελάτη για τη δημιουργία

ασύγχρονης εφαρμογής Web. Με Ajax, οι διαδικτυακές εφαρμογές μπορούν να στείλουν δεδομένα και να γίνει ανάκτηση από έναν διακομιστή ασύγχρονα στο παρασκήνιο. Η JavaScript και το XMLHttpRequest αντικείμενο παρέχει μια μέθοδο για την ανταλλαγή δεδομένων ασύγχρονα μεταξύ Browser και Server για να αποφύγουμε την πλήρη επαναφόρτωση της ιστοσελίδας.

Η JavaScript μετατρέπει το XML σε αντικείμενο XML DOM (Document Object Model) με όνομα xmlhttp.responseXML και όλο το περιεχόμενο το εκχωρεί σε νέα μεταβλητή. Η μεταβλητή αυτή μετά μεταφέρει όλο το XML όσο εκτελείται ο κώδικας και κάθε φορά που χρειαστεί μια τιμή από element ή attribute χρησιμοποιούμε τις εξής εντολές:

```
element=xmlDoc.getElementsByTagName("elementname")[0].childNodes[0].nodeValue
```

```
attribute=xmlDoc.getElementsByTagName("element")[0].attributes.getNamedItem("attribute name").value
```

3.3.2 Server Side

3.3.2.1 Αρχείο PHP

Στη PHP υπάρχει μεγάλη βιβλιοθήκη από συναρτήσεις οι οποίες παρέχουν πολλές λειτουργίες. Μια από αυτές δίνοντας το path ενός αρχείου που βρίσκεται στον server μας επιστρέφει την ώρα που δημιουργήθηκε ή τροποποιήθηκε το αρχείο αυτό. Σε περίπτωση που θέλουμε να ενημερώσουμε την JavaScript ότι το Raspberry έχει πολύ ώρα να στείλει νέα δεδομένα ώστε να ενημερώσει τον επισκέπτη για αυτό, δημιουργήθηκε αρχείο που διαβάζει την τρέχουσα ώρα, την ώρα που δημιουργήθηκε το τελευταίο αρχείο και συγκρίνοντας αυτά τα 2 επιστρέφουμε true όταν το αρχείο φτάνει στην ώρα του ή false όταν το αρχείο έχει πολύ ώρα να δημιουργηθεί με αποτέλεσμα να μην υπάρχει επικοινωνία μεταξύ Raspberry και Server.

3.3.2.1 Δημιουργία Under Construction ιστοσελίδας

Όταν δημιουργήσουμε Under Construction ιστοσελίδα κάνουμε redirect προς αυτή με .htaccess αρχείο βάζοντας κάποια rules στο αρχείο αυτό. Με αυτό τον τρόπο δεν χρειάζεται το αρχείο html που περιέχει την Under Construction να το κάνουμε rename σε index.html

Παραδείγματα:

301 (Permanent)

Redirect 301 / http://www.example.com/

302 (Temporary)

Redirect 302 / http://www.example.com/

Redirect index.html to a specific subfolder:

Redirect /index.html http://example.com/newdirectory/

(Set the default handler)

DirectoryIndex index.html

Προβλήματα που αντιμετωπίστηκαν

Κατά τη διάρκεια του boot το Run Level rc.local είχε εντολή μέσα η οποία δεν εκτελέστηκε σε background και δεν ολοκληρώθηκε με αποτέλεσμα να έχουμε αποτυχία εκκίνησης.

Λύση:

Βάζουμε το & στο τέλος κάθε εντολής.

Η κάρτα microSD της μειώνεται η διάρκεια ζωής όσοις πιο πολλές εγγραφές γίνονται.

Λύση:

Αποθηκεύουμε τα προσωρινά δεδομένα στο /tmp έτσι ώστε να αποθηκεύονται στη μνήμη RAM του Raspberry και όχι στην κάρτα microSD. Προσοχή, μετά από restart τα δεδομένα χάνονται.

Δεν γινόταν ποτέ η ίδια αντιστοίχιση στις συσκευές μετά από restart.

Λύση:

Δικά μας ονόματα συσκευών USB με Udev Rules χρησιμοποιώντας από την συσκευή το Vendor ID και το Device ID.

Μερικές φορές η μπαταρία εξαντλείται πιο γρήγορα.

Λύση:

Η συσκευές που δεν έχουν σταθερή κατανάλωση ενέργειας είναι το μόντεμ και ο ανεμιστήρας. Για το μόντεμ ρυθμίζουμε το Network mode σε:

GSM only ή 2G έχει χαμηλή ταχύτητα και μικρή κατανάλωση ενέργειας.

WCDMA only ή 3G έχει υψηλή ταχύτητα και μεγάλη κατανάλωση ενέργειας.

GSM / WCDMA auto επιλέγει 2G ή 3G ανάλογα με την ισχύ του σήματος.

Για τον ανεμιστήρα μπορούμε κάποιες μέρες που δεν έχει πολύ ηλιοφάνεια να τον απενεργοποιήσουμε.

Πιθανά προβλήματα

Τα νέα δεδομένα έφτασαν στον server αλλά η ιστοσελίδα εξακολουθεί να δείχνει τα παλιά.

Λύση:

Τα δεδομένα διαβάζονται ακόμα από την μνήμη cache. Με Ctrl+F5 σβήνουμε την μνήμη.

Τα δεδομένα σταμάτησαν να ανανεώνονται.

Λύση:

Ενέργειες που πρέπει να κάνουμε:

Έλεγχος αν εξαντλήθηκε η μπαταρία.

Από το κινητό μας τηλέφωνο καλούμε στο 69*****.

Αν η κλήση δεν είναι εφικτή τότε η μπαταρία πιθανόν να εξαντλήθηκε.

Αν η κλήση είναι εφικτή τότε πιθανό να εξαντλήθηκαν τα MB από το Internet. Το ελέγχουμε μπαίνοντας στο my account της Vodafone. Αν υπάρχουν διαθέσιμα MB τότε περιμένουμε να κάνει Restart το πρόγραμμα την ώρα που ορίσαμε το Crontab και σε περίπτωση που το σύστημα έχει κολλήσει τότε κάνουμε αναπάντητη στο GSM Alarm Dialer για να κόψει το ρεύμα και να ξεκινήσουν όλα μαζί και η τελευταία εναλλακτική γινόμαστε ψύχραιμοι παίρνουμε το αυτοκίνητο και πάμε ένα μικρό ταξιδάκι να το επισκεφτούμε!

Ας πάρουμε και το πολόμετρο μαζί μας όλο και κάπου θα χρειαστεί...

Αναφορές / Links

- [1] <http://www.davisnet.com>
- [2] <http://www.schneider-electric.com>
- [3] <https://www.raspberrypi.org>
- [4] <http://www.kingston.com>
- [5] <http://www.huawei.com>
- [6] <https://el.wikipedia.org/wiki/HTML>
- [7] <http://dide.flo.sch.gr/Plinet/Tutorials-CSS-Bible/Tutorials-CSS-Bible-1.html>
- [8] <https://el.wikipedia.org/wiki/XML>
- [9] <https://el.wikipedia.org/wiki/JavaScript>
- [10] <https://el.wikipedia.org/wiki/PHP>
- [11] <https://help.ubuntu.com/community/CronHowto>
- [12] www.w3schools.com

AT Commands

To AT προέρχεται από το Attention και

To ID προέρχεται από το Interface Description.

Ας δούμε μερικές εντολές:

ATI Request Identification

ATZ Resets the modem and recalls a user profile

AT+GMI ID information about manufacturer

AT+GMM ID information about product model

AT+GMR ID information about software version

AT+GSN ID information about ESN/IMEI

AT+GCAP ID information about capability list

AT&F Returns the configuration to the factory settings

AT&V Query Current Configuration

AT^U2DIAG Disable or enable the following USB-components

AT+COPS Select operator

AT+CLCK Lock facility

AT+CFUN Set operation mode

AT+CPIN Enter PIN

AT+CGDCONT Define PDP Context

AT+CGACT Activate or Deactivate PDP context

AT+CGATT Attach or Detach PS domain

AT+CGREG PS domain registration status

AT+CGSMS SMS bearer domain

AT+CSMS Select messaging service

AT+CPMS Select message storage

AT+CMGF Set message format

AT+CSCA Service Center Address

AT+CMGS Send Short Message

AT+CMGD Delete message

AT+CMGL List messages

AT+CMGR Read message

AT^SYSINFOEX Extended System Information Query Command

AT+CSQ Report signal strength

AT^CPIN Manage PIN

Linux commands help

chat

Usage: chat [-e] [-E] [-v] [-V] [-t timeout] [-r report-file]

[-T phone-number] [-U phone-number2] {-f chat-file | chat-script}

cp

Usage: cp [OPTION]... [-T] SOURCE DEST

or: cp [OPTION]... SOURCE... DIRECTORY

or: cp [OPTION]... -t DIRECTORY SOURCE...

Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.

- a, --archive same as -dR --preserve=all
- attributes-only don't copy the file data, just the attributes
- backup[=CONTROL] make a backup of each existing destination file
- b like --backup but does not accept an argument
- copy-contents copy contents of special files when recursive
- d same as --no-dereference --preserve=links
- f, --force if an existing destination file cannot be
 opened, remove it and try again (this option
 is ignored when the -n option is also used)
- i, --interactive prompt before overwrite (overrides a previous -n
 option)
- H follow command-line symbolic links in SOURCE
- l, --link hard link files instead of copying
- L, --dereference always follow symbolic links in SOURCE
- n, --no-clobber do not overwrite an existing file (overrides
 a previous -i option)
- P, --no-dereference never follow symbolic links in SOURCE
- p same as --preserve=mode,ownership,timestamps
- preserve[=ATTR_LIST] preserve the specified attributes (default:
 mode,ownership,timestamps), if possible
 additional attributes: context, links, xattr,
 all

--no-preserve=ATTR_LIST don't preserve the specified attributes
 --parents use full source file name under DIRECTORY
 -R, -r, --recursive copy directories recursively
 --reflink[=WHEN] control clone/CoW copies. See below
 --remove-destination remove each existing destination file before
 attempting to open it (contrast with --force)
 --sparse=WHEN control creation of sparse files. See below
 --strip-trailing-slashes remove any trailing slashes from each SOURCE
 argument
 -s, --symbolic-link make symbolic links instead of copying
 -S, --suffix=SUFFIX override the usual backup suffix
 -t, --target-directory=DIRECTORY copy all SOURCE arguments into DIRECTORY
 -T, --no-target-directory treat DEST as a normal file
 -u, --update copy only when the SOURCE file is newer
 than the destination file or when the
 destination file is missing
 -v, --verbose explain what is being done
 -x, --one-file-system stay on this file system
 --help display this help and exit
 --version output version information and exit

By default, sparse SOURCE files are detected by a crude heuristic and the
 corresponding DEST file is made sparse as well. That is the behavior

selected by `--sparse=auto`. Specify `--sparse=always` to create a sparse DEST file whenever the SOURCE file contains a long enough sequence of zero bytes. Use `--sparse=never` to inhibit creation of sparse files.

When `--reflink[=always]` is specified, perform a lightweight copy, where the data blocks are copied only when modified. If this is not possible the copy fails, or if `--reflink=auto` is specified, fall back to a standard copy.

The backup suffix is '~', unless set with `--suffix` or `SIMPLE_BACKUP_SUFFIX`.

The version control method may be selected via the `--backup` option or through the `VERSION_CONTROL` environment variable. Here are the values:

`none, off` never make backups (even if `--backup` is given)
`numbered, t` make numbered backups
`existing, nil` numbered if numbered backups exist, simple otherwise
`simple, never` always make simple backups

As a special case, `cp` makes a backup of SOURCE when the `force` and `backup` options are given and SOURCE and DEST are the same name for an existing, regular file.

cut

Usage: `cut OPTION... [FILE]...`

Print selected parts of lines from each FILE to standard output.

Mandatory arguments to long options are mandatory for short options too.

-b, --bytes=LIST select only these bytes

-c, --characters=LIST select only these characters

-d, --delimiter=DELIM use DELIM instead of TAB for field delimiter

-f, --fields=LIST select only these fields; also print any line

that contains no delimiter character, unless

the -s option is specified

-n (ignored)

--complement complement the set of selected bytes, characters

or fields

-s, --only-delimited do not print lines not containing delimiters

--output-delimiter=STRING use STRING as the output delimiter

the default is to use the input delimiter

--help display this help and exit

--version output version information and exit

Use one, and only one of -b, -c or -f. Each LIST is made up of one

range, or many ranges separated by commas. Selected input is written

in the same order that it is read, and is written exactly once.

Each range is one of:

- N N'th byte, character or field, counted from 1
- N- from N'th byte, character or field, to end of line
- N-M from N'th to M'th (included) byte, character or field
- M from first to M'th (included) byte, character or field

With no FILE, or when FILE is -, read standard input.

date

Usage: date [OPTION]... [+FORMAT]

or: date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

Display the current time in the given FORMAT, or set the system date.

Mandatory arguments to long options are mandatory for short options too.

-d, --date=STRING display time described by STRING, not 'now'

-f, --file=DATEFILE like --date once for each line of DATEFILE

-I[TIMESPEC], --iso-8601[=TIMESPEC] output date/time in ISO 8601 format.

TIMESPEC='date' for date only (the default),

'hours', 'minutes', 'seconds', or 'ns' for date

and time to the indicated precision.

-r, --reference=FILE display the last modification time of FILE

-R, --rfc-2822 output date and time in RFC 2822 format.

Example: Mon, 07 Aug 2006 12:34:56 -0600

--rfc-3339=TIMESPEC output date and time in RFC 3339 format.

TIMESPEC='date', 'seconds', or 'ns' for

date and time to the indicated precision.

Date and time components are separated by

a single space: 2006-08-07 12:34:56-06:00

-s, --set=STRING set time described by STRING

-u, --utc, --universal print or set Coordinated Universal Time

--help display this help and exit

--version output version information and exit

FORMAT controls the output. Interpreted sequences are:

%% a literal %

%a locale's abbreviated weekday name (e.g., Sun)

%A locale's full weekday name (e.g., Sunday)

%b locale's abbreviated month name (e.g., Jan)

%B locale's full month name (e.g., January)

%c locale's date and time (e.g., Thu Mar 3 23:05:25 2005)

%C century; like %Y, except omit last two digits (e.g., 20)

%d day of month (e.g., 01)

%D date; same as %m/%d/%y

%e day of month, space padded; same as %_d

%F full date; same as %Y-%m-%d

%g last two digits of year of ISO week number (see %G)

%G year of ISO week number (see %V); normally useful only with %V

%h same as %b

%H hour (00..23)

%I hour (01..12)

%j day of year (001..366)

%k hour, space padded (0..23); same as %_H

%l hour, space padded (1..12); same as %_I

%m month (01..12)

%M minute (00..59)

%n a newline

%N nanoseconds (000000000..999999999)

%p locale's equivalent of either AM or PM; blank if not known

%P like %p, but lower case

%r locale's 12-hour clock time (e.g., 11:11:04 PM)

%R 24-hour hour and minute; same as %H:%M

%s seconds since 1970-01-01 00:00:00 UTC

%S second (00..60)

%t a tab

%T time; same as %H:%M:%S

%u day of week (1..7); 1 is Monday

%U week number of year, with Sunday as first day of week (00..53)

%V ISO week number, with Monday as first day of week (01..53)

%w day of week (0..6); 0 is Sunday

%W week number of year, with Monday as first day of week (00..53)

%x locale's date representation (e.g., 12/31/99)

%X locale's time representation (e.g., 23:13:48)

%y last two digits of year (00..99)

%Y year

%Z +hhmm numeric time zone (e.g., -0400)

:%z +hh:mm numeric time zone (e.g., -04:00)

%::z +hh:mm:ss numeric time zone (e.g., -04:00:00)

%:::z numeric time zone with : to necessary precision (e.g., -04, +05:30)

%Z alphabetic time zone abbreviation (e.g., EDT)

By default, date pads numeric fields with zeroes.

The following optional flags may follow '%':

- (hyphen) do not pad the field

_ (underscore) pad with spaces

0 (zero) pad with zeros

^ use upper case if possible

use opposite case if possible

After any flags comes an optional field width, as a decimal number;

then an optional modifier, which is either

E to use the locale's alternate representations if available, or

O to use the locale's alternate numeric symbols if available.

Examples:

Convert seconds since the epoch (1970-01-01 UTC) to a date

```
$ date --date='@2147483647'
```

Show the time on the west coast of the US (use tzselect(1) to find TZ)

```
$ TZ='America/Los_Angeles' date
```

Show the local time for 9AM next Friday on the west coast of the US

```
$ date --date='TZ="America/Los_Angeles" 09:00 next Fri'
```

expr

Usage: expr EXPRESSION

or: expr OPTION

--help display this help and exit

--version output version information and exit

Print the value of EXPRESSION to standard output. A blank line below

separates increasing precedence groups. EXPRESSION may be:

ARG1 | ARG2 ARG1 if it is neither null nor 0, otherwise ARG2

ARG1 & ARG2 ARG1 if neither argument is null or 0, otherwise 0

ARG1 < ARG2 ARG1 is less than ARG2

ARG1 <= ARG2 ARG1 is less than or equal to ARG2

$ARG1 = ARG2$ $ARG1$ is equal to $ARG2$
 $ARG1 != ARG2$ $ARG1$ is unequal to $ARG2$
 $ARG1 >= ARG2$ $ARG1$ is greater than or equal to $ARG2$
 $ARG1 > ARG2$ $ARG1$ is greater than $ARG2$
 $ARG1 + ARG2$ arithmetic sum of $ARG1$ and $ARG2$
 $ARG1 - ARG2$ arithmetic difference of $ARG1$ and $ARG2$
 $ARG1 * ARG2$ arithmetic product of $ARG1$ and $ARG2$
 $ARG1 / ARG2$ arithmetic quotient of $ARG1$ divided by $ARG2$
 $ARG1 \% ARG2$ arithmetic remainder of $ARG1$ divided by $ARG2$
 $STRING : REGEXP$ anchored pattern match of $REGEXP$ in $STRING$
 $match\ STRING\ REGEXP$ same as $STRING : REGEXP$
 $substr\ STRING\ POS\ LENGTH$ substring of $STRING$, POS counted from 1
 $index\ STRING\ CHARS$ index in $STRING$ where any $CHARS$ is found, or 0
 $length\ STRING$ length of $STRING$
 $+\ TOKEN$ interpret $TOKEN$ as a string, even if it is a
 keyword like 'match' or an operator like '/'
 $(\ EXPRESSION)$ value of $EXPRESSION$

Beware that many operators need to be escaped or quoted for shells.

Comparisons are arithmetic if both ARGs are numbers, else lexicographical.

Pattern matches return the string matched between $\backslash($ and $\backslash)$ or null; if

$\backslash($ and $\backslash)$ are not used, they return the number of characters matched or 0.

Exit status is 0 if EXPRESSION is neither null nor 0, 1 if EXPRESSION is null or 0, 2 if EXPRESSION is syntactically invalid, and 3 if an error occurred.

grep

Usage: `grep [OPTION]... PATTERN [FILE]...`

Search for PATTERN in each FILE or standard input.

PATTERN is, by default, a basic regular expression (BRE).

Example: `grep -i 'hello world' menu.h main.c`

Regex selection and interpretation:

- E, --extended-regexp PATTERN is an extended regular expression (ERE)
- F, --fixed-strings PATTERN is a set of newline-separated fixed strings
- G, --basic-regexp PATTERN is a basic regular expression (BRE)
- P, --perl-regexp PATTERN is a Perl regular expression
- e, --regexp=PATTERN use PATTERN for matching
- f, --file=FILE obtain PATTERN from FILE
- i, --ignore-case ignore case distinctions
- w, --word-regexp force PATTERN to match only whole words
- x, --line-regexp force PATTERN to match only whole lines
- z, --null-data a data line ends in 0 byte, not newline

Miscellaneous:

- s, --no-messages suppress error messages

-v, --invert-match select non-matching lines
-V, --version print version information and exit
--help display this help and exit
--mmap deprecated no-op; evokes a warning

Output control:

-m, --max-count=NUM stop after NUM matches
-b, --byte-offset print the byte offset with output lines
-n, --line-number print line number with output lines
--line-buffered flush output on every line
-H, --with-filename print the file name for each match
-h, --no-filename suppress the file name prefix on output
--label=LABEL use LABEL as the standard input file name prefix
-o, --only-matching show only the part of a line matching PATTERN
-q, --quiet, --silent suppress all normal output
--binary-files=TYPE assume that binary files are TYPE;
 TYPE is 'binary', 'text', or 'without-match'
-a, --text equivalent to --binary-files=text
-I equivalent to --binary-files=without-match
-d, --directories=ACTION how to handle directories;
 ACTION is 'read', 'recurse', or 'skip'
-D, --devices=ACTION how to handle devices, FIFOs and sockets;
 ACTION is 'read' or 'skip'

-r, --recursive like --directories=recurse

-R, --dereference-recursive likewise, but follow all symlinks

 --include=FILE_PATTERN search only files that match FILE_PATTERN

 --exclude=FILE_PATTERN skip files and directories matching FILE_PATTERN

 --exclude-from=FILE skip files matching any file pattern from FILE

 --exclude-dir=PATTERN directories that match PATTERN will be skipped.

-L, --files-without-match print only names of FILES containing no match

-l, --files-with-matches print only names of FILES containing matches

-c, --count print only a count of matching lines per FILE

-T, --initial-tab make tabs line up (if needed)

-Z, --null print 0 byte after FILE name

Context control:

-B, --before-context=NUM print NUM lines of leading context

-A, --after-context=NUM print NUM lines of trailing context

-C, --context=NUM print NUM lines of output context

-NUM same as --context=NUM

 --color[=WHEN],

 --colour[=WHEN] use markers to highlight the matching strings;

 WHEN is 'always', 'never', or 'auto'

-U, --binary do not strip CR characters at EOL (MSDOS/Windows)

-u, --unix-byte-offsets report offsets as if CRs were not there

 (MSDOS/Windows)

'egrep' means 'grep -E'. 'fgrep' means 'grep -F'.

Direct invocation as either 'egrep' or 'fgrep' is deprecated.

When FILE is -, read standard input. With no FILE, read . if a command-line

-r is given, - otherwise. If fewer than two FILEs are given, assume -h.

Exit status is 0 if any line is selected, 1 otherwise;

if any error occurs and -q is not given, the exit status is 2.

kill

kill: usage: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill -l [sigspec]

ls

Usage: ls [OPTION]... [FILE]...

List information about the FILEs (the current directory by default).

Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all do not ignore entries starting with .

-A, --almost-all do not list implied . and ..

--author with -l, print the author of each file

-b, --escape print C-style escapes for nongraphic characters

--block-size=SIZE scale sizes by SIZE before printing them. E.g.,

'--block-size=M' prints sizes in units of

1,048,576 bytes. See SIZE format below.

-B, --ignore-backups do not list implied entries ending with ~

-c with -lt: sort by, and show, ctime (time of last modification of file status information)
with -l: show ctime and sort by name
otherwise: sort by ctime, newest first

-C list entries by columns

--color[=WHEN] colorize the output. WHEN defaults to 'always' or can be 'never' or 'auto'. More info below

-d, --directory list directory entries instead of contents, and do not dereference symbolic links

-D, --dired generate output designed for Emacs' dired mode

-f do not sort, enable -aU, disable -ls --color

-F, --classify append indicator (one of */=>@|) to entries

--file-type likewise, except do not append '*'

--format=WORD across -x, commas -m, horizontal -x, long -l, single-column -l, verbose -l, vertical -C

--full-time like -l --time-style=full-iso

-g like -l, but do not list owner

--group-directories-first
group directories before files.
augment with a --sort option, but any use of --sort=none (-U) disables grouping

-G, --no-group in a long listing, don't print group names

`-h, --human-readable` with `-l`, print sizes in human readable format
(e.g., 1K 234M 2G)

`--si` likewise, but use powers of 1000 not 1024

`-H, --dereference-command-line`
follow symbolic links listed on the command line

`--dereference-command-line-symlink-to-dir`
follow each command line symbolic link
that points to a directory

`--hide=PATTERN` do not list implied entries matching shell PATTERN
(overridden by `-a` or `-A`)

`--indicator-style=WORD` append indicator with style WORD to entry names:
none (default), slash (-p),
file-type (--file-type), classify (-F)

`-i, --inode` print the index number of each file

`-I, --ignore=PATTERN` do not list implied entries matching shell PATTERN

`-k, --kibibytes` use 1024-byte blocks

`-l` use a long listing format

`-L, --dereference` when showing file information for a symbolic
link, show information for the file the link
references rather than for the link itself

`-m` fill width with a comma separated list of entries

`-n, --numeric-uid-gid` like `-l`, but list numeric user and group IDs

`-N, --literal` print raw entry names (don't treat e.g. control

characters specially)

-o like -l, but do not list group information

-p, --indicator-style=slash

 append / indicator to directories

-q, --hide-control-chars print ? instead of non graphic characters

 --show-control-chars show non graphic characters as-is (default

 unless program is 'ls' and output is a terminal)

-Q, --quote-name enclose entry names in double quotes

 --quoting-style=WORD use quoting style WORD for entry names:

 literal, locale, shell, shell-always, c, escape

-r, --reverse reverse order while sorting

-R, --recursive list subdirectories recursively

-s, --size print the allocated size of each file, in blocks

-S sort by file size

 --sort=WORD sort by WORD instead of name: none -U,

 extension -X, size -S, time -t, version -v

 --time=WORD with -l, show time as WORD instead of modification

 time: atime -u, access -u, use -u, ctime -c,

 or status -c; use specified time as sort key

 if --sort=time

 --time-style=STYLE with -l, show times using style STYLE:

 full-iso, long-iso, iso, locale, +FORMAT.

 FORMAT is interpreted like 'date'; if FORMAT is

FORMAT1<newline>FORMAT2, FORMAT1 applies to
 non-recent files and FORMAT2 to recent files;
 if STYLE is prefixed with 'posix-', STYLE
 takes effect only outside the POSIX locale

-t sort by modification time, newest first

-T, --tabsize=COLS assume tab stops at each COLS instead of 8

-u with -lt: sort by, and show, access time
 with -l: show access time and sort by name
 otherwise: sort by access time

-U do not sort; list entries in directory order

-v natural sort of (version) numbers within text

-w, --width=COLS assume screen width instead of current value

-x list entries by lines instead of by columns

-X sort alphabetically by entry extension

-Z, --context print any SELinux security context of each file

-1 list one file per line

--help display this help and exit

--version output version information and exit

SIZE is an integer and optional unit (example: 10M is 10*1024*1024). Units
 are K, M, G, T, P, E, Z, Y (powers of 1024) or KB, MB, ... (powers of 1000).

Using color to distinguish file types is disabled both by default and

with `--color=never`. With `--color=auto`, `ls` emits color codes only when standard output is connected to a terminal. The `LS_COLORS` environment variable can change the settings. Use the `dircolors` command to set it.

Exit status:

- 0 if OK,
- 1 if minor problems (e.g., cannot access subdirectory),
- 2 if serious trouble (e.g., cannot access command-line argument).

mv

Usage: `mv [OPTION]... [-T] SOURCE DEST`

or: `mv [OPTION]... SOURCE... DIRECTORY`

or: `mv [OPTION]... -t DIRECTORY SOURCE...`

Rename `SOURCE` to `DEST`, or move `SOURCE(s)` to `DIRECTORY`.

Mandatory arguments to long options are mandatory for short options too.

- `--backup[=CONTROL]` make a backup of each existing destination file
- `-b` like `--backup` but does not accept an argument
- `-f, --force` do not prompt before overwriting
- `-i, --interactive` prompt before overwrite
- `-n, --no-clobber` do not overwrite an existing file

If you specify more than one of `-i`, `-f`, `-n`, only the final one takes effect.

`--strip-trailing-slashes` remove any trailing slashes from each SOURCE

argument

`-S, --suffix=SUFFIX` override the usual backup suffix

`-t, --target-directory=DIRECTORY` move all SOURCE arguments into DIRECTORY

`-T, --no-target-directory` treat DEST as a normal file

`-u, --update` move only when the SOURCE file is newer

than the destination file or when the

destination file is missing

`-v, --verbose` explain what is being done

`--help` display this help and exit

`--version` output version information and exit

The backup suffix is '~', unless set with `--suffix` or `SIMPLE_BACKUP_SUFFIX`.

The version control method may be selected via the `--backup` option or through

the `VERSION_CONTROL` environment variable. Here are the values:

`none, off` never make backups (even if `--backup` is given)

`numbered, t` make numbered backups

`existing, nil` numbered if numbered backups exist, simple otherwise

`simple, never` always make simple backups

netstat

usage: `netstat [-vWeenNcCF] [<Af>] -r` netstat {`-V|--version|-h|--help`}

netstat [-vWnNcaeol] [<Socket> ...]

netstat { [-vWeenNac] -i | [-cWnNe] -M | -s }

- r, --route display routing table
- i, --interfaces display interface table
- g, --groups display multicast group memberships
- s, --statistics display networking statistics (like SNMP)
- M, --masquerade display masqueraded connections

- v, --verbose be verbose
- W, --wide don't truncate IP addresses
- n, --numeric don't resolve names
- numeric-hosts don't resolve host names
- numeric-ports don't resolve port names
- numeric-users don't resolve user names
- N, --symbolic resolve hardware names
- e, --extend display other/more information
- p, --programs display PID/Program name for sockets
- c, --continuous continuous listing

- l, --listening display listening server sockets
- a, --all, --listening display all sockets (default: connected)
- o, --timers display timers

- F, --fib display Forwarding Information Base (default)
- C, --cache display routing cache instead of FIB

<Socket>={-t|--tcp} {-u|--udp} {-w|--raw} {-x|--unix} --ax25 --ipx --netrom

<AF>=Use '-6|-4' or '-A <af>' or '--<af>'; default: inet

pppd

pppd version 2.4.5

Usage: pppd [options], where options are:

- <device> Communicate over the named device
- <speed> Set the baud rate to <speed>
- <loc>:<rem> Set the local and/or remote interface IP addresses. Either one may be omitted.
- asynmap <n> Set the desired async map to hex <n>
- auth Require authentication from peer
- connect <p> Invoke shell command <p> to set up the serial line
- crtsects Use hardware RTS/CTS flow control
- defaultroute Add default route through interface
- file <f> Take options from file <f>
- modem Use modem control lines
- mru <n> Set MRU value to <n> for negotiation

when run without the `-f` option, or when in a system runlevel other than 0 or 6, it will actually execute `/sbin/shutdown`.

rm

Usage: `rm [OPTION]... FILE...`

Remove (unlink) the FILE(s).

- `-f, --force` ignore nonexistent files and arguments, never prompt
- `-i` prompt before every removal
- `-I` prompt once before removing more than three files, or when removing recursively. Less intrusive than `-i`, while still giving protection against most mistakes
- `--interactive[=WHEN]` prompt according to WHEN: never, once (`-I`), or always (`-i`). Without WHEN, prompt always
- `--one-file-system` when removing a hierarchy recursively, skip any directory that is on a file system different from that of the corresponding command line argument
- `--no-preserve-root` do not treat `'/'` specially
- `--preserve-root` do not remove `'/'` (default)
- `-r, -R, --recursive` remove directories and their contents recursively
- `-d, --dir` remove empty directories
- `-v, --verbose` explain what is being done
- `--help` display this help and exit
- `--version` output version information and exit

By default, `rm` does not remove directories. Use the `--recursive` (`-r` or `-R`) option to remove each listed directory, too, along with all of its contents.

To remove a file whose name starts with a '-', for example '-foo', use one of these commands:

```
rm -- -foo
```

```
rm ./-foo
```

Note that if you use `rm` to remove a file, it might be possible to recover some of its contents, given sufficient expertise and/or time. For greater assurance that the contents are truly unrecoverable, consider using `shred`.

sed

Usage: `sed [OPTION]... {script-only-if-no-other-script} [input-file]...`

```
-n, --quiet, --silent
```

suppress automatic printing of pattern space

```
-e script, --expression=script
```

add the script to the commands to be executed

```
-f script-file, --file=script-file
```

add the contents of script-file to the commands to be executed

```
--follow-symlinks
```

follow symlinks when processing in place

`-i[SUFFIX], --in-place[=SUFFIX]`

edit files in place (makes backup if SUFFIX supplied)

`-l N, --line-length=N`

specify the desired line-wrap length for the ``l'` command

`--posix`

disable all GNU extensions.

`-r, --regexp-extended`

use extended regular expressions in the script.

`-s, --separate`

consider files as separate rather than as a single continuous

long stream.

`-u, --unbuffered`

load minimal amounts of data from the input files and flush

the output buffers more often

`-z, --null-data`

separate lines by NUL characters

`--help` display this help and exit

`--version` output version information and exit

If no `-e, --expression, -f, or --file` option is given, then the first non-option argument is taken as the sed script to interpret. All remaining arguments are names of input files; if no input files are

specified, then the standard input is read.

shutdown

Usage: shutdown [OPTION]... TIME [MESSAGE]

Bring the system down.

Options:

- r reboot after shutdown
- h halt or power off after shutdown
- H halt after shutdown (implies -h)
- P power off after shutdown (implies -h)
- c cancel a running shutdown
- k only send warnings, don't shutdown
- q, --quiet reduce output to errors only
- v, --verbose increase output to include informational messages
- help display this help and exit
- version output version information and exit

TIME may have different formats, the most common is simply the word 'now' which will bring the system down immediately. Other valid formats are +m, where m is the number of minutes to wait until shutting down and hh:mm which specifies the time on the 24hr clock.

Logged in users are warned by a message sent to their terminal, you may include an optional MESSAGE included with this. Messages can be sent without actually bringing the system down by using the -k option.

If TIME is given, the command will remain in the foreground until the shutdown occurs. It can be cancelled by Control-C, or by another user using the -c option.

The system is brought down into maintenance (single-user) mode by default, you can change this with either the -r or -h option which specify a reboot or system halt respectively. The -h option can be further modified with -H or -P to specify whether to halt the system, or to power it off afterwards. The default is left up to the shutdown scripts.

stat

Usage: stat [OPTION]... FILE...

Display file or file system status.

Mandatory arguments to long options are mandatory for short options too.

-L, --dereference follow links

-f, --file-system display file system status instead of file status

-c --format=FORMAT use the specified FORMAT instead of the default;

output a newline after each use of FORMAT

`--printf=FORMAT` like `--format`, but interpret backslash escapes,

and do not output a mandatory trailing newline.

If you want a newline, include `\n` in `FORMAT`

`-t, --terse` print the information in terse form

`--help` display this help and exit

`--version` output version information and exit

The valid format sequences for files (without `--file-system`):

`%a` access rights in octal

`%A` access rights in human readable form

`%b` number of blocks allocated (see `%B`)

`%B` the size in bytes of each block reported by `%b`

`%C` SELinux security context string

`%d` device number in decimal

`%D` device number in hex

`%f` raw mode in hex

`%F` file type

`%g` group ID of owner

`%G` group name of owner

`%h` number of hard links

`%i` inode number

`%m` mount point

%n file name

%N quoted file name with dereference if symbolic link

%o optimal I/O transfer size hint

%s total size, in bytes

%t major device type in hex

%T minor device type in hex

%u user ID of owner

%U user name of owner

%w time of file birth, human-readable; - if unknown

%W time of file birth, seconds since Epoch; 0 if unknown

%x time of last access, human-readable

%X time of last access, seconds since Epoch

%y time of last modification, human-readable

%Y time of last modification, seconds since Epoch

%z time of last change, human-readable

%Z time of last change, seconds since Epoch

Valid format sequences for file systems:

%a free blocks available to non-superuser

%b total data blocks in file system

%c total file nodes in file system

%d free file nodes in file system

%f free blocks in file system
%i file system ID in hex
%l maximum length of filenames
%n file name
%s block size (for faster transfers)
%S fundamental block size (for block counts)
%t file system type in hex
%T file system type in human readable form

NOTE: your shell may have its own version of stat, which usually supersedes the version described here. Please refer to your shell's documentation for details about the options it supports.

tr

Usage: tr [OPTION]... SET1 [SET2]

Translate, squeeze, and/or delete characters from standard input, writing to standard output.

-c, -C, --complement use the complement of SET1
-d, --delete delete characters in SET1, do not translate
-s, --squeeze-repeats replace each input sequence of a repeated character that is listed in SET1 with a single occurrence of that character

-t, --truncate-set1 first truncate SET1 to length of SET2

--help display this help and exit

--version output version information and exit

SETs are specified as strings of characters. Most represent themselves.

Interpreted sequences are:

\NNN character with octal value NNN (1 to 3 octal digits)

\\ backslash

\a audible BEL

\b backspace

\f form feed

\n new line

\r return

\t horizontal tab

\v vertical tab

CHAR1-CHAR2 all characters from CHAR1 to CHAR2 in ascending order

[CHAR*] in SET2, copies of CHAR until length of SET1

[CHAR*REPEAT] REPEAT copies of CHAR, REPEAT octal if starting with 0

[:alnum:] all letters and digits

[:alpha:] all letters

[:blank:] all horizontal whitespace

[:cntrl:] all control characters

[digit:] all digits

[graph:] all printable characters, not including space

[lower:] all lower case letters

[print:] all printable characters, including space

[punct:] all punctuation characters

[space:] all horizontal or vertical whitespace

[upper:] all upper case letters

[xdigit:] all hexadecimal digits

[=CHAR=] all characters which are equivalent to CHAR

Translation occurs if -d is not given and both SET1 and SET2 appear.

-t may be used only when translating. SET2 is extended to length of SET1 by repeating its last character as necessary. Excess characters of SET2 are ignored. Only [:lower:] and [:upper:] are guaranteed to expand in ascending order; used in SET2 while translating, they may only be used in pairs to specify case conversion. -s uses SET1 if not translating nor deleting; else squeezing uses SET2 and occurs after translation or deletion.

wc

Usage: wc [OPTION]... [FILE]...

or: wc [OPTION]... --files0-from=F

Print newline, word, and byte counts for each FILE, and a total line if

more than one FILE is specified. With no FILE, or when FILE is -,
read standard input. A word is a non-zero-length sequence of characters
delimited by white space.

The options below may be used to select which counts are printed, always in
the following order: newline, word, character, byte, maximum line length.

- c, --bytes print the byte counts
- m, --chars print the character counts
- l, --lines print the newline counts
- files0-from=F read input from the files specified by
 NUL-terminated names in file F;
 If F is - then read names from standard input
- L, --max-line-length print the length of the longest line
- w, --words print the word counts
- help display this help and exit
- version output version information and exit

Report wc bugs to bug-coreutils@gnu.org

GNU coreutils home page: [<http://www.gnu.org/software/coreutils/>](http://www.gnu.org/software/coreutils/)

General help using GNU software: [<http://www.gnu.org/gethelp/>](http://www.gnu.org/gethelp/)