ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ



Διπλωματική εργασία:

Android web server

Εισηγητής:

Κωνσταντίνος Τσερπές

Ημερομηνία κατάθεσης Εργασίας:

Φοιτητής:

Δημήτρης Νικηφοράκης (Α.Μ.13110)

Περίληψη

Η ευκολία σχεδιασμού εφαρμογών η προοδευτική αύξηση των δυνατοτήτων που παρέχουν οι συσκευές με λειτουργικό android έχουν δώσει στους επαγγελματίες που σχεδιάζουν εφαρμογές για αυτή τη πλατφόρμα με ένα μεγάλο φάσμα εφαρμογών τις οποίες να μπορούν να εκμεταλλευτούν. Λόγω του μεγάλου αριθμού δυνατοτήτων δικτύωσης που προσφέρουν (wi-fi, bluetooth, υποστήριξη 3G/4G δικτύων) μια από τις πιο διαδεδομένες τέτοιες δυνατότητες είναι αυτή του web server. Μέσω μιας τέτοιας εφαρμογής ο χρήστης μπορεί διαχειριστεί πλευρές της συσκευής του αλλά και να επεκτείνει τη λειτουργικότητα της με μεθόδους οι οποίες είναι απλές στη χρήση, ευέλικτες και διαδεδομένες.

Για τις ανάγκες μιας τέτοιας εφαρμογής υλοποιήθηκε ένας δικτυακός εξυπηρετητής σε node.js όσο και ενα απλό web interface που να εξυπηρετεί τις ανάγκες διαχείρησης του. Επίσης για τις ανάγκες μιας τέτοιας εφαρμογής εξετάστηκαν τρόποι εγκατάστασης του node.js σε Android.

Το αποτέλεσμα είναι ενας εξυπηρετητής το οποίο μπορεί να διαμοιράζει αρχεία και να εξυπηρετεί πολλαπλούς χρήστες ταυτόχρονα και ο οποίος μπορεί να χρησιμοποιήσει ενα βασικό μηχανισμό services για πιθανές επεκτάσεις. Η συγκεκριμένη εφαρμογή δοκιμάστηκε σε πραγματικές συνθήκες και το αποτέλεσμα είναι μια εφαρμογή η οποία μπορεί να ανταπεξέλθει στο μέτρο των δυνατοτήτων της εκάστοτε συσκευής

Abstract

The ease of application design, the progressive increase of features that are present on Android enabled devices has given the oportunity to proffesional developers a large array of applications. Because of the many ways of networking present on such devices (wi-fi, bluetooth, 3G/4G network support) such a common application is that of a web server. Through such an application a user can administer aspects of his device and extend it's functionality through methods that are easy to use, versatile and common.

For the needs of such an application a web server in node.js was implemented allong with a simple web interface for administrative purpoces. For the same reasons installation for node.js on android devices were examined.

The result is a web server that can share files andservice several users concurently. This server also displays a simple service mechanism for extentions. This application was given testing on running enviroments and the result is an application that meets its expectations to the capacity of the underlying device.

Περιεχόμενα

1 Εισαγωγή..... 5
2 Το πρωτόκολλο HTTP..... 6
3 Προδιαγραφές..... 9
4 Εργαλεία ανάπτυξης εφαρμογών για android..... 11
4.1 Γλώσσες προγραμματισμού..... 11
4.2 Chroot..... 12
5 Ο κώδικας της εργασίας..... 14
5.1 Το σκεπτικό..... 14
5.2 Παραπλήσιο έργο..... 15
6 Προσέγγιση..... 17
6.1 Λεπτομέρειες της αρχιτεκτονικής..... 17
6.2 Υποθέσεις και παραδοχές..... 18
7 Λεπτομέρειες υλοποίησης..... 20
7.1 Server-side..... 20
7.2 Administration..... 22
7.3 Node.js build..... 24
7.4 Εγκατάσταση Chroot …..... 25
8 Πειραματικά αποτελέσματα..... 27
8.1 Συνθήκες πειραμάτων..... 27
8.2 Αποτελέσματα..... 28
9 Επίδειξη συστήματος..... 30
10 Συμπεράσματα..... 35
11 Βιβλιογραφία..... 37

Εισαγωγή

I

Με τον όρο smartphones αναφερόμαστε σε κινητά τηλέφωνα τα οποία χρησιμοποιούν κάποια μορφή προηγμένου λειτουργικού συστήματος. Κατά κανόνα συνδυάζουν λειτουργίες και χαρακτηριστικά κινητών τηλεφώνων και άλλων ειδών κινητών συσκευών όπως τα personal digital assistant (PDA), media player και συσκευές καθοδήγησης GPS. Τα περισσότερα smartphone χρησιμοποιούν οθόνες αφής σαν την κύρια μορφή εισόδου / εξόδου, μπορούν να τρέξουν third party apps και αρκετά συχνά μπορούν να χρησιμοποιηθούν για κλήσεις βίντεο. Τα τελευταία χρόνια έχουν επικρατήσει μοντέλα τα οποία υποστηρίζουν δυνατότητες δικτύωσης (Wi-fi, δίκτυα 3ης, 4ης γενιάς mobile broadband internet) και συχνά εξοπλίζονται με αισθητήρια κίνησης, κάμερες, GPS, και συστήματα πληρωμών.

Το Android είναι ένα λειτουργικό ανοιχτού κώδικα που αναπτύχθηκε από τον Andy Rubin με την υποστήριξη της Google και κατασκευαστές hardware και software (όπως η Intel, HTC, ARM, Motorola and Samsung) οι οποίοι απαρτίζουν την Open Handset Alliance. Τον Οκτώβριο του 2008 η HTC παρουσίασε το HTC Dream, το πρώτο κινητό που να χρησιμοποιεί το συγκεκριμένο λειτουργικό. Κατά κανόνα συνηθισμένες εφαρμογές που έρχονταν προ εγκατεστημένες ήταν συνδεδεμένες με υπηρεσίες της Google όπως για χάρτες , ημερολόγιο και ανάγνωση email από το Gmail. Επίσης υπήρχε πάντα ένας web browser με υποστήριξη για HTML. Το Android υποστηρίζει την δημιουργία εφαρμογών οι οποίες συνήθως προσφέρονται μέσω του Google Play, μια υπηρεσία που ξεκίνησε από το 2008 με το όνομα Android Market. Μέχρι το 2010 το Android είχε γίνει το πιο διαδεδομένο λειτουργικό για συσκευές όπως κινητά και tablet.

Η ευκολία σχεδιασμού εφαρμογών και η προοδευτική αύξηση των δυνατοτήτων που παρέχουν οι συγκεκριμένες συσκευές έχουν δώσει στους επαγγελματίες που σχεδιάζουν εφαρμογές για αυτή τη πλατφόρμα με ένα μεγάλο φάσμα δυνατοτήτων τις οποίες να μπορούν να εκμεταλλευτούν. Οι συσκευές android που βρίσκονται στην αγορά αυτή τη στιγμή στην αγορά μπορούν να δικτυωθούν με ένα μεγάλο αριθμό τρόπων, μέσω Wi-fi, bluetooth, μέσω δικτύων 3G/4G, κάτι που επιτρέπει στις πιο δυνατές από αυτές τις συσκευές να μπορούν να υπερβούν τα απλά όρια συσκευών που απλά καταναλώνουν υπηρεσίες και δεδομένα από άλλα πιο ισχυρά υπολογιστικά συστήματα και να παρέχουν με τη σειρά τους δεδομένα και υπηρεσίες μέσω TCP/IP σε άλλα συστήματα σε ένα δίκτυο. Μια από τις πιο διαδεδομένες τέτοιες υπηρεσίες είναι αυτή του web server.

Το πρωτόκολλο HTTP

II

Με τον όρο Hypertext (υπερκείμενο) αναφερόμαστε σε δομημένο κείμενο που χρησιμοποιεί λογικούς συνδέσμους (hyperlinks) ανάμεσα σε κόμβους που περιέχουν κείμενο. Με τη βοήθεια του HTTP δίνεται η δυνατότητα ανταλλαγής και μεταφοράς υπερκείμενου.

Τα πρώτυπα ανάπτυξης του HTTP συντονίζονταν απο την Internet Engineering Task Force (IETF) και το World Wide Web Consortium (W3C), φτάνοντας σε ορίμανση με την δημοσίευση μιας σειράς RFC (Requests for Comments) με κυριότερο το RFC 2616 το πρώτο χρονικά κείμενο στο οποίο δίνεται η περιγραφή του HTTP/1.1, την πιο χρησιμοποιούμενη έκδοση του HTTP σήμερα. Τον Ιούνιο του 2014 το RFC 2616 υπερσκελίστηκε σαν κείμενο αναφοράς απο τα RFCs 7230, 7231, 7232, 7233, 7234, and 7235.

Η κύρια λειτουργία ενός εξυπηρετητή είναι η αποθήκευση, επεξεργασία και παροχή σελίδων στους πελάτες το. Η επικοινωνία μεταξύ των δυο μερών γίνεται με τη βοήθεια του Hypertext Transfer Protocol (HTTP) ενώ το είδος του περιεχόμενου το οποίο ανταλλάσσεται συνήθως είναι έγγραφα HTML (το οποίο συνεπάγεται την αποστολή εικόνων, αρχείων css, scripts τα οποία συμπληρώνουν το κείμενο της σελίδας) .

Ένας user agent όπως ένας browsesr ή ένα πρόγραμμα (ένας crowler για παράδειγμα)αρχικοποιεί την επικοινωνία τους στέλνοντας ένα αίτημα για ένα resource χρησιμοποιώντας το HTTP και ο server ανταποκρίνεται αποστέλλοντας το συγκεκριμένο resource ή ένα μήνυμα λάθους εάν είναι αδύνατο να το βρει. Το resource αυτό είναι συνήθως ένα αρχείο στη δευτερεύουσα μνήμη αλλά αυτό σχετίζεται με το πως έχει υλοποιηθεί ο server.

Αν και η κύρια λειτουργία ενός web server είναι η παροχή περιεχόμενου, η πλήρης υλοποίηση του HTTP περιλαμβάνει τρόπους ανταλλαγής δεδομένων μεταξύ client-server κάτι που συμβαίνει ως επί το πλείστον όταν συμπληρώνουμε μια φόρμα ή όταν ανεβάζουμε αρχεία.

Πολλοί server γενικής χρήσης υποστηρίζουν server-side scripting χρησιμοποιώντας γλώσσες scripting όπως η ASP και η PHP, το οποίο επιτρέπει την επέκταση των λειτουργιών του server χωρίς την αλλοίωση του κυρίως σώματος του προγράμματος του. Οι συνηθέστερες εφαρμογές που βρίσκει γόνιμο χώρο αυτή η λειτουργικρότητα είναι η κατασκευή κειμένων HTML δυναμικά (χωρίς την

παρέμβαση κάποιου χρήστη) ή για την παρουσίαση δεδομένων από βάσεις δεδομένων.

Μια άλλη εφαρμογή web server είναι σε ενσωματωμένα συστήματα όπως εκτυπωτές, routers και κάμερες τα οποία παρέχουν περιεχόμενο μέσω τοπικών δικτύων. Ένας εξυπηρετητής μπορεί να χρησιμοποιηθεί σαν μέρος της παρακολούθησης της συσκευής ή/και της διαχείρισης της. Αυτό συνεπάγεται την απαλοιφή της ανάγκης για περεταίρω λογισμικό που να απαιτείται για την διαχείριση της πέρα από ένα browser.

Τα οφέλη από τον σχεδιασμό μιας τέτοιας εφαρμογής για συσκευές με Android είναι η δυνατότητα πρόσβασης στις λειτουργίες που παρέχουν οι συσκευές με android με την χρήση πρωτοκόλλων τα οποία είναι ανοιχτά, με σαφή τρόπο λειτουργίας και συμπεριφοράς και απόλυτα κατανοητά από άλλες συσκευές οι οποίες τα ακολουθούν.

Με αυτό το τρόπο είναι δυνατό να μπορούμε για παράδειγμα να αντλούμε πληροφορίες από μια συσκευή:

- Για τη θέση της στο χώρο σε βάθος χρόνου (με τη χρήση του GPS).
- Για τη χρήση της κάμερας που έχει, όπως και να έχουμε πρόσβαση στις φωτογραφίες που τραβήχτηκαν με αυτή.
- Αριθμητικά και γραφικά στατιστικών για τις λειτουργίες επιμέρους υποσυστημάτων της συσκευής, για τον όγκο των δεδομένων που περάσανε από και προς το ασύρματο δίκτυο όπως και για τον χρόνο ομιλίας του χρήστη.
- Τέλος, σημαντικός παράγοντας για την ανάπτυξη web server σε Android αναλογικά με άλλες συσκευές όπως της Apple ή σε συσκευές με ιδιόκτητα λειτουργικά είναι η δυνατότητα χρήσης C και Java πάνω από συσκευές στις οποίες τρέχει ένας προσαρμοσμένος πυρήνας Linux ανοιχτού κώδικα, κάτι που συνεπάγεται δυνατότητα αξιολόγησης τόσο του web server όσο και του πυρήνα απέναντι σε ένα σώμα γνώσης που παρέχει τη δυνατότητα ελαχιστοποίησης τουλάχιστον των πιο συνηθισμένων σφαλμάτων στα οποία ένας developer μπορεί να υποπέσει.

Ο χρήστης μπορεί να ωφεληθεί από μια τέτοια εφαρμογή μέσω της οποίας μπορεί να του δοθεί ευέλικτη πρόσβαση σε υπηρεσίες και πληροφορίες τις οποίες να αντλεί από πηγές στο internet και τις οποίες να ανασυνθέτει σε περιλήψεις οι οποίες να έχουν νόημα για τον χρήστη (πχ εκδηλώσεις σε μια περιοχή σε μια συγκεκριμένη χρονική στιγμή, ο καιρός εκείνη την ημέρα καθώς και προτεινόμενες υπηρεσίες ταξί).

Με την παρουσία όμως τέτοιων λειτουργιών σε συσκευές οι οποίες δεν είχαν σχεδιαστεί αρχικά για τέτοιου είδους λειτουργίες πρέπει να δοθεί σημασία σε παράπλευρες λεπτομέρειες οι οποίες σχετίζονται με το συγκεκριμένο είδος εφαρμογών:

- Λεπτομέρειες που σχετίζονται με το μεγάλο φάσμα συσκευών που σχετίζονται με κοινό παρανομαστή το **λειτουργικό** το οποίο τρέχει πάνω τους. Μια συσκευή που τρέχει Android δεν είναι υποχρεωτικό ότι θα μπορεί να εξυπηρετήσει άλλες συσκευές γιατί πολύ απλά δεν θα έρχεται με ενσωματωμένη υποστήριξη για επικοινωνία TCP/IP ή μπορεί να μην χρησιμοποιήσει άλλα υποσυστήματα όπως αποθηκευτικούς χώρους, κάμερες ή GPS για να παρέχει περιεχόμενο και κάποια μορφή λειτουργικότητας σε άλλες συσκευές.
- Λεπτομέρειες που σχετίζονται με την ασφάλεια της συσκευής. Το Android παρά το διαφορετικό όνομα δεν παύει να είναι ένας πυρήνας Linux προσαρμοσμένος για χρήση σε ενσωματωμένα συστήματα με περιορισμένους αποθηκευτικούς χώρους, RAM, επεξεργαστή και χρόνο λειτουργίας αφού η πηγή ενέργειας είναι κάποια μορφή μπαταρίας. Αυτό οδηγεί σε τροποποιήσεις και προσαρμογές οι οποίες με τη σειρά τους οδηγούν σε δυνητικά κενά ασφάλειας και νέους τρόπους με τους οποίους οι συγκεκριμένες συσκευές να μπορούν να παραβιαστούν.

Κατά συνέπεια η χρήση μιας συσκευής με Android σαν web server είναι συχνά εφικτή, μπορεί να δίνει λειτουργικά πλεονεκτήματα που μπορεί να μην υπάρχουν σε μεγαλύτερα συστήματα (ή για να αναπληρώσει σε λειτουργικότητα κενά που δεν υπάρχουν σε smartphones και tablets) αλλά αυτό δεν έρχεται χωρίς το ανάλογο σχεδιαστικό τίμημα.

Προδιαγραφές

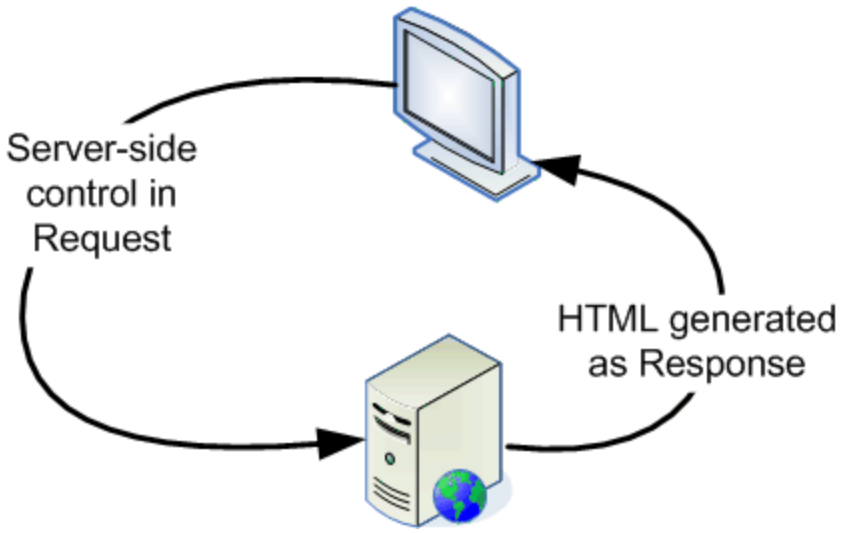
III

Μέχρι τώρα έχουμε δει συνοπτικά τι είναι το πρωτόκολλο HTTP, πως λειτουργεί ένας εξυπηρετητής και ποιες λειτουργίες εκτελεί συνήθως. Επίσης παρουσιάστηκαν πιθανά οφέλη από την ύπαρξη μιας τέτοιας εφαρμογής σε συσκευές στις οποίες τρέχουν Android.

Ο στόχος της παρούσας πτυχιακής είναι η υλοποίηση ενός τέτοιου απλού web server για περιβάλλον android. Εν τούτοις, ακόμα και ένα είδος εφαρμογών με τόσο συγκεκριμένο εύρος δυνατοτήτων δεν γίνεται να καλυφθεί σε μια πτυχιακή. Για λόγους απλότητας ο βασικός πυρήνας λειτουργιών που θα χρησιμοποιήσουμε θα είναι για να:

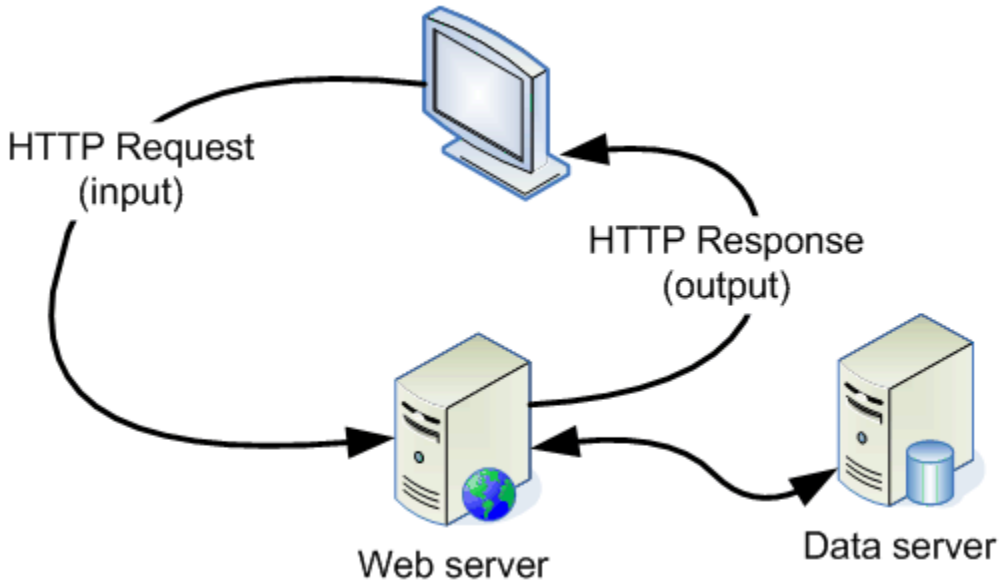
- Μπορεί να παρουσιάσει στον χρήστη περιεχόμενο, εικόνες, HTML έγγραφα, αρχεία κειμένου.
- Μπορεί να στείλει στον χρήστη αρχεία (pdf, zip, ps για παράδειγμα).

Οι παραπάνω δυο λειτουργίες καλύπτουν τις πιο βασικές και συνηθέστερες λειτουργίες για τις οποίες μπορεί κάποιος να χρειαστεί να χρησιμοποιήσει ένα web server. Αυτό συνεπάγεται πως τελικά όταν ξεκινήσει να δουλεύει ο web server θα μπορεί να παρέχει κάποιο είδος πληροφορίας στον χρήστη.



Μια επιθυμητή λειτουργία παρουσιάζεται συχνά είναι η δυνατότητα επέκτασης της λειτουργικότητας του server. Αυτή η λειτουργικότητα εμφανίζεται συνήθως στους μεγαλύτερους web server όπως ο Apache εν τούτοις γίνεται αναφορά εδώ για τρεις λόγους:

- Γιατί μας παρέχει τη δυνατότητα τροποποίησης της συμπεριφοράς του web server στις ανάγκες και στο βαθμό που είναι χρήσιμες οι υπηρεσίες του στον χρήστη. Αν αυτό σημαίνει πως κάποια μέρη του θα πρέπει να αντικατασταθούν προς όφελος κάποιων άλλων και κατά συνέπεια να αυξήσει την αξία του τότε η παρουσία μιας τέτοιας δυνατότητας είναι αν όχι επιθυμητή, σίγουρα είναι αναγκαία.
- Αναφέρθηκε προηγουμένως πως μια εφαρμογή που μπορεί να έχει ένας web server σε μια συσκευή Android σχετίζεται με την αναφορά πληροφοριών για την λειτουργία της συσκευής και για την λειτουργία επιμέρους υποσυστημάτων. Η δυνατότητα αλλαγής της συμπεριφοράς του web server **μας δίνει την δυνατότητα προσαρμογής της εφαρμογής για να εξυπηρετήσει συσκευές με νέες λειτουργίες αλλά και για να αφαιρέσει λειτουργίες από συσκευές στις οποίες δεν υπάρχουν.**
- Τέλος είναι επιθυμητή η δυνατότητα κατανάλωσης υπηρεσιών (REST, SOAP για παράδειγμα) οι οποίες υπάρχουν και οι οποίες μπορεί να υποστηριχτούν από ένα τέτοιο μηχανισμό (ενημέρωση για τον καιρό, ηλεκτρονικές συναλλαγές, δρομολόγια κλπ).



Εργαλεία ανάπτυξης εφαρμογών για android

IV

4.1 Γλώσσες προγραμματισμού

Ο πιο συνηθισμένος τρόπος με τον οποίο ένας προγραμματιστής μπορεί να αναπτύξει εφαρμογές είναι με την βοήθεια του Android software development kit (SDK) το οποίο η Google παρουσίασε μαζί με τις πρώτες συσκευές. Με τη βοήθεια του είναι δυνατή η ανάπτυξη εφαρμογών σε μια διάλεκτο της Java ειδικά προσαρμοσμένης για τις συσκευές στις οποίες τρέχει το Android.

Αυτή η προσέγγιση έχει συντριπτικά πλεονεκτήματα σε σχέση με άλλες λιγότερο γνωστές λύσεις:

1. Οι εφαρμογές που γράφονται με αυτό το τρόπο είναι δυνατό να τρέξουν πρακτικά σε κάθε συσκευή που τρέχει Android
2. Παρέχει προστασία αφού οι εφαρμογές τρέχουν μέσα σε ένα runtime enviroment το οποίο παρέχει δικλίδες ασφαλείας απέναντι σε κακόβουλες ενέργειες και συνηθισμένα προγραμματιστικά σφάλματα.
3. Παρέχει τη δυνατότητα ανάπτυξης εφαρμογών με γραφικό περιβάλλον, δυνατότητα που δεν υπάρχει σε άλλες λύσεις

Η δεύτερη πιο συνήθης λύση ανάπτυξη εφαρμογών για συσκευές android είναι με τη βοήθεια του NDK (native development kit) το οποίο παρέχει τη δυνατότητα ανάπτυξης native εφαρμογών με C/C++ κάτι που μπορεί να οδηγήσει στη δημιουργία εφαρμογών οι οποίες είναι γρήγορες λόγω του native κώδικα που εκτελείται αλλά οι οποίες συχνά είναι προσαρμοσμένες για οικογένειες συσκευών ή οι οποίες χρησιμοποιούν υπερβολικά πολύπλοκα API για να επικοινωνήσουν με το λειτουργικό ή τα υποσυστήματα της συσκευής.

Οι παραπάνω δυο λύσεις έχουν παράγει στο παρελθόν προϊόντα τα οποία να ενσωματώνουν web server και scripted γλώσσες (όπως τα [3], [4]) και με την βοήθεια των οποίων να επιτρέπουν να εγκατασταθούν σε μια συσκευή με Android γνωστά εργαλεία όπως η PHP ο Apache (ή ο nginx) και η mySQL. Επίσης είναι δυνατόν με τη βοήθεια τους να τρέξουν και εκπαιδευτικές εφαρμογές σε C/C++ και Java με τις οποίες να κατανοήσει καλύτερα πως λειτουργεί ένας web server (όπως τα [5],[6], [7],[8],[9]).

Με βάση τα παραπάνω 2 εργαλεία στο παρελθόν έχουν συσπειρωθεί αρκετές φορές κοινότητες προγραμματιστών οι οποίοι να αναπτύξουν interpreters για γλώσσες προγραμματισμού όπως η ruby, python, php, perl για android όπως το sl4a ([10]) ή να προσαρμόσουν ήδη γνωστές όπως ο rhino ([11]).

4.2 Chroot

Μία λύση που έχει βρει γόνιμο έδαφος για ανάπτυξη εφαρμογών και σε κοινότητες που ασχολούνται με ασφάλεια είναι εφαρμογές που επιτρέπουν τη δημιουργία ενός περιβάλλοντος που αποκαλείται στον κόσμο του unix σαν chroot ([12]).

Με τον όρο chroot σε ένα περιβάλλον unix αναφερόμαστε σε μια λειτουργία που αλλάζει το root folder για μια διαδικασία και τα παιδιά της. Ένα πρόγραμμα που τρέχει σε ένα τέτοιο τροποποιημένο περιβάλλον δεν μπορεί να ονοματίσει (και κατά συνέπεια να έχει πρόσβαση σε) αρχεία έξω από το προκαθορισμένο δέντρο ονομάτων. Ο όρος chroot αναφέρεται τόσο στη κλήση συστήματος όσο και στο ανάλογο wrapper πρόγραμμα. Το τροποποιημένο περιβάλλον αποκαλείται “chroot jail”.

Σαν κλήση συστήματος χρησιμοποιήθηκε για πρώτη φορά με την έκδοση 7 του Unix το 1979 και χρησιμοποιήθηκε το 1982 στις εκδόσεις του BSD. Αυτές οι πρώτες εκδόσεις χρησιμοποιήθηκαν για δοκιμές εγκαταστάσεων και σαν honeypot σε περιπτώσεις παραβιάσεων. Με ανάλογες λειτουργίες έχει χρησιμοποιηθεί και σε πιο πρόσφατες εκδόσεις Unix όπως στο FreeBSD, σε Solaris (με το όνομα Containers), ενώ υπάρχει και σε μεγάλες διανομές Linux όπως η Debian, η Ubuntu και η Fedora.

Σε ένα τέτοιο περιβάλλον είναι δυνατή η δημιουργία ενός αυτόνομου εικονικού περιβάλλοντος το οποίο μπορεί να χρησιμοποιηθεί για:

1. Ανάπτυξη και έλεγχο εφαρμογών. Ειδικά για εφαρμογές που είναι επικίνδυνο να τρέξουν σε συστήματα παραγωγής.
2. Στρώμα συμβατότητας. Για εφαρμογές με εξαρτήσεις από βιβλιοθήκες και εργαλεία που δεν είναι εύκολο να υπάρχουν μέσα σε ένα σύστημα παραγωγής.

3. Διαχωρισμό ευθυνών και ασφάλεια. Ότι ζημιές συμβούν μέσα σε ένα τέτοιο σύστημα δεν επεκτείνονται προς τα έξω χωρίς να καταβληθεί σημαντική προσπάθεια (αν και έχει αποδειχτεί πως δεν είναι αδύνατο).

Τα τελευταία χρόνια έχουν παρουσιαστεί τέτοιες εφαρμογές για περιβάλλον Android (όπως τα [13], [14]) και οι οποίες προσφέρουν τέτοιου είδους λειτουργικότητα, επιτρέποντας την λειτουργία προγραμμάτων από περιβάλλον linux σε Android όπως και κατά περίπτωση όπου είναι δυνατό την ανάπτυξη εφαρμογών κατευθείαν επάνω στην ίδια την συσκευή με την βοήθεια του GCC.

Ο κώδικας της εργασίας

V

Για την παρούσα εργασία χρησιμοποιήθηκαν αρκετά από τα προαναφερθέντα εργαλεία με τελικό σκοπό την ανάπτυξη ενός απλού web sever σε node.js.

5.1 Το σκεπτικό

Ιστορικά η javascript είχε και έχει κυρίαρχο ρόλο σαν γλώσσα προγραμματισμού στην πλευρά του χρήστη. Εν τούτοις είχε βρει γόνιμο έδαφος και σαν γλώσσα προγραμματισμού από την πρώτη στιγμή **και** για εφαρμογές σε επίπεδο server ([15],[16],[17]).

Το node.js αποτελεί την πιο πρόσφατη τέτοια προσπάθεια χρήσης της javascript σαν μια γλώσσα προγραμματισμού και περιβάλλον ανάπτυξης εφαρμογών σε περιβάλλον server.

Τα κίνητρα πίσω από τη χρήση ενός τέτοιου περιβάλλοντος για την ανάπτυξη εφαρμογών σε περιβάλλον node.js περιλαμβάνουν και δεν περιορίζονται στα εξής:

1. Οι εφαρμογές σε node.js γράφονται σε javascript και εκτελούνται με τη βοήθεια του Google V8, ενός μεταγλωττιστή με εξαιρετική απόδοση εκτέλεσης
2. Η event-driven αρχιτεκτονική του και το non-blocking I/O API παρέχουν ένα περιβάλλον το οποίο να παρέχει ένα σταθερό και γρήγορο περιβάλλον το οποίο να επιτρέπει την ανάπτυξη εφαρμογών πραγματικού χρόνου.
3. Ένα ενσωματωμένο σύνολο βιβλιοθηκών το οποίο παρέχει υποστήριξη για λειτουργίες που δεν θεωρούνται απλές, όπως χρήση DOM, βιβλιοθήκες networking (κάτι που επιτρέπει σε μια εφαρμογή να συμπεριφέρεται σαν ένας αυτόνομος web server χωρίς την υποστήριξη εργαλείων όπως ο IIS server ή ο Apache).
4. Μια τάση που παρουσιάζεται συχνά τα τελευταία χρόνια βρίσκει αρκετές γλώσσες προγραμματισμού να παράγουν αντί για κάποιο εκτελέσιμο πρόγραμμα κώδικα C ή javascript. Η κύρια αιτία για αυτή την τάση είναι η

δυνατότητα ενσωμάτωσης κώδικα από προϋπάρχουσες εφαρμογές σε νέα μεγαλύτερα σύνολα μετά το πέρας του κύκλου ζωής της αρχικής εφαρμογής. Έτσι, εν προκειμένω είναι εφικτό να ανακυκλώσει κάποιος κώδικα C/C++, ruby, python ([18],[19],[20]) .

5.2 Παραπλήσιο έργο

Η ανάπτυξη προγραμμάτων με τα οποία να προσφέρει ένας υπολογιστής κάποια απομακρυσμένη υπηρεσία αποτελεί μια από τις παλιότερες δραστηριότητες για τις οποίες χρησιμοποιήθηκαν υπολογιστές κάθε κλίμακας.

Η ανάπτυξη του HTTP ιστορικά μπορεί να εντοπιστεί από το 1989 και το έργο του Tim Berners-Lee και τις πρώτες εκδόσεις του HTTP με τη μορφή του πρωτοκόλλου όπως το γνωρίζουμε σήμερα μέσα στο 1995 [21].

Από το 1995 και μετά η μελέτη και ανάπτυξη εφαρμογών που χρησιμοποιούν το HTTP για να παρέχουν τις υπηρεσίες τους δεν έχει σταματήσει. Σε αυτό το διάστημα έχουν κατασκευαστεί εξυπηρετητές όπως ο Apache ([22]) και ο Nginx ([23]) οι οποίοι είναι σε θέση να στηρίξουν υπηρεσίες κάθε κλίμακας, και μικρότεροι όπως ο simple ([24]) και ο jetty ([25]) οι οποίοι κατασκευάστηκαν για ειδικές εφαρμογές και μικρό φόρτο εργασίας.

Σε αυτό το διάστημα ανάλογα με την κλίμακα υπάρχουν web server γραμμένοι πρακτικά σε κάθε γλώσσα προγραμματισμού και για κάθε λειτουργικό. Οι περισσότεροι από αυτούς που δημοσιεύονται συνήθως έχουν συγκεκριμένο στόχο, συνήθως εκπαιδευτικό ([7],[8]) ή πιο ειδικού σκοπού ([26]), για τον έλεγχο υπηρεσιών ή για την υποστήριξη μεγαλύτερων υπηρεσιών οι οποίες απαιτούν ένα πολύ συγκεκριμένο βαθμό λειτουργικότητας ([27]).

Το Android δεν αποτελεί εξαίρεση δεδομένης της παρουσίας για την πλατφόρμα του κάποιων από τους μεγαλύτερους web server οι οποίοι έχουν ήδη προσαρμοστεί για να τρέχουν σε αυτό ([3],[4]), κίνηση με την οποία μπορεί να αποκτήσει η κοινότητα των developer που τους αναπτύσσουν σε αυτές τις συσκευές μια πρώτη ένδειξη για τη συμπεριφορά ενός web server σε ενσωματωμένα συστήματα και ειδικότερα μερικοί από τους μεγαλύτερους και πιο πλήρεις σε λειτουργίες από αυτούς.

Τα κίνητρα πίσω από αυτή κίνηση κατά κανόνα βρίσκονταν στα πλεονεκτήματα που προσφέρουν τα περισσότερα tablet σε σχέση με άλλες συσκευές όπως φορητοί και netbook σαν ευέλικτες συσκευές ανάπτυξης και ελέγχου web εφαρμογών ([28], [29]). Το τελικό αποτέλεσμα αυτής της διαδρομής είναι εκτός από

τους παραπάνω web servers να υπάρχει ένα σύνολο από εργαλεία που να επιτρέπουν την ανάπτυξη εφαρμογών πάνω σε συσκευές που τρέχουν Android, interpreted γλώσσες όπως η python, η ruby, PHP, TCL, υπάρχουν υλοποιήσεις οι οποίες παρουσιάζουν ενεργή ανάπτυξη ([30],[31]) οι οποίες όμως είναι ακόμα σε πρώιμο στάδιο.

Για τις ανάγκες της πτυχιακής αναπτύχθηκε ένας απλός web server σε mode.js. Η λογική και τα οφέλη πίσω από αυτή την επιλογή πολλαπλά:

1 Η javascript είναι απλή στην εκμάθηση γλώσσα, κάτι που σημαίνει πως κάποιος μπορεί να αρχίσει άμεσα να αναπτύσσει κώδικα ο οποίος να καλύπτει τις ανάγκες των εφαρμογών του.
2 Επεκτασιμότητα: Ο developer είναι σε θέση να προσαρμόζει τις λειτουργίες των εφαρμογών του κατά περίπτωση γρηγορότερα σε σχέση με την ανάπτυξη κώδικα σε C++ ή Java γιατί δεν μεσολαβεί η διαδικασία build που απαιτούν οι παραπάνω 2 γλώσσες. Απλά ενημερώνει τα αρχεία του κώδικα του και φορτώνει τον interpreter με τα νέα αρχεία.
3 Οι πρόσφατες υλοποιήσεις της συγκεκριμένης γλώσσας έχουν βελτιστοποιηθεί αρκετά για να απαιτούν χαμηλότερες απαιτήσεις για την αποδοτική εκτέλεση κώδικα.
4 Βιβλιοθήκες ανοιχτού κώδικα. Υπάρχει ένα μεγάλο σώμα βιβλιοθηκών οι οποίες καλύπτουν από βασικές δουλειές όπως επεξεργασία κειμένου, web scrappers, στήσιμο web services (REST) έως πολυπλοκότερες όπως υποστήριξη ολόκληρων πρωτοκόλλων όπως το bittorent ή το IRC και τα οποία επιτρέπουν την δημιουργία σύνθετων εφαρμογών οι οποίες να τα υποστηρίζουν.

Ένας σημαντικός παράγοντας που μπορεί να αποθαρρύνει την ενασχόληση με την ανάπτυξη εφαρμογών με node.js σε android είναι το ευρύ φάσμα εργαλείων που απαιτεί. Το node.js έχει προσαρμοστεί για Android με διαφορετικές μεθοδολογίες σε διαφορετικές χρονικές στιγμές ([32],[33],[34]), αρκετές από τις οποίες μπορεί και να μην είναι πλέον χρήσιμες λόγω αλλαγών που έχουν υποστεί τα εργαλεία με τα οποία η εκάστοτε ομάδα έκανε το port, έλλειψη ενδιαφέροντος ή σε περιπτώσεις όπως το [33], η λειτουργία του είναι δυνατή αλλά στο παρόν στάδιο το συγκεκριμένο περιβάλλον χαρακτηρίζεται ακόμα pre-alpha, με ότι αυτό συνεπάγεται.

Κατά συνέπεια μέρος της παρούσας πτυχιακής δεν αποτελεί μόνο η υλοποίηση του web server, αλλά και του ίδιου του node.js για το android. Το τελικό αποτέλεσμα στο οποίο κατέληξα είναι πως ο κώδικας του node.js είναι εφικτό να προσαρμοστεί για συσκευές android χωρίς αυτό να αποτελεί μια οριστική λύση ή πως δεν είναι δυνατό να υπάρξουν προβλήματα ή παράπλευρες παρενέργειες που δεν είναι άμεσα εμφανείς. Εν τούτοις το περιβάλλον που δημιουργήθηκε για τις ανάγκες του προγράμματος της διπλωματικής δεν παρουσίασε κάποιο πρόβλημα που να απαιτούσε άμεση παρέμβαση στον κώδικα του ίδιου του node.js.

Η προσέγγιση

VI

6.1 Λεπτομέρειες αρχιτεκτονικής

Όπως αναφέρθηκε στην αρχή της πτυχιακής ο στόχος της πτυχιακής είναι τελικά η δημιουργία ενός web server για Android. Πιο συγκεκριμένα, ένα πρόγραμμα το οποίο να είναι δυνατό να διαμοιράζει αρχεία και να εξυπηρετεί πολλαπλούς χρήστες ταυτόχρονα.

Αυτό προϋποθέτει μια σειρά από μικρότερα κομμάτια τα οποία να είναι σε θέση να:

- Λαμβάνουν HTTP requests από το δίκτυο.
- Να καταλαβαίνουν τι είναι αυτό που ζητάνε οι clients από τον server
- Να παρέχουν το αρχείο / σελίδα στον χρήστη ή να του παρέχουν κάποιο σφάλμα από το οποίο επεξηγεί τι ακριβώς πήγε λάθος.

Προαιρετικά κομμάτια τα οποία μπορούν να προσφέρουν λειτουργικότητα και αξία σε μια τέτοια εφαρμογή είναι ένας μηχανισμός event log και τήρησης στατιστικών και ένας μηχανισμός εξυπηρέτησης δυναμικού περιεχόμενου (web services) .

6.2 Υποθέσεις και παραδοχές.

Κατά τη φάση του σχεδιασμού υιοθετήθηκαν μερικές βασικές υποθέσεις για το πως θα πρέπει να σχεδιαστεί και να λειτουργεί ο server για λόγους απλότητας και ασφάλειας:

1 Convention over configuration.

2 Ένας υπολογιστής που κάνει χρήση του tcp/ip μπορεί να δέχεται ανοιχτά όσα δεδομένα έρχονται από το υπόλοιπο δίκτυο και να απαντάει επιλεκτικά σε όσα δεδομένα στέλνει στο υπόλοιπο δίκτυο. Η φιλοσοφία πίσω από μια τέτοια λειτουργία εξασφαλίζει την ασφαλή και απρόσκλητη λειτουργία ενός υπολογιστή αφού εξοικονομεί πόρους και κυριότερα, να υλοποιεί μια πολιτική ασφάλειας που να ταιριάζει στα δεδομένα που ανταλλάσσει.

3 Δυνατότητα εκκίνησης node.js scripts σαν ξεχωριστές διαδικασίες κατά την εκκίνηση του server.

Η πρώτη παραδοχή γίνεται για λόγους απλότητας, ταχύτητας και ασφάλειας:

- Ο σχεδιασμός απλοποιείται εάν η συμπεριφορά του server όπως και το που αποθηκεύονται τα στατιστικά χρήσης, τα πιθανά προβλήματα και οι ρυθμίσεις του γίνονται με τη βοήθεια των ανάλογων αρχείων txt αρχεία με json encoded δεδομένα.
- Ο server οφείλει να έχει ένα και μόνο ένα μέρος από το οποίο παρέχει στους πελάτες του αρχεία. Η αντίθετη αντιμετώπιση αποτελεί ένα σύνηθες πρόβλημα που εμφανίζεται σε απλούς web servers και μεγαλύτερους με προβληματικές ρυθμίσεις και εγγενή προβλήματα που αποκαλείται tree traversal.
- Η δημιουργία λίστας με επιτρεπτά είδη αρχείων τα οποία να μπορεί να στείλει ο server είναι επιθυμητό για την αποφυγή αρχείων τα οποία ο διαχειριστής κρίνει σκόπιμο να μην μοιράζεται.
- Τέλος η ύπαρξη μηχανισμού whitelists κρίνεται σκόπιμο και επιθυμητό να υπάρχει για συσκευές οι οποίες δεν μπορούν να διαχειριστούν μεγάλο αριθμό συνδέσεων ή ο διαχειριστής κρίνει ότι θέλει να υπάρχει πρόσβαση σε περιορισμένο αριθμό χρηστών.

Η δεύτερη παραδοχή ονομάζεται αρχή του Postel (Postel's principle) και αποτελεί μια από τις σχεδιαστικές αρχές του tcp/ip Το HTTP επιτρέπει την ανταλλαγή μηνυμάτων λάθους εν τούτοις κρίθηκε σκόπιμο για την συγκεκριμένη υλοποίηση να μην αποστέλλονται μηνύματα λάθους στον κάθε client γιατί κατά τη διάρκεια των δοκιμών παρατηρήθηκε αισθητά έντονη λειτουργία του μηχανήματος στο οποίο έτρεχε ο κώδικα (με ότι αυτό μπορεί να συνεπάγεται) οπότε μια ομοιόμορφη αντιμετώπιση απλών σφαλμάτων κατά τη πληκτρολόγηση και denial of service attacks κρίθηκε σκόπιμο να αντιμετωπιστούν με την ίδια βαρύτητα.

Η εκκίνηση node.js scripts σε ξεχωριστές διαδικασίες αποτελεί ένα μηχανισμό με τον οποίο μπορούμε να επεκτείνουμε τη λειτουργικότητα του server και να παρέχουμε υπηρεσίες και λειτουργίες που δεν υπήρχαν στο πρωτότυπο πρόγραμμα. Η κύριες αιτίες για τις οποίες χρησιμοποιήθηκε αυτός ο μηχανισμός είναι η υλοποίηση ενός web interface με το οποίο να τροποποιούνται οι ρυθμίσεις του server και ενός REST API για αυτή τη δουλειά.

Με αυτό το μοντέλο λειτουργίας ο server επιβαρύνεται με την εκτέλεση των scripts κατά την εκκίνηση του μόνο ενώ τα συγκεκριμένα script είναι το καθένα υπεύθυνο για τη λειτουργία και την ασφάλεια του και μόνο και κατά συνέπεια δεν προκαλούν προβλήματα στην λειτουργία του υπόλοιπου server.

Λεπτομέρειες υλοποίησης

VII

7.1 Server side

Ο κώδικας του server διαμορφώθηκε από 15 κομμάτια. Περιληπτικά αποτελείται από τα εξής κομμάτια:

- Το κυρίως σώμα του server, το οποίο αναλαμβάνει την αρχικοποίηση των ρυθμίσεων, την λήψη και επεξεργασία των αιτημάτων από τους πελάτες και την διαχείριση των στατιστικών και των σφαλμάτων.
- Μια μέθοδο assert για παρακολούθηση της ροής του προγράμματος
- Μια μέθοδο ελέγχου whitelisted ip
- Δυο μέθοδοι καταγραφής στοιχείων των clients από τα στοιχεία των HTTP requests.
- Μια μέθοδος καταγραφής προβλημάτων.
- Μια μέθοδο ελέγχου επιτρεπτών URL
- Μια μέθοδο ύπαρξης αρχείων
- Μια μέθοδο τήρησης βασικών στατιστικών
- Μια μέθοδο διαχείρισης αρχείων τα οποία ο χρήστης θα βλέπει στην οθόνη του (εικόνες, html αρχεία)
- Μια μέθοδο διαχείρισης αρχείων τα οποία ο χρήστης θα προσπαθήσει να κατεβάσει.
- Μια μέθοδο ενημέρωσης βασικών mime type τα οποία ενδιαφερόμαστε να υποστηρίζει ο server
- Δυο μέθοδοι αδειοδότησης αποστολής επιτρεπτών τύπων αρχείων

Όλα τα δεδομένα που χρειάζεται ο server κατά την εκκίνηση του, τα στατιστικά και τα σφάλματα του βρίσκονται αποθηκεύονται σε μια σειρά αρχείων txt στον βασικό φάκελο του server. Τα δεδομένα σε κάθε ένα από αυτά τα αρχεία αποθηκεύονται σαν JSON objects.

Πιο συγκεκριμένα:

- Οι ρυθμίσεις αποθηκεύονται στο αρχείο settings.txt
- Τα στατιστικά στο αρχείο statistics.txt
- Η whitelist στο αρχείο whitelist.txt
- Τα σφάλματα στο exceptions.txt

Ένας περιληπτικός πίνακας των μεθόδων από τις οποίες απαρτίζεται ο server καθώς και τα αρχεία ρυθμίσεων όπως και τα logfiles από τα οποία εξαρτάται μπορεί να σχηματιστεί με μία συνοπτική περιγραφή των λειτουργιών κάθε μιας. Για αναλυτικότερη περιγραφή των λειτουργιών ο αναγνώστης παραπέμπεται στα ανάλογα αρχεία κώδικα τα οποία περιέχουν αναλυτικότερα σχόλια και παρατηρήσεις για κάθε μια από αυτές ξεχωριστά.

Μέθοδοι server	
SERVER_ASSERT	μέθοδος assert
IP_WHITELIST_OK	Έλεγχος αποδεκτών IP από αρχείο αποδεκτών IP
IP_LOG	Μέθοδος καταγραφής στοιχείων και προτιμήσεων χρήστη **με βάση τα στοιχεία που παρέχει το http request και μόνο**
EXCEPTION_LOG_INSERT	Ρουτίνα αποθήκευσης εξαιρέσεων.
CHECK_CONTENT_EXCEPTIONS	Συνάρτηση έλεγχου επιτρεπτών URL.
FILE_CHECK	Επιστρέφει θετική/αρνητική απάντηση για την ύπαρξη ενός αρχείου.
HANDLE_SERVER_STATISTICS	Μέθοδος τήρησης στατιστικών.
HANDLE_STATIC_CONTENT	Μέθοδος διαχείρισης στοιχείων τα οποία βλέπει ο χρήστης στην οθόνη του
HANDLE_STATIC_FILES	Μέθοδος διαχείρισης στοιχείων τα οποία κατεβάζει ο χρήστης
GET_MIME_TYPE	Επιστρέφει το Mime type ενός τύπου αρχείου.
BROWSER_CONTENTS_TYPE_OK FILE_CONTENTS_OK	-Επιστρέφουν το αν είναι σωστό ο server να δώσει πρόσβαση σε αυτό το τύπο αρχείων.
Αρχεία ρυθμίσεων	
settings.txt	Ρυθμίσεις server:

	-Η θύρα από την οποία ακούει για αιτήματα -Είδη αρχείων που επιτρέπεται να προσφέρει στον χρήστη -Mime types αρχείων -Ονόματα αρχείων που δεν επιτρέπεται να παρουσιάσει σε client.
statistics.txt	Στατιστικά χρήσης: -Μετρητές χρήσης κάθε στατικού αρχείου.
whitelist.txt	Επιτρεπτές IP
Exceptions.Log.txt	Ενημερώσεις σφαλμάτων

7.2 Administration

Η συγκεκριμένη υπηρεσία έχει διπλό σκοπό:

Πρωταρχικός στόχος ήταν η δημιουργία μιας βασικής υπηρεσίας σε node.js, άμεσα συνδεδεμένη με την λειτουργία του web server η οποία να χρησιμέψει σαν απλό πρότυπο για άλλες υπηρεσίες οι οποίες δυνητικά να μπορούν να λειτουργήσουν συμπληρωματικά στις ήδη υπάρχουσες λειτουργίες.

Δημιουργήθηκε για να δώσει πρόσβαση σε επιμέρους κομμάτια του server είτε μέσω ενός απλού HTML interface είτε μέσω κλήσεων REST μέσω του οποίου να μπορεί ένας διαχειριστής να τροποποιήσει τις ρυθμίσεις του server και να αντλήσει πληροφορίες για σφάλματα και στατιστικά λειτουργίας του server.

Μια εναλλακτική σε αυτό το κομμάτι της πτυχιακής θα ήταν η δημιουργία μιας εφαρμογής σε java από την οποία να ασκούνται τα διαχειριστικά καθήκοντα, όμως η έλλειψη **απομακρυσμένης** διαχείρισης την οποία προσφέρει η παρούσα λύση θα την καθιστούσε πάντα υποδεέστερη. Επίσης η δυνατότητα σύνδεσης της με ένα HTML interface την καθιστά ανεξάρτητη πλατφόρμας, κάτι που συνεπάγεται την διαχείριση του server από οποιοδήποτε συσκευή έχει browser.

Για λόγους απλότητας και ταχύτητας στην εκτέλεση:

- Η γλώσσα στην οποία είναι γραμμένες θεωρείται πως είναι javascript.
- Γίνονται fork κατά την εκκίνηση του server
- Βρίσκονται όλες συγκεντρωμένες κάθε μία στον δικό της φάκελο μέσα στον φάκελο SERVICES και κάθε μία έχει ένα αρχείο με όνομα server.js στον οποίο βρίσκεται ο βασικός κώδικας του server.

Το τελικό αποτέλεσμα αυτών των παραδοχών είναι ένα σύνολο από διαδικασίες οι οποίες:

- Δεν επιβαρύνουν περισσότερο από όσο χρειάζεται την λειτουργία του server
- Κάθε service αποτελεί μια ξεχωριστή διαδικασία με το δικό της χώρο λειτουργίας και συνθήκες κατά τις οποίες λειτουργεί και τερματίζει. Εφόσον όμως υπάρχει ενδεχόμενο κάποια από αυτές να μην συμπεριφέρεται όπως προβλέπεται μπορεί να τερματιστεί χωρίς να επηρεάζεται υποχρεωτικά η λειτουργία του υπόλοιπου συστήματος.

Το τελικό πρόγραμμα είναι δυνατόν μέσω ενός REST api να επηρεάσει τις ρυθμίσεις του server που αφορούν:

- Τη θύρα από την οποία εξυπηρετεί τα αιτήματα
- τους τύπους αρχείων στους οποίους επιτρέπει την πρόσβαση
- Τα εκάστοτε mime types των τύπων αρχείων που εξυπηρετεί

Επίσης δίνει πρόσβαση σε πληροφορίες γύρω από τον server και τις λειτουργίες του:

- Λίστα των services τα οποία αφορούν τον server και τα οποία έχει αρχίσει
- Λίστα των mime types τα οποία υποστηρίζει ο server
- Μπορεί να προσφέρει αντίγραφο των ρυθμίσεων του server, αντίγραφο των σφαλμάτων που διαχειρίστηκε κατά την εκτέλεση του ο server καθώς και των στατιστικών χρήσης του server.

Συνοπτικά τα παραπάνω σχηματίζουν το εξής REST API:

Path	**Χρήση**	**παράδειγμα**
settings/set/port/port#	Ρύθμιση θύρα από την οποία θα ακούει για δεδομένα ο server	http://localhost:8081/settings/set/port/8080
/settings/add/browser/:TYPE	Ορισμός ενός τύπου δεδομένων τα οποία θα παρουσιάζονται απευθείας στον χρήση	http://localhost:8081/settings/add/browser/pdf
/settings/add/mime/:NAME/:TYPE	Ορισμός του mime type ενός τύπου αρχείου με μεγαλύτερη ακρίβεια	http://localhost:8081/settings/add/dd/mime/js/"text/javascript"

/settings/service/names	Επιστρέφει μια λίστα path των service που βρήκε ο server και ξεκίνησε	http://localhost:8081/settings/service/names
/settings/mime/types	επιστρέφει μια λίστα path των mime types που υποστηρίζει ο server	http://localhost:8081/settings/mime/types
/getServerInfo/:TYPE	Επιστρέφει τις πληροφορίες που σχετίζονται με τις ρυθμίσεις, τα λάθος url που ζητάνε οι χρήστες αλλά και τα στατιστικά των αιτημάτων που κάνουν οι χρήστες.	http://localhost:8081/getServerInfo/settings http://localhost:8081/getServerInfo/exceptions http://localhost:8081/getServerInfo/statistics

7.3 Node.js build

Ένα τελευταίο κομμάτι που πρέπει να δοθούν κάποιες εξηγήσεις είναι το build του ίδιου του node.js. Τα τελευταία 4 χρόνια έχουν υπάρξει προσπάθειες προσαρμογής του σε πλήρη εφαρμογή Android σε java και ελάχιστα native κομμάτια οι οποίες όμως είτε σταμάτησαν είτε σε πλήρη java τα οποία όμως δεν έχουν παράγει ακόμα σταθερά συστήματα προσαρμογής (αν και υπάρχουν προσπάθειες που δείχνει ενθαρρυντικά σημάδια).

Ο βασικός τρόπος με τον οποίο χρησιμοποιούμε το node.js σε Android είναι σαν native εκτελέσιμο. Αυτό δυσχεραίνει τα πράγματα σε επίπεδο υποστήριξης αφού η Google έχει δώσει μεγαλύτερη βαρύτητα στην κατασκευή εφαρμογών σε java ενώ για λόγους ασφάλειας υπάρχουν αυστηρότεροι περιορισμοί στην λειτουργία τους. Τέλος λόγω του μεγάλου εύρους συσκευών που τρέχουν Android υπάρχει σοβαρό ρίσκο να συναντήσουμε συσκευές στις οποίες το περιβάλλον μας να μην τρέχει καθόλου ή ακόμα χειρότερα να τρέχει λάθος και να μην παρουσιάζει μηνύματα λάθους. Παρόλα αυτά όμως ένα μεγάλο σώμα γνώσης που αφορά interpreters υπάρχει σε C και κατά συνέπεια αυτή η τάση δεν πρόκειται να σταματήσει σύντομα.

Παρά τα προβλήματα που παρουσιάζει μια τέτοια προσέγγιση τα τελευταία χρόνια εξαιτίας της αύξησης συσκευών με ARM επεξεργαστές (κινητά, tablet, συσκευές Rasberry pi) υπάρχει μια κοινότητα ανθρώπων οι οποίοι έχουν αναλύσει ένα μεγάλο μέρος από αυτά τα προβλήματα και έχουν συμβάλει στην απλοποίηση της διαδικασία μεταφοράς του node.js σε ενσωματωμένα περιβάλλοντα κάτι που σημαίνει πως τουλάχιστον από την έκδοση 0.10 του node.js μπορεί να

αντιμετωπιστεί σαν μια τυπική διαδικασία build από τον πηγαίο κώδικα, κάτι που παρουσιάζεται αναλυτικότερα στο [35].

Οι 2 πιο επικρατέστερες μέθοδοι που έχουν επικρατήσει για την εκτέλεση του σε περιβάλλον android περιλαμβάνουν το compile του node.js είτε απευθείας με τη βοήθεια του GCC σε περιβάλλον chroot είτε το build του με τη βοήθεια του NDK.

Η διαδικασία μέσω της οποίας κάνουμε build τον κώδικα μέσω του NDK είναι η απλούστερη από τις 2 αφού ακολουθεί μια διαδικασία η οποία σε περιβάλλον Linux δεν απέχει από τη τυπική διαδικασία με την οποία μπορεί ένας χρήστης linux να κάνει build ένα πρόγραμμα από τον κώδικα του (μια αναλυτική περιγραφή της διαδικασίας και η οποία χρησιμοποιήθηκε για τις ανάγκες της παρούσας εργασίας βρίσκεται στο [35]).

Περιληπτικά, για το build του node.js με τη βοήθεια του NDK η διαδικασία αποτελείται από:

- Εγκατάσταση του NDK,
- Download της πιο πρόσφατης έκδοσης
- Ρύθμιση των compile script. Οι περισσότερες από τις επιλογές γίνονται γιατί το build το οποίο κάνουμε κατά κανόνα γίνεται για επεξεργαστές ARM, κάτι που σημαίνει ότι κομμάτια που προσδιορίζουν κάτι αντίθετο πρέπει να αλλάξουν
- configure της διαδικασίας με το αντίστοιχο script
- Ρύθμιση της python που απαιτείται για το build (αποτελεί γνωστό θέμα του NDK και η ρύθμιση της απαιτείται για το build)
- Εκτελούμε το make script

Η εκτέλεση του binary μπορεί να γίνει με την βοήθεια του ADB είτε χειροκίνητα στη τοποθεσία /data/local/tmp/Release στην συσκευή. Ο συγκεκριμένος φάκελος αποτελεί ένα από τα λίγα σημεία στο filesystem τα οποία επιτρέπουν την ύπαρξη εκτελέσιμων αρχείων αλλά και την αλλαγή των προνομίων αρχείων σε εκτελέσιμα κάτι που σημαίνει πως μετά την μεταφορά του στον συγκεκριμένο φάκελο γίνεται εφικτή η εκτέλεση του κώδικα του server.

7.4 Εγκατάσταση chroot

Η διαδικασία εγκατάστασης και λειτουργίας του node.js σε περιβάλλον chroot σχετίζεται άμεσα με τα προνόμια τα οποία έχουμε σαν χρήστες επάνω στην συσκευή όσο και στις ρυθμίσεις και το hardware πάνω από το οποίο τρέχει. Τα μέχρι τώρα πειράματα σε έχουν δείξει πως εφόσον το περιβάλλον chroot έχει εγκατασταθεί επιτυχώς είναι εφικτή και η εγκατάσταση του node.js όσο και η διαδικασία build **χωρίς την ανάγκη εξωτερικής υποστήριξης για τη διαδικασία του compile από κάποιο ξένο μηχάνημα απευθείας πάνω στην ίδια τη συσκευή.**

Η διαδικασία δημιουργίας, εγκατάστασης και λειτουργίας ενός περιβάλλοντος chroot είναι κάτι που συνήθως συνδέεται με το είδος της εφαρμογής για την οποία δημιουργούμε ένα τέτοιο περιβάλλον και κατά κανόνα γίνεται χειροκίνητα με σκοπό την ρύθμιση του μεγαλύτερου μέρους των επιμέρους κομματιών τα οποία θα απαρτίζουν το νέο περιβάλλον. Τα τελευταία χρόνια οι μεγαλύτερες διανομές όπως η Fedora και η Debian έχουν δημιουργήσει εργαλεία και οδηγούς με τα οποία οι κοινότητες έχουν απλοποιήσει την διαδικασία ενώ σε Android υπάρχουν αντίστοιχα εργαλεία.

Δημιουργία chroot περιβάλλοντος η οποία να μπορεί να χρησιμοποιηθεί σε rooted κινητά χωρίς εξωτερικά εργαλεία όπως το LinuxDeploy ή το GNUroot είναι εφικτή ([39]). Στις περισσότερες περιπτώσεις η διαδικασία περιλαμβάνει:

- Τη δημιουργία μιας loop συσκευής που θα αποτελέσει τη βάση του νέου συστήματος
- Μορφοποίηση του filesystem της νέας συσκευής ([40],[41])
- Αρχικοποίηση του συστήματος με τη βοήθεια εργαλείων όπως το debootstrap το οποίο χρησιμοποιεί η διανομή Debian
- Μεταφορά του νέου filesystem στη συσκευή
- Εκκίνηση του νέου συστήματος με τη χρήση της εντολής chroot (μια συνηθισμένη μέθοδος χρήσης της είναι μέσω του πακέτου busybox)
- Ρύθμιση του συστήματος
- Χρήση εργαλείων και εφαρμογών

Και στις 2 περιπτώσεις δεδομένου ότι παράγουμε native αξίζει να σημειωθεί η σημασία που έχει η βαρύτητα που έχει η σωστή ρύθμιση του makefile και του NDK αφού είναι δυνατόν να παράγουμε εκτελέσιμα τα οποία δεν εκτελούνται γιατί δεν έγιναν compile για σωστό επεξεργαστή ή ακόμα και γιατί ο επεξεργαστής στον οποίο θέλουν να δουλέψουν δεν υποστηρίζει λειτουργίες τις οποίες έχει ανάγκη το node.js όπως για παράδειγμα το ενδεχόμενο ο επεξεργαστής να μην έχει λειτουργίες blx, ένα πρόβλημα που έχει παρουσιαστεί στο παρελθόν και το οποίο απαιτεί τις ανάλογες ρυθμίσεις ([37])

Πειραματικά αποτελέσματα

VIII

8.1 Συνθήκες πειραμάτων

Η πτυχιακή μέχρι τώρα περιγράφει περιληπτικά ένα σύνολο εργαλείων, διαδικασιών και αναφορών με τα οποία μπορούμε να δημιουργήσουμε μια λειτουργική έκδοση του node.js μέσω του οποίου μπορούμε να δημιουργήσουμε ένα web server ο οποίος να μπορεί να παρέχει υπηρεσίες και περιεχόμενο στους χρήστες του.

Ένα σημαντικό κομμάτι μαζί με το **πως** μπορεί να επιτευχθεί ο συγκεκριμένος στόχος είναι το πόσο αποτελεσματική είναι η συγκεκριμένη λύση. Για να αξιολογηθεί η συγκεκριμένη προσέγγιση έγινε εγκατάσταση του node.js και του κώδικα που γράφτηκε σε 3 συστήματα, ένα tablet, ένα smartphone και σε ένα εικονικό pc το οποίο έτρεχε σε virtualbox την έκδοση 4.4.2 του Android x86, ένα port του Android για PC.

Οι συγκεκριμένες επιλογές έγιναν γιατί καλύπτουν ένα μεγάλο φάσμα **πραγματικών** συσκευών με λειτουργικό Android οι οποίες μπορούν να αναλάβουν τον ρόλο ενός web server. Οι συγκεκριμένες ανήκουν σε ένα φάσμα προσβάσιμων συσκευών οι οποίες είναι άμεσα διαθέσιμες με χαμηλό κόστος και αποτελούν ιδανικές συσκευές ελέγχου ακριβώς γιατί μπορούν να δείξουν τις συμπεριφορές που μπορεί να παρουσιαστούν σε μικρά δίκτυα όπως αυτό ενός γραφείου ή μιας επιχείρησης.

Με αυτό το σκεπτικό κρίθηκε σκόπιμο σε αυτό το περιβάλλον ενός τοπικού δικτύου και με άλλους 3 φορητούς υπολογιστές διαθέσιμους κρίθηκε σκόπιμο να δοκιμαστεί η αξιοπιστία κάθε συσκευής σε βάθος χρόνου όταν όλοι οι υπόλοιπες συσκευές προσπαθούσαν ταυτόχρονα να κατεβάσουν δεδομένα από ένα μηχάνημα. Αυτό είχε σαν αποτέλεσμα για ένα server κάθε φορά να προσπαθούσαν να εξυπηρετηθούν ταυτόχρονα άλλες 4 συσκευές για μεγάλα διαστήματα (3 υπολογιστές και 2 συσκευές android, η μία από αυτές με χρέη server). Αυτή η διαδικασία είναι άμεσα αυτοματοποιήσιμη και γινόταν εκτέλεση της μέχρι την εξάντληση της μπαταρίας κάθε μιας από τις συσκευές με Android.

Στις 2 συσκευές υπήρχε δυνατότητα εγκατάστασης του node.js τόσο σε επίπεδο binary όσο και σε περιβάλλον chroot. Στην εικονική μηχανή χρησιμοποιήθηκε επιτυχής χειροκίνητη κατασκευή και εκκίνηση ενός περιβάλλοντος chroot με βάση τον οδηγό του [26] η χρήση του όμως ήταν περιορισμένη λόγω της δυσκολίας χρήσης που παρουσίαζε λόγω του στρώματος εξομοίωσης.

8.2 Αποτελέσματα

Συνολικά οι 3 συσκευές μέσα σε αυτό το τηρουμένων των αναλογιών μικρό δίκτυο ήταν σε θέση να αντεπεξέλθουν στην ταυτόχρονη εξυπηρέτηση ενός μικρού αριθμού πελατών οι οποίοι να αιτούνται ταυτόχρονα δεδομένα για τον χρόνο ζωής που προσέφερε σε κάθε περίπτωση η μπαταρία της κάθε συσκευής.

Αυτό δίνει ενθαρρυντικά σημάδια γύρω από την ποιότητα και τις αντοχές των υποσυστημάτων από τα οποία απαρτίζονταν οι συσκευές στις οποίες δοκιμάστηκαν, την γενικότερη ποιότητα του ίδιου του λειτουργικού και του node.js.

Οι server από την φύση τους είναι μηχανήματα τα οποία είναι αναμενόμενο ότι θα προσεγγίζουν προοδευτικά το 100% της χρήσης κάθε πόρου που είναι διαθέσιμος. Αυτό συνεπάγεται πως σε παρατεταμένη χρήση οι συγκεκριμένες συσκευές θα αντιμετωπίσουν ένα σύνολο προβλημάτων που αφορούν με τα λειτουργικά όρια τα οποία αφορούν τα υποσυστήματα της συσκευής:

- Οι αποθηκευτικοί χώροι κατά βάση είναι SD cards, οι οποίες έχουν πολύ μεγάλο όμως συγκεκριμένο όριο ασφαλών εγγραφών / αναγνώσεων πέρα από το οποίο τα δεδομένα αρχίζουν να παρουσιάζουν σημεία εκφυλισμού.

- Ο επεξεργαστής επίσης είναι ένα υποσύστημα το οποίο είναι αναμενόμενο να αγγίξει το μέγιστο των δυνατοτήτων του ανάλογα με το πλήθος των αιτημάτων. Δεδομένης της αδυναμίας χρήσης αποδοτικών μηχανισμών ψύξης που υπάρχουν σε μεγαλύτερα συστήματα η πιθανότητα αστοχιών που συνεπάγονται της υπερθέρμανση του επεξεργαστή είναι μεγάλη όπως και η καταστροφή του επεξεργαστή, γεγονός καταστροφικό για ολόκληρη τη συσκευή αφού οι επεξεργαστές δεν μπορούν να αντικατασταθούν σε αυτό το είδος συσκευών.

- Οι μπαταρίες είναι σχετικά πιθανό να έχουν να αντιμετωπίσουν προβλήματα μακροπρόθεσμα. Η οθόνη και τα κυκλώματα του ασύρματου δικτύου σε πλήρη λειτουργία είναι τα πιο ενεργοβόρα υποσυστήματα σε tablet και smartphones κάτι που μεταφράζεται σε συχνότερους κύκλους φορτίσεων και ότι φθορά συνεπάγεται.

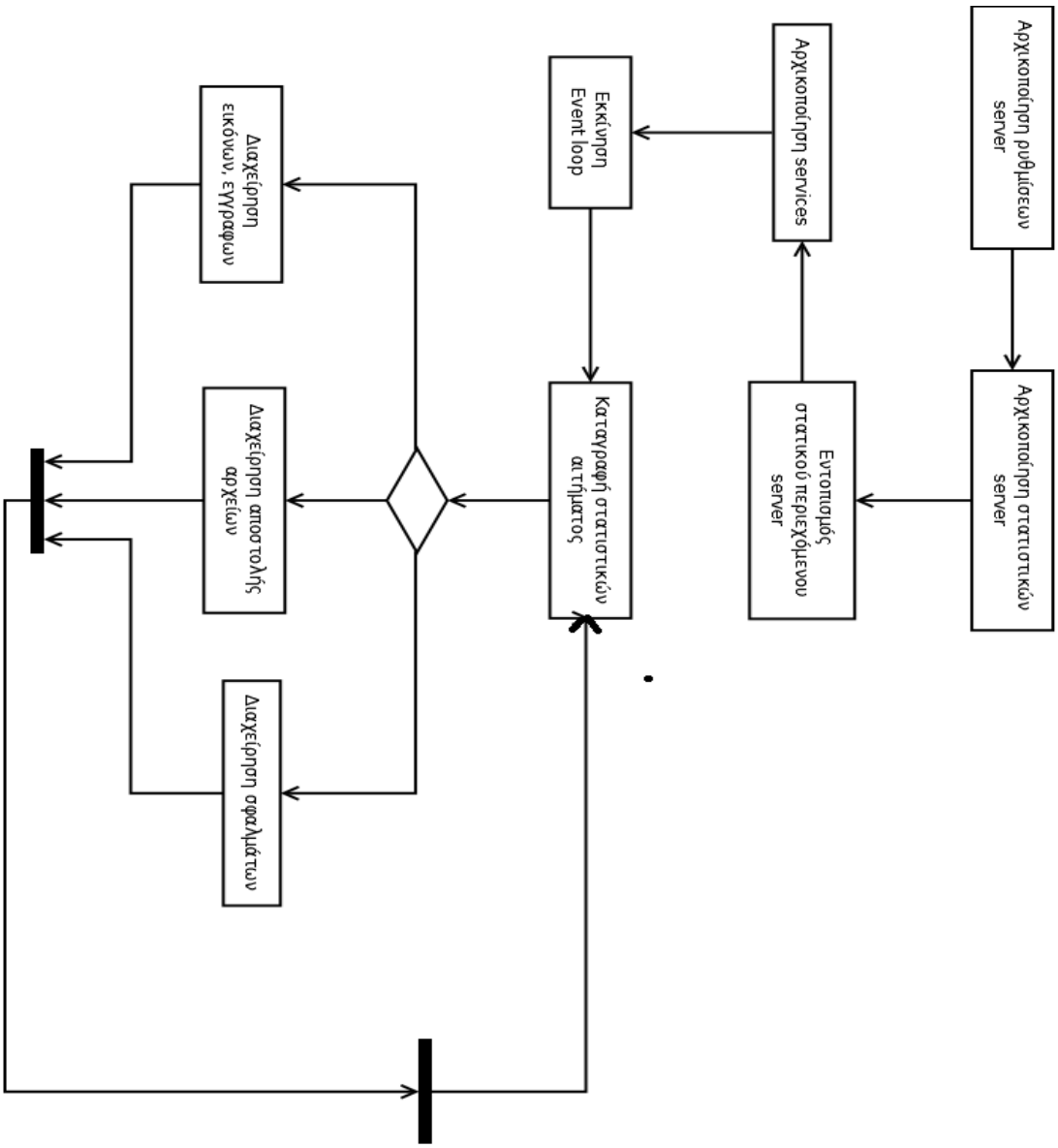
Κατά συνέπεια παρατεταμένη χρήση συσκευών με Android μπορεί να επιφέρει πάνω στη συσκευή μεγαλύτερο αριθμό αστοχιών λόγω της φύσης τους. Ενσωμάτωση μηχανισμών whitelist, ο ορισμός ενός μέγιστου ορίου αιτημάτων τα οποία να εξυπηρετεί σε κάθε δεδομένη χρονική στιγμή ο server είναι άμεσα υλοποιήσιμα για τον server της πτυχιακής (στον οποίο υλοποιήθηκε και δοκιμάστηκε ένας απλός μηχανισμός whitelist) ενώ σε βάθος χρόνου κρίνεται σκόπιμη η χρήση συνθετότερων μηχανισμών υποστήριξης απέναντι σε πιθανά κακόβουλες πλην νοσηρές χρήσεις:

- Ενσωμάτωση συστημάτων intrusion detection συστημάτων
- Ενσωμάτωση μηχανισμών παρακολούθησης του βαθμού χρήσης των υποσυστημάτων.
- Λογισμικό παρακολούθησης της κατάστασης του hardware και διάγνωσης / επισκευής σφαλμάτων σε αυτό

Επίδειξη συστήματος

IX

Συνολικά η συμπεριφορά του server προσδιορίζεται από μια συμπεριφορά όπως αυτή που παρουσιάζει το κάτωθι διάγραμμα ροής.



Για ένα σύνολο ρυθμίσεων όπως αυτή της εικόνας 1, το οποίο περιγράφει ένα server ο οποίος ακούει για μηνύματα στη θύρα 80 και ο οποίος είναι δυνατό να παρουσιάσει στον browser ενός χρήστη μια εικόνα jpg όπως η εικόνα 2. Η εικόνα 3 παρουσιάζει μια ροή μηνυμάτων της μεθόδου assert κατά τη διάρκεια εκτέλεσης του κώδικα

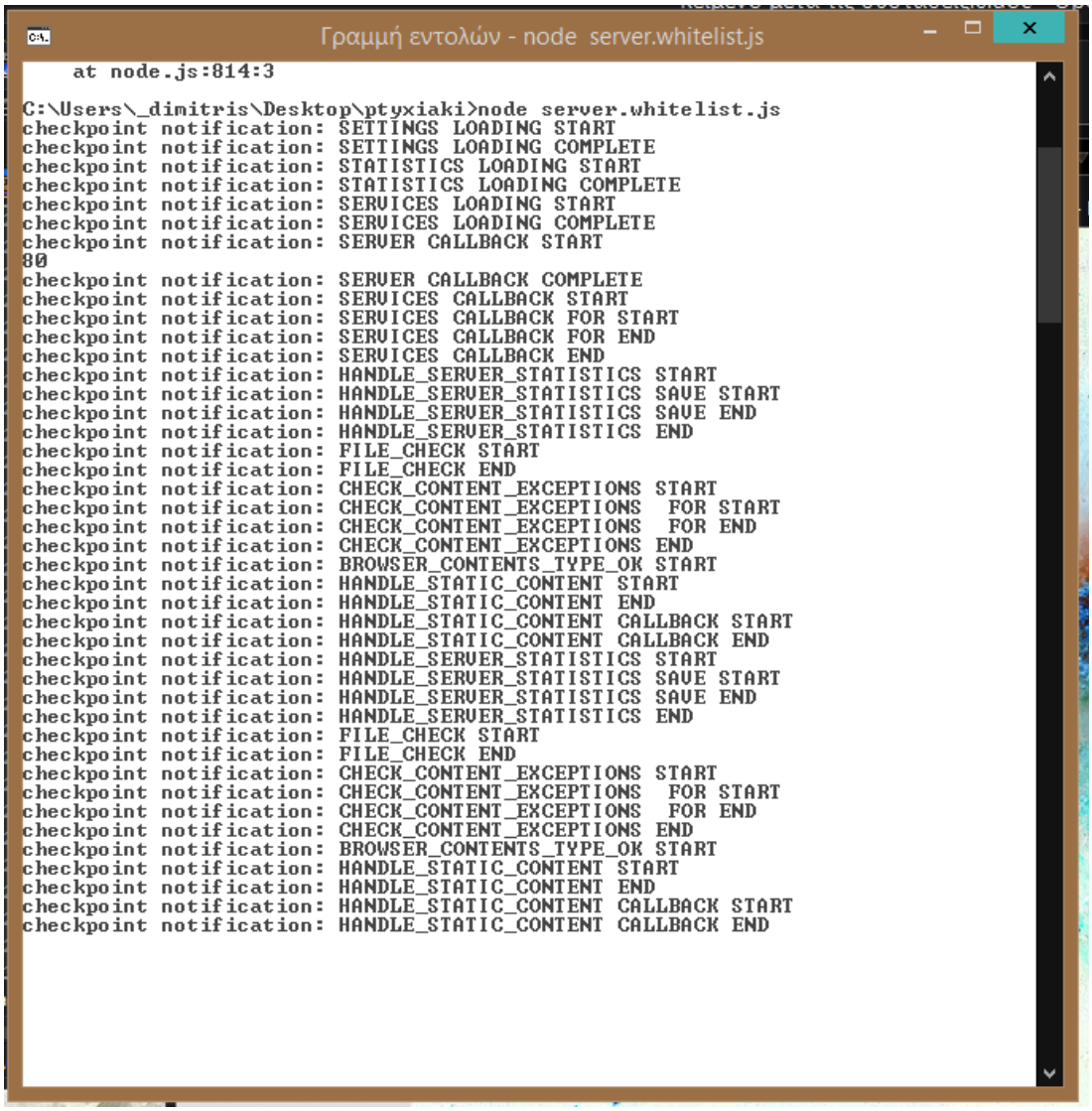
```
 "SERVER_PORT":"80",
 "BROWSER_CONTENTS":["html","xml","jpg","jpeg","gif"],
 "FILE_CONTENTS":
["pdf","txt","flv","mov","mpg","avi","mp4","mkv","zip","rar","foo","foo"],
 "MIME_TYPES"{
      "js":"text/javascript",
      "css":"text/css",
      "zip":"application/zip",
      "jpg":"image/jpeg",
      "rar":"application/rar" },

"FILE_EXCLUDE":[
 "settings.txt","settings.old.txt","statistics.txt","exceptions.log.txt","server.js"
]}
```

Εικόνα 1: Ένα τυπικό σύνολο ρυθμίσεων του server



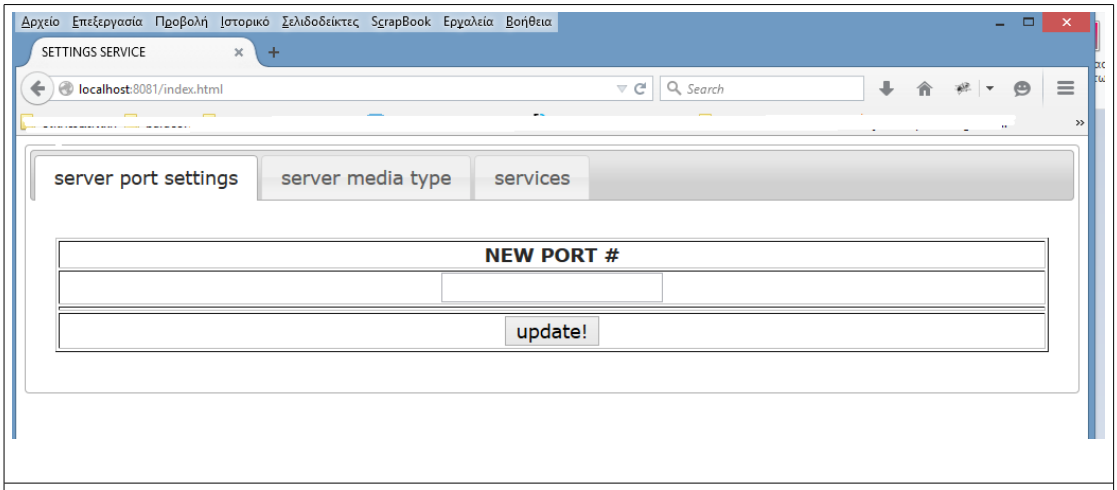
Εικόνα 2: Τι θέλει να δει ο χρήστης



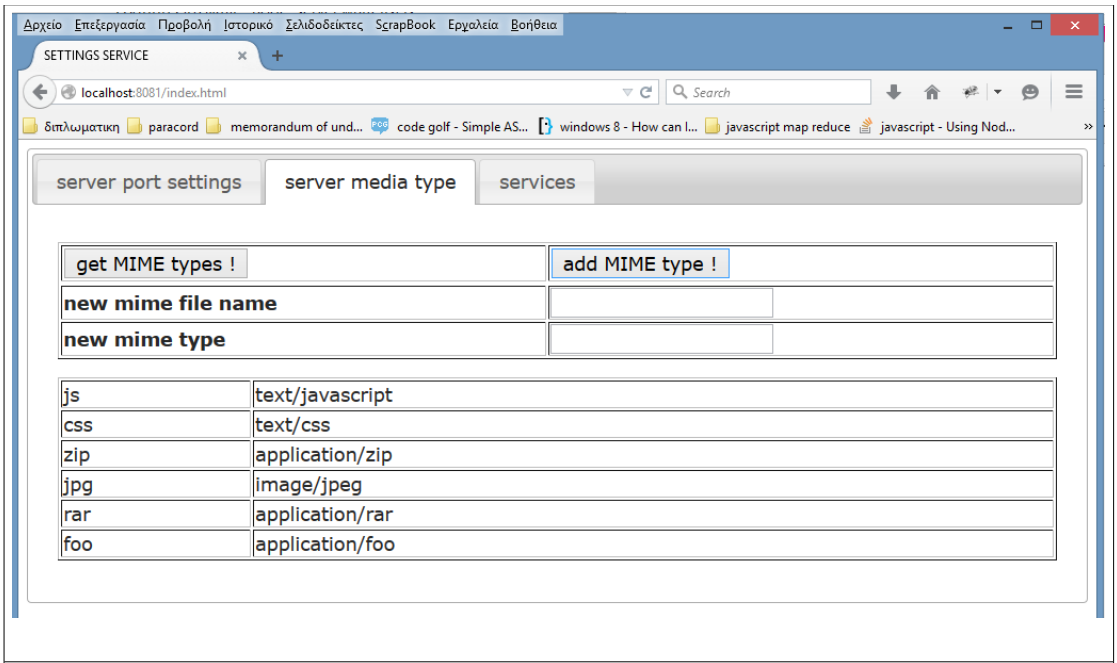
Εικόνα 3: Ένα υπόδειγμα της πορείας εκτέλεσης του server

Ανάλογη πορεία εκτέλεσης παρουσιάζεται κατά τη παρουσίαση html σελίδων ενώ κατά την αποστολή αρχείων ή τη διαχείριση σφαλμάτων διαφοροποιείται κατά την κλήση των ανάλογων μεθόδων.

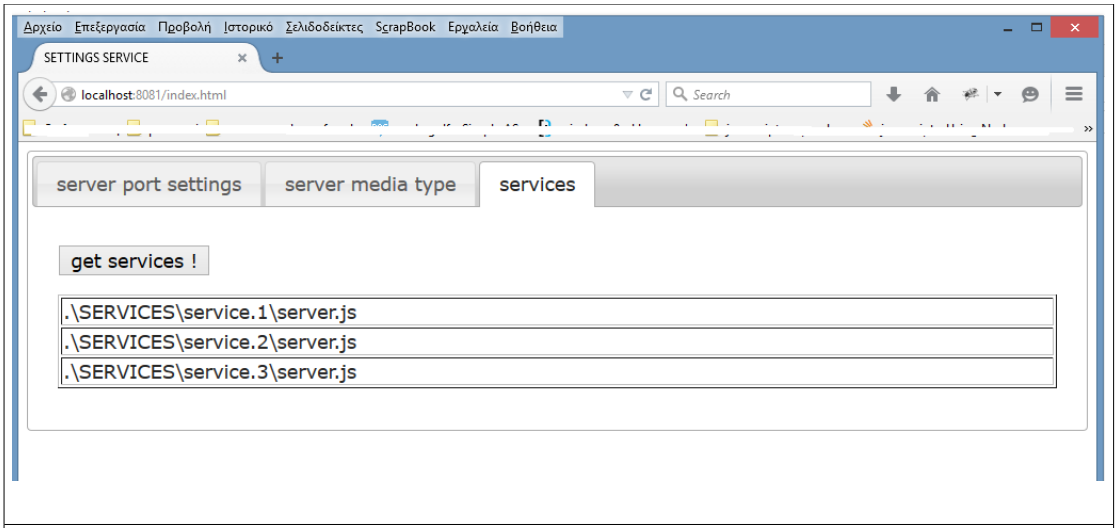
Οι εικόνες 4,5 και 6 αντιπροσωπεύουν ένα δείγμα του admin interface



Εικόνα 4: Το μενού ρύθμισης της θύρας του server



Εικόνα 5: Το μενού ρύθμισης των mime types τα οποία χρησιμοποιεί ο server



Εικόνα 6: Ο κατάλογος των services που αρχικοποιεί ο server

Συμπεράσματα

X

Το συνολικό αποτέλεσμα της εργασίας περιλαμβάνει:

- Ένα σύνολο εργαλείων με την βοήθεια των οποίων μπορεί να κατασκευαστεί ένα ασφαλές περιβάλλον λειτουργίας για ένα web server σε Android
- Ένα σύνολο μεθόδων κατασκευής του Node.js για android οι οποίες δοκιμάστηκαν για την κατασκευή του node.js και για την υλοποίηση του web server
- Ένα web interface και ένα REST api με το οποίο να μπορεί να αλληλεπιδράσει ο administrator με τον server.

Συνολικά, ο ίδιος ο server έδειξε μια καλή συμπεριφορά για μεγάλα διαστήματα λειτουργίας. Το υλικό πάνω στο οποίο έτρεχε όμως έδειξε σημάδια καταπόνησης κάτι που σημαίνει ότι:

- Είναι σημαντικό να λαμβάνονται περιοριστικά μέτρα απέναντι στην καταπόνηση των επιμέρους υποσυστημάτων.
- Απαιτείται σημαντικός βαθμός οικειότητας ανάμεσα στον εκάστοτε developer και τις συσκευές στις οποίες γράφει εφαρμογές όπως αυτές.
- Ένα σημαντικό κομμάτι εφαρμογών το οποίο μπορεί να συμπληρώσει θετικά τον web server της παρούσας πτυχιακής αφορά σε προγράμματα που εστιάζουν στην λήψη στοιχείων για την λειτουργία του hardware της συσκευής και πιο συγκεκριμένα για τον φόρτο που ο web server ασκεί στην συσκευή.

Εν τούτοις, για ένα μετριοπαθή φόρτο εργασίας ένα σύνολο εφαρμογών στις οποίες μπορεί να βρει εφαρμογή ο συγκεκριμένος web server αφορά στην παροχή πληροφοριών από τους αισθητήρες με την βοήθεια του scemantic web, με την έννοια ότι η συσκευή μπορεί να:

- Αντλεί πληροφορίες που αφορούν τον χρήστη (από το GPS, από QR codes τα οποία εισάγει ο χρήστης με την κάμερα, από δεδομένα που εισάγονται με κάποιο ασύρματο πρωτόκολλο -bluetooth για παράδειγμα)
- Να τις επεξεργάζεται με σκοπό να αποτελέσουν είσοδο σε κάποια άλλη απομακρυσμένη υπηρεσία
- Να αποτελέσουν αντικείμενο παρουσίασης στον χρήστη σε μορφή αναφοράς μέσω ενός browser είτε στην ίδια τη συσκευή είτε σε κάποια άλλη συσκευή.

Επίσης αξίζει να σημειωθεί ότι το node.js δεν περιορίζεται σαν περιβάλλον ανάπτυξης εφαρμογών από το πρωτόκολλο HTTP, μέχρι τώρα ένα πλήθος πρωτοκόλλων έχει αναπτυχθεί, κάτι που σημαίνει πως υπάρχει η δυνατότητα μετατροπής μιας συσκευής που το υποστηρίζει σε IRC server, την δυνατότητα αποστολής email ή την ανταλλαγή αρχείων ανάμεσα σε συσκευές με το πρωτόκολλο Bittorent.

Αυτό συνεπάγεται ένα απεριόριστο εύρος εφαρμογών το οποίο δεν περιορίζεται εύκολα.

Βιβλιογραφία

XI

1	Wikipedia https://en.wikipedia.org/wiki/Smartphone
2	HTTP Made Really Easy (HTTP Made Really Easy) http://www.jmarshall.com/easy/http/
3	DroidPHP/DroidPHP (GitHub) https://github.com/DroidPHP/DroidPHP
4	KSWEB: server + PHP + MySQL - Android Apps on Google Play https://play.google.com/store/apps/details?id=ru.kslabs.ksweb
5	A Web Server in 150 Lines http://cs.au.dk/~amoeller/WWW/javaweb/server.html
6	A Simple Web Server http://www.java2s.com/Code/Java/Network-Protocol/ASimpleWebServer.htm
7	Simple TCP/IP web server (Simple TCP/IP web server) http://www.paulgriffiths.net/program/c/webserv.php
8	Hello world/Web server (- Rosetta Code) http://rosettacode.org/wiki/Hello_world/Web_server
9	nweb: a tiny, safe Web server http://www.ibm.com/developerworks/systems/library/es-nweb/
10	damonkohler/sl4a (GitHub) https://github.com/damonkohler/sl4a
11	Rhino (Mozilla Developer Network) https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino
12	Wikipedia http://en.wikipedia.org/wiki/Chroot
13	Linux Deploy - Android Apps on Google Play https://play.google.com/store/apps/details?id=ru.meefik.linuxdeploy&hl=el
14	GNURoot - Android Apps on Google Play https://play.google.com/store/apps/details?id=champion.gnuroot

15	Evolution of Server-Side Java Script http://java.dzone.com/articles/evolution-server-side-java
16	Wikipedia https://en.wikipedia.org/wiki/JavaScript#Server-side_JavaScript
17	Wikipedia https://en.wikipedia.org/wiki/Comparison_of_server-side_JavaScript_solutions
18	kripken/emscripten (GitHub) https://github.com/kripken/emscripten/wiki
19	Opal Ruby to Javascript Compiler (Opal: Ruby to Javascript Compiler) http://opalrb.org/
20	Rusthon http://rusthon.github.io/Rusthon/
21	Wikipedia https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
22	Wikipedia https://en.wikipedia.org/wiki/Apache_HTTP_Server
23	Wikipedia https://en.wikipedia.org/wiki/Nginx
24	Simple 6.0.1 http://simpleweb.sourceforge.net/
25	Jetty (Servlet Engine and Http Server) http://www.eclipse.org/jetty/
26	com.sun.net.httpserver (Java HTTP Server) http://docs.oracle.com/javase/6/docs/jre/api/net/httpserver/spec/com/sun/net/httpserver/package-summary.html
27	Wikipedia https://en.wikipedia.org/wiki/Plone_%28software%29
28	How To Turn Your Android Tablet Into A Web Development Environment (Antony D'Andrea) http://antonydandrea.com/how-to-turn-your-android-tablet-into-a-web-development-environment/
29	Working on an Android tablet (Henri Bergius) http://bergie.iki.fi/blog/working-on-android/
30	damonkohler/sl4a https://github.com/damonkohler/sl4a
31	AndroWish http://www.androwish.org/index.html/home
32	Android framework for node.js applications https://github.com/paddybyers/anode
33	Run Node.js on Android https://github.com/InstantWebP2P/node-android

34	http://readwrite.com/2011/03/22/running-nodejs-on-a-rooted-android-phone
35	Building and running Node.js for Android http://www.goland.org/nodejsonandroid/
36	Debian on Android (Debian on Android) https://evilzone.org/android/debian-on-android/
37	TechFindings (TechFindings) http://zo0ok.com/techfindings/archives/1138
38	Debian on Android https://evilzone.org/android/debian-on-android/
39	Debian on Android https://evilzone.org/android/debian-on-android/
40	Create a virtual hard drive volume within a file Linux https://thelinuxexperiment.com/create-a-virtual-hard-drive-volume-within-a-file-in-linux/
41	Quickly create a large file on a Linux system http://stackoverflow.com/questions/257844/quickly-create-a-large-file-on-a-linux-system