



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ενσωματωμένα Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ ANDROID-DEVKIT Νο:04

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα

1.	Σκοπός της άσκησης	4
2.	Παραδοτέα	4
3.	Περιγραφή της εργαστηριακής άσκησης	4
3.1	Προσαρμογή του συστήματος	4
3.1.1	Εγκατάσταση νέου προϊόντος	5
3.1.2	Αλλαγή φόντου εργασίας.....	5
3.1.3	Προσθήκη έξτρα ιδιοτήτων στο σύστημα.....	5
3.2	Δημιουργία μιας βιβλιοθήκης.....	6
3.2.1	Δημιουργώντας μια στατική βιβλιοθήκη.....	6
3.2.2	Δημιουργώντας μια κοινή βιβλιοθήκη	6
3.2.3	Ενσωμάτωση της βιβλιοθήκης στην εικόνα του Android	7
3.3	Προσθήκη μιας εγγενούς (<i>native</i>) εφαρμογής στη δομή	7
3.3.1	Προσθήκη του δυαδικού αρχείου στη μεταγλωττισμένη εικόνα	7
3.3.2	Διόρθωση προβλημάτων	8
3.4	Ανάπτυξη βιβλιοθήκης JNI	8
3.4.1	Γράψιμο συνδέσεων.....	8
3.4.2	Ενσωμάτωση με το δομικό σύστημα	9
3.4.3	Επέκταση.....	10
3.5	Δημιουργία εφαρμογής με το SDK	10
3.5.1	Εγγραφή της εφαρμογής	10
3.5.2	Ενσωμάτωση της εφαρμογής.....	10
3.5.3	Επέκταση.....	11

1. Σκοπός της άσκησης

Τροποποίηση χαρακτηριστικών συστήματος. Προσθήκη εφαρμογών..

2. Παραδοτέα

(A) 3 ερωτήσεις

(C) 9 ασκήσεις

- **Παραδοτέο C1:** Demonstration or Video.
- **Παραδοτέο C2:** Demonstration or Video.
- **Παραδοτέο C3:** Demonstration or Video.
- **Παραδοτέο C4:** Demonstration or Video or Screenshot.
- **Παραδοτέο C5:** Demonstration or Video or Screenshot.
- **Παραδοτέο C6:** Το αρχείο Android.mk
- **Παραδοτέο C7:** Demonstration or Video or Screenshot.
- **Παραδοτέο C8:** Το αρχείο ueventd.omap3.rc.
- **Παραδοτέο C9:** Demonstration or Video or Screenshot.

Σημείωση:

Τα εργαστήρια απαιτούν την ύπαρξη λειτουργικού συστήματος Linux. Θα πρέπει να γίνει κανονική εγκατάσταση του λειτουργικού και όχι εκτέλεση μέσω εικονικής μηχανής καθώς κάτι τέτοιο δεν υποστηρίζεται. Η παρούσα εργασία έχει υλοποιηθεί σε Ubuntu 10.4 64bit, το οποίο και προτείνεται.

Το υλικό του εργαστηρίου αναπτύχθηκε σε συνεργασία με τον κ. Τάσο Φετινίδη.

Χρησιμοποιήθηκε υλικό από [http:// free-electrons.com](http://free-electrons.com) .

3. Περιγραφή της εργαστηριακής άσκησης

3.1 Προσαρμογή του συστήματος

Μετά από αυτό το εργαστήριο, θα είστε σε θέση να:

- ✓ Χρησιμοποιήσετε το σύστημα ρύθμισης προϊόντος.
- ✓ Αλλάξετε το προκαθορισμένο φόντο εργασίας.
- ✓ Προσθέσετε παραπάνω ιδιότητες στο σύστημα.
- ✓ Χρησιμοποιήσετε τις επικαλύψεις του προϊόντος.

3.1.1 Εγκατάσταση νέου προϊόντος

Επαναφέρετε τις ρυθμίσεις που κάνατε για την πλακέτα beagleboard και ορίστε μια νέα με το όνομα training. Αυτή η πλακέτα θα έχει όλες τις ιδιότητες της πλακέτας beagleboard προς το παρόν και κάποιες παραπάνω που θα προσθέσουμε κατά τη διάρκεια των εργαστηρίων.

Καθώς χρησιμοποιούμε την ίδια πλατφόρμα, το όνομα του προϊόντος – πλακέτας που επιστρέφει ο πυρήνας δεν αλλάζει ακόμα και αν μεταγλωττίσουμε ένα διαφορετικό προϊόν. Αλλάξτε το όνομα του προϊόντος που χρησιμοποιείτε στη γραμμή εντολών και περνάει στον πυρήνα έτσι ώστε να ταιριάζει με το όνομα του προϊόντος σας.

Μπορείτε να περάσετε το όνομα του προϊόντος με την παράμετρο του πυρήνα **androidboot.hardware**

Τώρα, ρυθμίστε ένα προσωπικό αρχείο ρυθμίσεων που θα χρησιμοποιηθεί από το προϊόν σας και κάντε το να δημιουργήσει τον κατάλογο **/data/hello_world**.

Καθώς θα χρειαστεί να αντιγράψουμε προ – κατασκευασμένα αρχεία στην εικόνα, προτιμάμε να καθορίσουμε έτοιμες μονάδες - modules πάνω από τον αρχαιωμένο μηχανισμό PRODUCT_COPY_FILES.

Το σύστημα θα πρέπει να μπορεί να μεταγλωττίσει και να εκκινήσει απροβλημάτιστα με την πλακέτα **DevKit8000** και με όλες τις διορθώσεις που κάναμε νωρίτερα.

Παραδοτέο C1: Demonstration or Video.

3.1.2 Αλλαγή φόντου εργασίας

Αρχικά, ορίστε ένα κενό overlay στον κατάλογο του προϊόντος σας. Το αρχικό φόντο εργασίας βρίσκεται στον κατάλογο: **frameworks/base/core/res/res/drawable/**

Χρησιμοποιήστε τον μηχανισμό overlay για να αντικαταστήσετε το υπάρχον φόντο με ένα της επιλογής σας χωρίς να τροποποιήσετε τον αρχικό πηγαίο κώδικα.

Παραδοτέο C2: Demonstration or Video.

3.1.3 Προσθήκη έξτρα ιδιοτήτων στο σύστημα.

Όπως είδαμε, οι ιδιότητες χρησιμοποιούνται εκτενώς από το λειτουργικό σύστημα Android. Επεκτείνετε το αρχείο system.prop από το προϊόν training προσθέτοντας μια ιδιότητα **foo.bar**. Εκκινήστε το σύστημα και χρησιμοποιήστε την εντολή getprop για να ελέγξετε ότι η εντολή έχει όντως προστεθεί.

Παραδοτέο C3: Demonstration or Video.

3.2 Δημιουργία μιας βιβλιοθήκης

Μετά από αυτό το εργαστήριο, θα είστε ικανοί να:

- ✓ προσθέσετε μια εξωτερική βιβλιοθήκη στο δομικό σύστημα του Android.
- ✓ την μεταγλωττίσετε στατικά και δυναμικά.
- ✓ προσθέσετε ένα συστατικό στη δομή.

3.2.1 Δημιουργώντας μια στατική βιβλιοθήκη

Για να πάρετε τον πηγαίο κώδικα του libusb, μεταβείτε στη σελίδα www.libusb.org και κατεβάστε την έκδοση 1.0.9.

Αποσυμπίστε το αρχείο στον φάκελο **external/libusb**.

Για αυτή την βιβλιοθήκη, όλα τα απαιτούμενα αρχεία **.c** βρίσκονται στον φάκελο libusb και στους υποφακέλους του.

Οι επικεφαλίδες βρίσκονται και αυτές στους ίδιους φακέλους.

Κάντε το δομικό σύστημα να δημιουργήσει ένα αρχείο **libusb-static.a**

Ερώτηση A1: Το παραπάνω αρχείο έχει κατάληξη **.a** .

Τι συμβολίζει αυτή η κατάληξη; Ποια είναι η μορφή των αρχείων που έχουν την κατάληξη **.a** ;

Θα βρείτε μια απύουσα επικεφαλίδα που θα χρειαστεί να δημιουργήσετε.

Στη πραγματικότητα, το αρχείο config.h που παράγεται από το autconf δεν δημιουργείται καθόλου καθώς το δομικό σύστημα του Android αγνοεί όλα τα άλλα δομικά συστήματα. Θα πρέπει να το δημιουργήσετε μόνοι σας.

Αν το κάνετε με επιτυχία, το δομικό σύστημα θα πρέπει να προχωρήσει στη διαδικασία της κατασκευής και θα πρέπει να δείτε έναν νέο φάκελο στον κατάλογο out.

Παραδοτέο C4: Demonstration or Video or Screenshot.
--

3.2.2 Δημιουργώντας μια κοινή βιβλιοθήκη

Τώρα, θα χρειαστεί να χτίσουμε μια κοινή βιβλιοθήκη παράλληλα με την στατική και θα την ονομάσουμε libusb.so.

Ερώτηση A2: Το παραπάνω αρχείο έχει κατάληξη **.so** .

Τι συμβολίζει αυτή η κατάληξη; Ποια είναι η μορφή των αρχείων που έχουν την κατάληξη **.so** ;

Χρησιμοποιήστε το ίδιο αρχείο config.h και τροποποιήστε το αρχείο Android.mk έτσι ώστε να μεταγλωττίζετε. Έπειτα, ελέγξτε ότι το κοινό αντικείμενο είναι παρών στον φάκελο out.

Παραδοτέο C5: Demonstration or Video or Screenshot.

3.2.3 Ενσωμάτωση της βιβλιοθήκης στην εικόνα του Android

Όπως μπορείτε να δείτε, η βιβλιοθήκη σας έχει μεταγλωττιστεί κατά την διεργασία της δόμησης, αλλά αν εκκινήσετε την δημιουργημένη εικόνα ή αν κοιτάξετε μέσα στον κατάλογο out/target/product/system/lib folder, μπορείτε να δείτε ότι το κοινό αντικείμενο δεν είναι παρών.

Τροποποιήστε τα κατάλληλα αρχεία έτσι ώστε μέσα στις εικόνες να έχετε δυο παραλλαγές από την βιβλιοθήκη που μεταγλωττίσαμε.

Ερώτηση A3: Ποια αρχεία τροποποιήσατε και πως;

3.3 Προσθήκη μιας εγγενούς (*native*) εφαρμογής στη δομή

Μετά από αυτό το εργαστήριο, θα είστε σε θέση να:

- ✓ προσθέσετε ένα εξωτερικό δυαδικό αρχείο στο σύστημα.
- ✓ διατυπώσετε τις εξαρτήσεις σε άλλα στοιχεία στο δομικό σύστημα.

3.3.1 Προσθήκη του δυαδικού αρχείου στη μεταγλωττισμένη εικόνα

Αντιγράψτε το αρχείο **mlbin.c** από τον κατάλογο **/home/<user>/felabs/android/native-app** και τοποθετήστε το στον κατάλογο **external/ml** .

Όπως και στη περίπτωση του libusb, πρέπει να κάνετε ένα αρχείο **Android.mk** και να δώσετε όλες τις λεπτομέρειες που χρειάζονται από το δομικό σύστημα για την μεταγλώττιση. Αλλά, σε αντίθεση με το libusb, το δυαδικό αρχείο είναι εκτελέσιμο και εξαρτάται από άλλα κομμάτια λογισμικού.

Παραδοτέο C6: Το αρχείο Android.mk.

Κάντε το να μεταγλωττιστεί και να ενσωματωθεί στις δημιουργημένες εικόνες. Όταν έχετε τις εικόνες, φορτώστε την πλακέτα και τρέξτε τον εκτοξευτή πυραύλων για να δοκιμάσετε την εφαρμογή. Θα πρέπει να δείτε τον εκτοξευτή να μετακινείται.

Παραδοτέο C7: Demonstration or Video or Screenshot.

3.3.2 Διόρθωση προβλημάτων

Ωστόσο, όταν ξεκινήσετε τις δοκιμές, θα δείτε ότι το libusb δε μπορεί να ανοίξει όλες τις συσκευές usb λόγω περιορισμένων δικαιωμάτων.

Αυτό μπορεί να διορθωθεί μέσα από το αρχείο **ueventd.rc**.

Προσθέστε μια ρύθμιση συγκεκριμένης συσκευής ueventd στο αρχείο της δομής σας για να κάνετε τα αρχεία στον κατάλογο **/dev/bus/usb/** καθολικά προσβάσιμα.

Προσοχή

Ο ueventd parser περιέχει σφάλματα στην έκδοση Gingerbread και δεν διαβάζει την μεταβλητή androidboot.

hardware

Θα πρέπει να ονομάσετε το αρχείο **ueventd.omap3.rc**.

Παραδοτέο C8: Το αρχείο ueventd.omap3.rc.
--

3.4 Ανάπτυξη βιβλιοθήκης JNI

Στο τέλος αυτού του εργαστηρίου, θα είστε ικανοί να:

- ✓ Αναπτύξετε συνδέσεις από Java σε C.
- ✓ Ενσωματώσετε αυτές τις συνδέσεις στο δομικό σύστημα.
- ✓ Τροποποιήσετε το Android framework.
- ✓ Χρησιμοποιήσετε JNI συνδέσεις.

3.4.1 Γράψιμο συνδέσεων

Ο κώδικας θα πρέπει να είναι αρκετά παρόμοιος με αυτόν που γράψατε στο αντίστοιχο εργαστήριο εφαρμογών. Έτσι, θα βρείτε ένα σκελετό του φακέλου για να ενσωματώσετε στο προϊόν σας στον κατάλογο **/home/<user>/felabs/android/jni** .

Κυρίως, θα έχετε να τροποποιήσετε τα πρωτότυπα των συναρτήσεων από τις προηγούμενες εφαρμογές με σκοπό να δουλέψουν με το JNI.

Ως υπενθύμιση, το JNI απαιτεί τα πρωτότυπα των συναρτήσεων να είναι όπως αυτό:

```
JNIEXPORT <jni type>  
JNICALL Java_<package_class>_<function_name>(JNIEnv *env, jobject this)
```

Προσέξτε ότι το όνομα της συνάρτησης δεν μπορεί να περιέχει το underscore (_) αν θέλετε να λειτουργεί σωστά.

Εκτός από το φάκελο JNI, υπάρχει επίσης ένας φάκελος java που περιέχει μια διεπαφή Java, την MissileBackendImpl.

Στον ίδιο φάκελο, γράψτε την USBBackend κλάση που υλοποιεί το interface που χρησιμοποιεί τις συνδέσεις σας.

Έχετε ένα παράδειγμα μιας τέτοιας κατηγορίας στο αρχείο DummyBackend.java.

3.4.2 Ενσωμάτωση με το δομικό σύστημα

Τώρα, μπορείτε να το ενσωματώσετε στο δομικό σύστημα, έτσι ώστε να παράγει μια βιβλιοθήκη με κατάληξη .jar που είναι στο προϊόν σας με τις εξαρτήσεις σωστά εκφρασμένες.

Μπορείτε να βρείτε την τεκμηρίωση για το πως να ενσωματώσετε τμήματα συγκεκριμένης συσκευής στον κατάλογο **device/sample/frameworks**.

3.4.3 Δοκιμάζοντας τις συνδέσεις

Θα πρέπει τώρα να έχουμε ένα σύστημα με τα αρχεία:

/system/framework/com.fe.android.backend.jar, που περιλαμβάνουν τις Java classes, **/system/lib/liblauncher_jni.so**, τις JNI συνδέσεις καθώς και το **/system/lib/libusb.so**.

Δείτε τι κάνατε χρησιμοποιώντας την Main κλάση που υπάρχει στον πηγαίο κώδικα Java καλώντας απευθείας το Dalvik μέσω της εντολής app_process. Θα πρέπει να παρέχετε το μονοπάτι προς την κλάση καθώς και το όνομα της κλάσης για να λειτουργήσει. Θα πρέπει να μοιάζει κάπως έτσι:

```
CLASSPATH=path/to/java.jar app_process /system/bin com.fe.android.Main
```

Παραδοτέο C8: Demonstration or Video or Screenshot.

3.4.4 Επέκταση

Θα δείτε ότι τα δυαδικά αρχεία που δημιουργήσαμε στο προηγούμενο εργαστήριο και οι συνδέσεις μοιράζονται αρκετά κοινά τμήματα κώδικα. Αυτό δεν είναι και πολύ βολικό για να διορθώσετε προβλήματα και bugs επηρεάζουν αυτά τα τμήματα καθώς θα πρέπει να διορθωθούν σε δυο διαφορετικά μέρη.

Αν έχετε χρόνο, ασχοληθείτε με σκοπό να κάνετε αυτά τα κοινά τμήματα κώδικα μέρος μια κοινής βιβλιοθήκης που θα χρησιμοποιείτε τόσο από τις συνδέσεις όσο και από τα δυαδικά αρχεία.

3.5 Δημιουργία εφαρμογής με το SDK

Στο τέλος αυτού του εργαστηρίου, θα είστε σε θέση να:

- ✓ Γράψετε μια εφαρμογή Android
- ✓ Ενσωματώσετε μια εφαρμογή στο Android δομικό σύστημα

3.5.1 Εγγραφή της εφαρμογής

Μεταβείτε στον κατάλογο `/home/<user>/felabs/android/app`.

Θα βρείτε τα βασικά αρχεία για την εφαρμογή `MissileControl`.

Αυτή η εφαρμογή είναι ατελής καθώς μέρος από κάποια `activities` απουσιάζουν.

Ωστόσο το UI μέρος (*διεπαφή χρήστη*) είναι σχεδόν έτοιμο. Έτσι, γράψτε τον κώδικα που θα εκκινεί την δραστηριότητα με την δεδομένη διάταξη και τους απαιτούμενους ελέγχους όπως ορίζονται στην περιγραφή.

Αυτοί οι έλεγχοι προσαρμόζουν την συμπεριφορά κάθε κουμπιού: μετακινούν τον εκτοξευτή, πυροβολούν και άλλα. Κάθε κουμπί θα πρέπει να είναι επεξηγηματικό από μόνο του, εκτός από τον διακόπτη USB/Emulation που εναλλάσσει την μέθοδο μεταξύ USB και Emulation.

Η εφαρμογή τώρα χρησιμοποιεί τρία επίπεδα εργασίας, την εφαρμογή καθ' εαυτή που είναι μια απολύτως κανονική Android εφαρμογή που βασίζεται σε συνδέσεις Java → C ενσωματωμένες στο Android framework, το οποίο με την σειρά του βασίζεται στο `libusb` που συμπεριλάβαμε στις βιβλιοθήκες του συστήματος.

3.5.2 Ενσωμάτωση της εφαρμογής

Για ακόμη μια φορά, σημειώστε στο αρχείο `Android.mk` για να χτίσει την εφαρμογή στη εικόνα Android και να θέσει τις εξαρτήσεις στις βιβλιοθήκες JNI έτσι ώστε η εφαρμογή να μεταγλωττιστεί και να τρέχει σωστά.

Όταν δοκιμάζετε την εφαρμογή, θα δείτε ότι κρασάρει γιατί δεν βρίσκει το αρχείο .jar που περιέχει της Java συνδέσεις στο libusb.

Αυτό συμβαίνει διότι το μοντέλο ασφαλείας του Android αρνείται να φορτώσει μη έμπιστα αρχεία .jar.

Ωστόσο, μπορείτε να κάνετε το Android να δέχεται ένα αρχείο .jar με το να τοποθετήσετε ένα xml αρχείο παρόμοιο με το AndroidManifest.xml στον κατάλογο /system/etc/permissions.

Υπάρχουν πολλά τέτοια παραδείγματα από τέτοια αρχεία σε ένα φάκελο, οπότε υιοθετήστε ένα για την περίπτωση μας.

Μετά την επανοικοδόμηση (rebuild), η εφαρμογή θα πρέπει να δουλεύει σωστά.

Παραδοτέο C9: Demonstration or Video or Screenshot.
--

3.5.3 Επέκταση

Χρησιμοποιήσαμε απευθείας συνδέσεις και κλήσεις για να χρησιμοποιήσουμε τις βιβλιοθήκες μας. Ωστόσο, είδαμε ότι με το Android, τείνουμε να χρησιμοποιήσουμε υπηρεσίες, για λόγους ασφαλείας καθώς και για να διατηρήσουμε μια συνεπή κατάσταση στο σύστημα.

Αν έχετε διαθέσιμο χρόνο, δημιουργήστε μια νέα υπηρεσία που θα χρησιμοποιείται από την εφαρμογή σας και αντικαταστήστε τις απευθείας κλήσεις συναρτήσεων.