

ANDROID PROJECT: TASK_MANAGER

ΝΙΚΟΣ ΓΕΩΡΓΙΑΔΗΣ ΑΕΜ 2043

ΘΟΔΩΡΗΣ ΦΑΣΟΥΛΑΣ ΑΕΜ 2096

Το task_manager είναι μια εφαρμογή στην οποία ο χρήστης μπορεί να βρει διαφορές γενικές πληροφορίες του κινητού , να δει πίες διεργασίες εκτελούνται την δεδομένη χρονική στιγμή καθώς και οποιαδήποτε χρονική στιγμή θέλει εφόσον υπάρχει μια βάση δεδομένων στο κινητό η οποία ανανεώνεται κάθε 20 δευτερόλεπτα και παρέχει σημαντικές πληροφορίες για την κάθε διεργασία που τρέχει στο παρασκήνιο. Ακόμα ο χρήστης έχει την δυνατότητα να τερματίσει όλες τις διεργασίες του παρασκήνιου.

Περίληπτικά ο σχεδιασμός και η υλοποίησης της παραπάνω εφαρμογής:

Από την activity splash.java η οποία το μόνο που κάνει είναι να φορτώνει για 5 δευτερόλεπτα μια εικόνα , καλούμε με την intent την menu.java η οποία δημιουργεί ένα ListAdapter δλδ σαν να φτιάχνω ένα TableRow σε xml . Σε αυτό το ListAdapter έχω την επιλογή General_Information όπου βλέπω της γενικές πληροφορίες του κινητού όπως το όνομα του λειτουργικού, την έκδοση του πυρήνα , την γλώσσα του συστήματος, την εξωτερική και εσωτερική χωρητικότητα του κινητού και το υπόλοιπο της μπαταρίας . Το επόμενο κελί είναι το Service όπου εμφανίζονται όλες οι διεργασίες που τρέχουν εκείνη την στιγμή καθώς και το pid τους το package name τους και κάποιες πληροφορίες για την μνήμη που δεσμεύουν. Το παρακάτω κελί είναι το ram στο οποίο υπάρχει η χωρητικότητας της μνήμης καθώς και ο ελεύθερος χώρος της μνήμης του κινητού. Το τελευταίο κελί είναι το kill_proc στο οποίο έχουμε την δυνατότητα να τερματίσουμε όλες τις διεργασίες του παρασκήνιου.

General Information:

Δημιουργούμε ένα αντικείμενο BroadcastReceiver

Η παρακάτω συνάρτηση καλείται όταν δεχτούμε ένα αντικείμενο intent
onReceive(Context, Intent);

Η intent.getIntExtra(); επιστρέφει το υπόλοιπο της μπαταρίας στα 100.

Η system.getProperty("os.version"); επιστρέφει την έκδοση του πυρήνα

φτιάχνουμε ένα αντικείμενο statfs και στον κατασκευαστή περνάμε αυτή την παράμετρο (Environment.getExternalStorageDirectory().getAbsolutePath()); Για να βρούμε την χωρητικότητα της εξωτερικής μνήμης ενώ για να βρούμε της εσωτερικής περνάμε αυτήν (Environment.getRootDirectory().getAbsolutePath());

getAvailableBlocks() * statFs.getBlockSize() μας επιστρέφουν τις αντιστοιχες χωρητικοτιτες.

MENU.JAVA

Δημιουργείται το MENU με τις καθε επιλογες.

```
setListAdapter(new ArrayAdapter<String>(Menu.this,android.R.layout.simple_list_item_1, classes));
```

Μολις πατηθει ενα κελι απο το μενου τοτε καλειται αυτη

```
onListItemClick(ListView l, View v, int position, long id)
```

Το commands ειναι το αντιστοιχο ονομα του καθε κελιου.τα ονοματα αυτα ειναι αποθηκευμενα σε εναν πινακα .στη συνεχεια η intent δημιουργει το εκαστοτε activity.

```
Class ourClass = Class.forName("com.example.task_manager."+ commands);
```

```
Intent ourIntent = new Intent(Menu.this, ourClass);
```

```
startActivity(ourIntent);
```

Service.java

```
ActivityManager am = (ActivityManager) this.getSystemService(ACTIVITY_SERVICE);
```

Δημιουργω ενα αντικειμενο activitymanager και στη συνεχεια δημιουργω μια λιστα με αντικειμενα runningserviceinfo τα οποια τα παιρνω Με την συναρτηση getrunningservices παιρνω ολες τις διεργασιες που τρεχουν εκεινη την στιγμη.

Και με διαφορα πεδια οπως το pid και το process παιρνω το pid της διεργασιας και το ονομα της .Με της συναρτησεις getpackagename() και getprocessmemoryinfo παιρνω το ονομα του πακετου της διεργασιας και πληροφοριες για την μνημη που χρησιμοποιει η διεργασία.

Ram.java

Για να παρω το totalmemory και το avalaiblememory φτιαχνω το παρακατω αντικειμενο

```
ActivityManager.MemoryInfo mInfo = new ActivityManager.MemoryInfo();
```

Και με τα πεδια .totalmem και .availmem τα παιρνω.

Kill_proc

Για να σκοτωσω ολες τις διεργασιες πρεπει πρωτα στο manifest να valw to permission KILL_BACKGROUND_PROCESSES και στην συνεχεια να καλεσω αυτην

killallbackgroundprocesses(packagename); παίρνωντας το package name της κάθε διεργασίας.

SQL

Για την υλοποίηση της βάσης μας SQLite χρησιμοποιήθηκαν δυο κλάσεις ,η MyDatabase και η MyDatabaseView .Η πρώτη για την υλοποίηση και η δεύτερη για την εμφάνιση της στην εφαρμογή μας.
Θα αναλύσουμε ξεχωριστά κάθε κλάση :
MyDatabase

Αρχικά ορίζουμε τις μεταβλητές που θέλουμε να αποθηκεύει η βάση μας .
Κρατάμε το PID της διεργασίας ,το όνομα της και το ποσοστό μνήμης που αυτή χρησιμοποιεί.Στη συνέχεια ορίζουμε κάποια επιπλέον στοιχεία της βάσης μας όπως το όνομα της ,την έκδοση της και το όνομα του πίνακα που χρησιμοποιεί.Στη συνέχεια δημιουργούμε κάποια στιγμιότυπα κλάσεων που θα αναλυθούν στη συνέχεια, όπως ο ourHelper το context και φυσικά η βάση ourDatabase.

Επειτα ,έχουμε την static class MyDbHelper η οποία επεκτείνει την κλάση SQLiteOpenHelper η οποία στην ουσία μας παρέχει δυνατότητες και μεθόδους για να δημιουργήσουμε την βάση μας. Συναρτήσεις από την SQL την execSQL με τη βοήθεια της οποίας δημιουργούμε ένα πίνακα με τρεις στήλες μια για κάθε στοιχείο που θέλουμε να αποθηκεύσουμε

Στη συνέχεια αφού τελειώσουμε με τη δημιουργία της βάσης υλοποιούμε συναρτήσεις για τον χειρισμό της. Μια από αυτές είναι η open() η οποία χρησιμοποιείται για να ανοίξουμε την βάση για να γράψουμε σε αυτή.Χρησιμοποιείται εδώ ο myHelper για να ανοίξουμε τη βάση για εγγραφή με την συνάρτηση getWritableDatabase(); . Αντιστοίχα χρησιμοποιείται η συνάρτηση close για να κλείσουμε την βάση μας πάλι με τη βοήθεια του helper. Ακόμα ,έχουμε τις συναρτήσεις getData και createEntry που χρησιμοποιούνται ως βοηθητικές για εμφάνιση και εισαγωγή στη βάση αντιστοίχα. Στην createEntry() χρησιμοποιούμε τη βοηθητική κλάση ContentValues ορίζοντας ένα στιγμιότυπο της και στη συνέχεια με την μέθοδο put(...) εισαγούμε τα δεδομένα στη βάση μας. Η getData() επιστρέφει ένα string το οποίο μπαίνει ως παραμετρο στην TextView και ουσιαστικά είναι βοηθητική για την απεικόνιση της βάσης μας.Αυτό γίνεται με τη βοήθεια του Κερσorra(Cursor) ο οποίος παίρνει το index κάθε στήλης και στη συνέχεια για κάθε index,προσθέτει το String στην συγκεκριμένη θέση στο String result που επιστρέφει η

συναρτηση

μας.

Η DatabaseView είναι μια νέα Activity που ασχολείται με την γραφική απεικόνιση της βάσης. Δηλαδή ορίζεται ένα TextView ,ένα στιγμιότυπο της βάσης MyDatabase db, ανοίγουμε την βάση ,ορίζουμε ένα Sting info το οποίο δεχεται την έξοδο της getData() και έπειτα προσαρμόζουμε το TextView της MyDatabaseView σύμφωνα με το info.