



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
**«Московский государственный технический университет имени Н.Э.
Баумана (национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ДОМАШНЯЯ РАБОТА

«Сервисы»

ДИСЦИПЛИНА: «Разработка мобильного ПО»

Выполнил: студент гр. ИУК4-52Б

_____ (_____ Губин Е.В._____)
(Подпись) (Ф.И.О.)

Проверил:

_____ (_____ Прудяк П.Н._____)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2024 г.

Цель: формирование практических навыков создания различных android-служб.

Задачи:

1. Научиться создавать различные службы для мобильного устройства.
2. Уметь понимать схемы взаимодействия службы с другими элементами платформы Android.
3. Разработать эффективные приложения с учетом аппаратных ограничений мобильных устройств.

Вариант 9

Создать службу контроля ОЗУ устройства. Используя средства службы, приложение должно выводить: общее количество памяти в килобайтах, объем доступной памяти для приложений, объем памяти, используемый для буферов, которые занимаются записью данных на диск, объем памяти в активном использовании. Также необходимо предусмотреть возможность включения и отключения службы.

Листинг программы:

MainActivity.kt:

```
package com.example.hw

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.content.IntentFilter
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var totalMemText: TextView
    private lateinit var availMemText: TextView
    private lateinit var bufferMemText: TextView
    private lateinit var activeMemText: TextView

    private val memoryReceiver = object : BroadcastReceiver() {
        override fun onReceive(context: Context, intent: Intent) {
            val totalMem = intent.getLongExtra("totalMem", 0)
            val availMem = intent.getLongExtra("availMem", 0)
            val bufferMem = intent.getLongExtra("threshold", 0)
            val activeMem = intent.getLongExtra("activeMem", 0)

            totalMemText.text = "Total Memory: $totalMem KB"
```

```

        availMemText.text = "Available Memory: $availMem KB"
        bufferMemText.text = "Buffer Memory: $bufferMem KB"
        activeMemText.text = "Active Memory: $activeMem KB"
    }
}

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    totalMemText = findViewById(R.id.totalMemText)
    availMemText = findViewById(R.id.availMemText)
    bufferMemText = findViewById(R.id.bufferMemText)
    activeMemText = findViewById(R.id.activeMemText)

    val startServiceButton = findViewById<Button>(R.id.startServiceButton)
    val stopServiceButton = findViewById<Button>(R.id.stopServiceButton)

    startServiceButton.setOnClickListener {
        startService(Intent(this, MemoryService::class.java))
    }

    stopServiceButton.setOnClickListener {
        stopService(Intent(this, MemoryService::class.java))
    }

    registerReceiver(memoryReceiver, IntentFilter("MEMORY_UPDATE"))
}

override fun onDestroy() {
    super.onDestroy()
    unregisterReceiver(memoryReceiver)
}
}

```

MemoryService.kt:

```

package com.example.hw

import android.app.Service
import android.content.Intent
import android.os.IBinder
import android.os.Handler
import android.os.Looper
import android.app.ActivityManager
import android.content.Context

class MemoryService : Service() {
    private val handler = Handler(Looper.getMainLooper())
    private val memoryInfo = ActivityManager.MemoryInfo()

    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        startMonitoring()
        return START_STICKY
    }

    private fun startMonitoring() {
        handler.postDelayed(object : Runnable {
            override fun run() {
                updateMemoryInfo()
                handler.postDelayed(this, 5000)
            }
        }, 0)
    }
}

```

```

private fun updateMemoryInfo() {
    val activityManager = getSystemService(Context.ACTIVITY_SERVICE) as ActivityManager
    activityManager.getMemoryInfo(memoryInfo)

    val intent = Intent("MEMORY_UPDATE")
    intent.putExtra("totalMem", memoryInfo.totalMem / 1024)
    intent.putExtra("availMem", memoryInfo.availMem / 1024)
    intent.putExtra("threshold", memoryInfo.threshold / 1024)
    intent.putExtra("activeMem", memoryInfo.totalMem - memoryInfo.availMem)

    sendBroadcast(intent)
}

override fun onBind(intent: Intent?): IBinder? {
    return null
}

override fun onDestroy() {
    super.onDestroy()
    handler.removeCallbacksAndMessages(null)
}
}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
        android:id="@+id/startServiceButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start Service" />

    <Button
        android:id="@+id/stopServiceButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Stop Service" />

    <TextView
        android:id="@+id/totalMemText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Total Memory: " />

    <TextView
        android:id="@+id/availMemText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Available Memory: " />

    <TextView
        android:id="@+id/bufferMemText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Buffer Memory: " />

    <TextView
        android:id="@+id/activeMemText"

```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Active Memory: " />
    </LinearLayout>
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

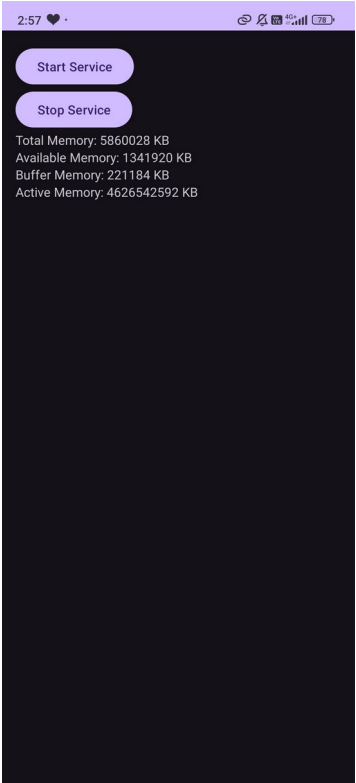
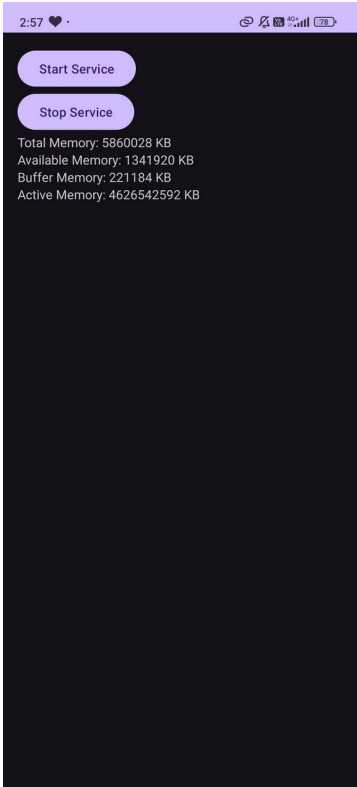
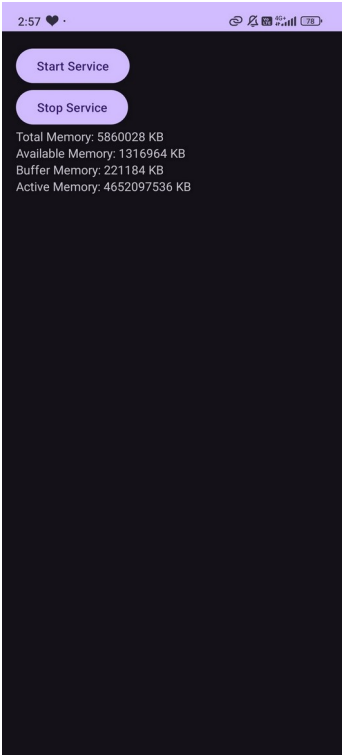
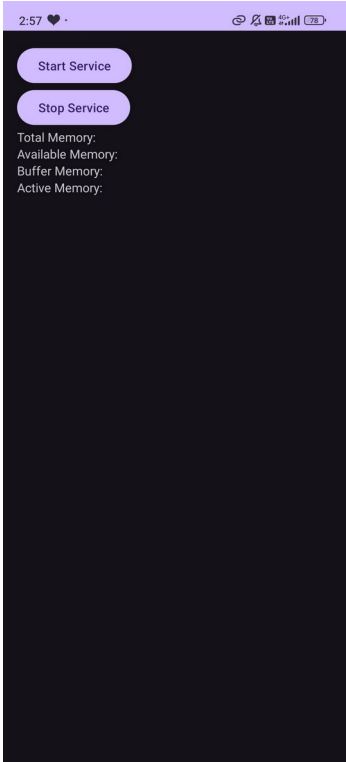
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.HW"
        tools:targetApi="31">
        <service
            android:name=".MemoryService"
            android:enabled="true"
            android:exported="true">

        </service>

        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Результаты работы:



Вывод: в ходе лабораторной работы был реализован сервис по управлению ОЗУ и пользовательский интерфейс для него.