



17-12-2020

Aplicación Paralela Andorid.

Ingeniería en Sistemas Computacionales.

Programación Paralela y Concurrente

Docente: I.S.C. Jesús Aranda Gamboa.

Adán Ruiz Villalobos

Núm. Control: 16070137

Correo: adnruiz1@gmail.com

Semestre: 9°

Instituto Tecnológico Superior de Jerez

Jerez de García Salinas Zacatecas, Zac.



Planteamiento del programa.

Realizar un programa que multiplique dos matrices (3x3) ingresadas por el usuario, dando como resultado la matriz. Este programa fue implementado utilizando programación paralela, para ser mas especifico se utilizo la clase AsyncTask y tres de sus métodos (preExecute, doInBackground y postExecute).

Resultado.

Para esta aplicación solo se utilizo una activity en la cual se agregaron 18 editText en los cuales el usuario deberá ingresar los valores de la matriz. Posteriormente se creó un botón el cual nos servirá para calcular la multiplucacion, finalmente agregamos 9 textView en el cual se mostrara el resultado de la matriz.

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

MULTIPLICAR

<div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>



Posteriormente declaramos las variables y elementos que se usaran para la app.

```
public class MainActivity extends AppCompatActivity {  
    int [][] numeros = new int[3][3];  
    int [][] numeros2 = new int[3][3];  
    int [][] resultado = new int[3][3];  
  
    private EditText a11,a12,a13,a21,a22,a23,a31,a32,a33, b11, b12, b13, b21, b22, b23, b31, b32, b33;  
    private TextView c11, c12, c13, c21, c22, c23, c31, c32, c33;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        //Matriz 1  
        a11=findViewById(R.id.a11);  
        a12=findViewById(R.id.a12);  
        a13 = findViewById(R.id.a13);  
        a21 = findViewById(R.id.a21);  
        a22 = findViewById(R.id.a22);  
        a23 = findViewById(R.id.a23);  
        a31 = findViewById(R.id.a31);  
        a32 = findViewById(R.id.a32);  
        a33 = findViewById(R.id.a33);  
        //Matriz 2  
        b11=findViewById(R.id.b11);  
        b12=findViewById(R.id.b12);  
        b13 = findViewById(R.id.b13);  
        b21 = findViewById(R.id.b21);  
        b22 = findViewById(R.id.b22);  
        b23 = findViewById(R.id.b23);  
        b31 = findViewById(R.id.b31);  
        b32 = findViewById(R.id.b32);  
        b33 = findViewById(R.id.b33);  
        //Matriz de
```



Creamos la clase AsyncTask y sus respectivos métodos.

```
class Task1 extends AsyncTask<String, Void, String>{
    public void multiplicar(){
        for(int i = 0; i < numeros2.length; i++){
            for(int j = 0; j < numeros2.length; j++){
                resultado[i][j] = numeros[i][j] * numeros2[i][j];

                c11.setText(""+resultado[0][0]);
                c12.setText(""+resultado[0][1]);
                c13.setText(""+resultado[0][2]);
                c21.setText(""+resultado[1][0]);
                c22.setText(""+resultado[1][1]);
                c23.setText(""+resultado[1][2]);
                c31.setText(""+resultado[2][0]);
                c32.setText(""+resultado[2][1]);
                c33.setText(""+resultado[2][2]);
            }
        }
    }

    @Override
    protected void onPreExecute() {
        numeros[0][0] = Integer.parseInt(a11.getText().toString());
        numeros[0][1] = Integer.parseInt(a12.getText().toString());
        numeros[0][2] = Integer.parseInt(a13.getText().toString());
        numeros[1][0] = Integer.parseInt(b12.getText().toString());
        numeros[1][1] = Integer.parseInt(b12.getText().toString());
        numeros[1][2] = Integer.parseInt(b12.getText().toString());

        numeros[2][0] = Integer.parseInt(a21.getText().toString());
        numeros[2][1] = Integer.parseInt(a22.getText().toString());
        numeros[2][2] = Integer.parseInt(a23.getText().toString());
        numeros[0][0] = Integer.parseInt(b21.getText().toString());
        numeros[0][1] = Integer.parseInt(b22.getText().toString());
        numeros[0][2] = Integer.parseInt(b23.getText().toString());
    }
}
```

Conclusión.

El manejo de los hilos es muy importante en las aplicaciones móviles ya que cada se pueden crear subprocesos de varias maneras para tener un optimo rendimiento de nuestra aplicación. Lo principal a evitar es que nuestro hilo principal de la aplicación se paralice y nuestra aplicación crash.