

# 임베디드 시스템 실습 lab11

2016310936 우승민

이번 exercise 는 JNI 의 사용법을 익히는 것입니다.

우선 terminal 에서 JNI 을 사용한 exercise 1 먼저 설명하겠습니다.

ex.java 코드입니다.

```
seungmin@seungmin-W65-W67RC: ~/java/5
public class ex
{
    public native int add(int x, int y);
    public native int difference(int x, int y);
    public native int product(int x, int y);
    public native double average(int x, int y);
}
```

구현할 함수들을 전부 native 로 선언해주었습니다.

main.java 코드입니다.

```
seungmin@seungmin-W65-W67RC: ~/java/5
public class main
{
    public static void main(String[] args)
    {
        System.loadLibrary("ex");
        int x = new ex().add(55,22);
        int y = new ex().difference(55,22);
        int z = new ex().product(55,22);
        double k = new ex().average(55,22);
        System.out.println(x);
        System.out.println(y);
        System.out.println(z);
        System.out.println(k);
    }
}
```

ex 를 불러오고 각 함수에 대해서 55 와 22 숫자를 넣은 값들을 출력해주었습니다.

ex.java 코드를 header 파일로 컴파일한 것입니다.

```
seungmin@seungmin-W65-W67RC: ~/java/5
/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class ex */

#ifndef _Included_ex
#define _Included_ex
#ifdef __cplusplus
extern "C" {
#endif
/*
 * Class:      ex
 * Method:     add
 * Signature:  (II)I
 */
JNIEXPORT jint JNICALL Java_ex_add
    (JNIEnv *, jobject, jint, jint);

/*
 * Class:      ex
 * Method:     difference
 * Signature:  (II)I
 */
JNIEXPORT jint JNICALL Java_ex_difference
    (JNIEnv *, jobject, jint, jint);
}
#endif
"ex.h" 45L, 779C
```

ex.c 코드입니다. ex.h 를 헤더파일로 불러오고 각 함수들은 a 와 b 를 인자로 받아서 계산합니다.

```
seungmin@seungmin-W65-W67RC: ~/java/5
#include <stdio.h>
#include "ex.h"

JNIEXPORT jint JNICALL Java_ex_add(JNIEnv *env, jobject obj, jint a, jint b){
    return a+b;
}

JNIEXPORT jint JNICALL Java_ex_difference(JNIEnv *env, jobject obj, jint a, jint b){
    return a-b;
}

JNIEXPORT jint JNICALL Java_ex_product(JNIEnv *env, jobject obj, jint a, jint b){
    return a*b;
}

JNIEXPORT double JNICALL Java_ex_average(JNIEnv *env, jobject obj, jint a, jint b){
    return ((double) a + (double) b) / 2 ;
}

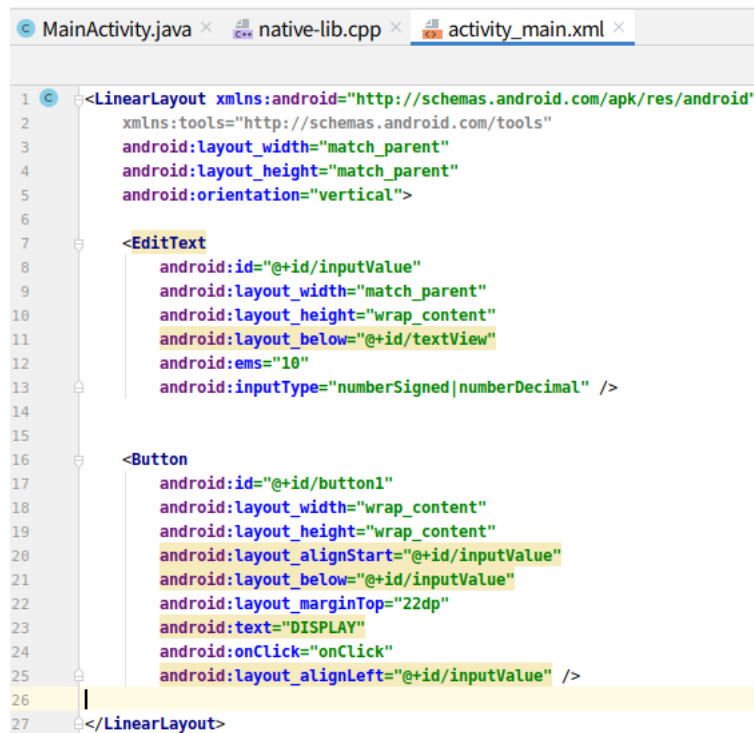
~
```

실행 화면입니다.

```
seungmin@seungmin-W65-W67RC:~/java/5$ vi ex.c
seungmin@seungmin-W65-W67RC:~/java/5$ gcc -shared -fPIC -I /usr/lib/jvm/java-11-
openjdk-amd64/include -I /usr/lib/jvm/java-11-openjdk-amd64/include/linux -o lib
ex.so ex.c
seungmin@seungmin-W65-W67RC:~/java/5$ export LD_LIBRARY_PATH=.
seungmin@seungmin-W65-W67RC:~/java/5$ java main
77
33
1210
38.5
seungmin@seungmin-W65-W67RC:~/java/5$
```

android studio 에서 JNI 를 활용하는 exercise2 에 대해 설명하겠습니다.

우선 layout 을 아래와 같이 작성하였습니다.



```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical">
6
7   <EditText
8     android:id="@+id/inputValue"
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    android:layout_below="@+id/textView"
12    android:ems="10"
13    android:inputType="numberSigned|numberDecimal" />
14
15
16   <Button
17     android:id="@+id/button1"
18     android:layout_width="wrap_content"
19     android:layout_height="wrap_content"
20     android:layout_alignStart="@+id/inputValue"
21     android:layout_below="@+id/inputValue"
22     android:layout_marginTop="22dp"
23     android:text="DISPLAY"
24     android:onClick="onClick"
25     android:layout_alignLeft="@+id/inputValue" />
26
27 </LinearLayout>
```

EditText 에서 숫자를 입력 받게 해주었고, Button 을 통해 segment 에 띄우게 해주었습니다.

아래가 MainActivity 코드입니다.



```
1 package com.example.week11;
2
3 import ...
4
14
15 public class MainActivity extends AppCompatActivity {
16     private EditText text;
17
18     // Used to load the 'native-lib' library on application startup.
19     static {
20         System.loadLibrary( libname: "native-lib");
21     }
22
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.activity_main);
27         text = (EditText) findViewById(R.id.inputValue);
28
29         // Example of a call to a native method
30     }
```

text 에 입력 받은 값을 옮기는 과정입니다.

```

32  @ public void onClick(View view) {
33      switch (view.getId()) {
34          case R.id.button1:
35              if (text.getText().length() == 0) {
36                  Toast.makeText(context: this, text: "Please enter a valid number",
37                      Toast.LENGTH_LONG).show();
38                  return;
39              }
40              int inputValue = Integer.valueOf(text.getText().toString());
41              stringFromJNI(inputValue);
42              break;
43          }
44      }
45
46  /**
47   * A native method that is implemented by the 'native-lib' native library,
48   * which is packaged with this application.
49   */
50  public native String stringFromJNI(int a);
51  }

```

그 후 button 이 눌렸을 때 inputValue 에 입력 받은 내용을 int 형식으로 변환해준 후 함수를 호출합니다.

호출된 함수가 있는 cpp 코드입니다.

```

c MainActivity.java x native-lib.cpp x activit
1  #include <jni.h>
2  #include <string>
3  #include <unistd.h>
4
5  extern "C" JNIEXPORT jstring JNICALL
6  MainActivity.stringFromJNI(
7      JNIEnv* env,
8      jobject /* this */, jint a) {
9      std::string hello = "Hello from C++";
10     syscall( number: 323, a);
11     return env->NewStringUTF(hello.c_str());
12 }

```

syscall 함수를 통해 segment 에 전달받은 숫자를 띄우게 해주었습니다.

실행 화면입니다.

