

Vježbe 6 - postavljanje okruženja

Matej Mihelčić

Prirodoslovno-matematički fakultet, Sveučilište u Zagrebu

matmih@math.hr

16. studenoga, 2022.



Postavljanje okruženja *Java Native Interface (JNI)*

Korištenje *JNI* za poziv *C/C++* funkcija zahtjeva:

- 1 Deklariranje funkcije s ključnom riječi *native* u *Java* kodu.
- 2 Stvaranje dokumenta zaglavlja *C/C++* funkcije koja odgovara našoj sistemskoj funkciji iz *Jave*. `javac Klasa.java -h direktorij`, gdje je `Klasa.java` klasa koja sadrži deklaraciju sistemske *C/C++* funkcije (treba se pozicionirati u direktorij koji sadrži izvorne datoteke *Java* projekta).
- 3 Implementacija dokumenta zaglavlja u odgovarajućoj `.c` ili `.cpp` datoteci.
- 4 Stvaranje dinamičke *C* biblioteke koja sadrži implementiranu funkciju i generirano zaglavlje.
- 5 Povezivanje dinamičke biblioteke s *Java* programom korištenjem naredbe
`static{System.loadLibrary("Putanja\ImeBiblioteke");}`.

Postavljanje okruženja *Java Native Interface (JNI)*

```
1 public class ProbaJNI {
2
3     public native void helloC(); //deklaracija
4
5     static {
6         System.loadLibrary("HelloWorld");
7     }
8
9     public static void main(String[] args) {
10         new ProbaJNI().helloC(); //poziv
11     }
12
13 }
```

Deklariranje sistemske funkcije.

javac generira datoteku zaglavlja imena imepaketa_Imeklase.h a odgovarajuća funkcija ima ime Java_imepaketa_Imeklase_imefunkcije.

Primjer odgovarajućeg zaglavlja C funkcije.

```
1 /* DO NOT EDIT THIS FILE - it is machine generated */
2 #include <jni.h>
3 /* Header for class probajni_ProbaJNI */
4 #ifndef _Included_probajni_ProbaJNI
5 #define _Included_probajni_ProbaJNI
6 #ifdef __cplusplus
7 extern "C" {
8 #endif
9 /*
10  * Class:      probajni_ProbaJNI
11  * Method:     helloC
12  * Signature:  ()V
13  */
14 JNIEXPORT void JNICALL Java_probajni_ProbaJNI_helloC
15     (JNIEnv *, jobject);
16 #ifdef __cplusplus
17 }
18 #endif
19 #endif
```

Odgovarajuća datoteka zaglavlja programskog jezika C.

Primjer implementacije odgovarajuće C funkcije.

```
1 #include "probajni_ProbaJNI.h"
2 JNIEXPORT void JNICALL Java_probajni_ProbaJNI_helloC(JNIEnv
   *env, jobject javaobj)
3 {
4     printf("Hello World: From C");
5     return;
6 }
```

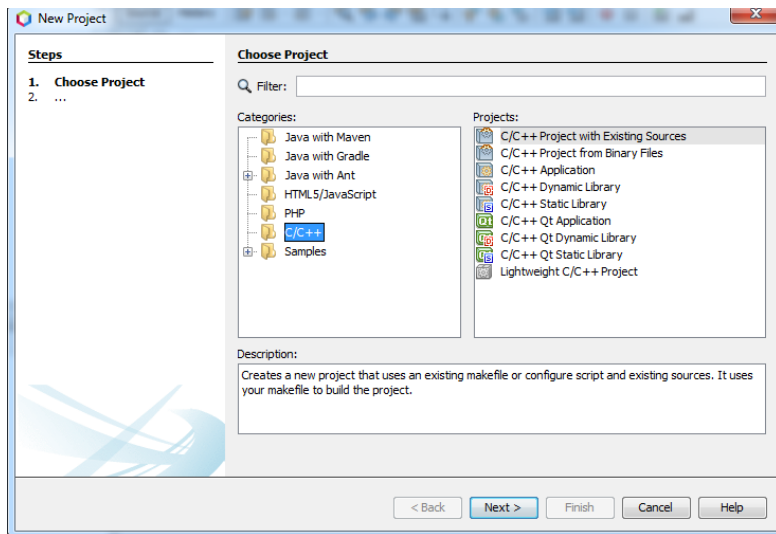
Implementacija C funkcije.

Moguće napraviti na više načina:

- Unutar *Apache NetBeans*-a (zahtjeva dodatno instaliranje *Java 8* (zbog toga je komplicirano). Uz ponešto muke **možda** proradi i bez *Java 8* ukoliko se kopira datoteka `unpack200.exe` (može se posebno skinuti) u `bin` folder *Java* instalacije.
- Korištenjem *CodeBlocks*-a.

U *Apache NetBeans*-u treba izabrati `Tools -> Plugins -> Settings -> Netbeans 8.2 plugin portal -> Available plugins -> Check for newest -> C/C++ -> install` . Ukoliko to prođe, unutar `File->NewProject` trebao bi biti dostupan prozor kao na slici. Treba izabrati `C/C++ Dynamic Library`.

Kreiranje dinamičke C biblioteke



Kreiranje dinamičke C biblioteke

Treba instalirati MingW prevodioc za **odgovarajuću arhitekturu**. **Pazite, ukoliko je JVM 64-bit a MingW 32-bit, program neće raditi!** Mingw je standardno 32-bitan, za 64-bitne arhitekture treba instalirati MingW-w64¹.

Nakon što je kreiran C/C++ projekt treba Project Properties -> C++ Compiler Options, dodati u putanju sljedeće: jdk/include i jdk/include/win32.

Project Properties -> C++ Compiler Options -> Additional Properties dodati -shared -m32 -Wl,-add-stdcall-alias i -static-libgcc -static-libstdc++.

Dodati dokumente zaglavlja i implementaciju u projekt i kreirati dinamičku biblioteku opcijom Build.

¹<http://mingw-w64.org/doku.php>

Kreiranje dinamičke C biblioteke

U *CodeBlocks*-u File->New->Project (izabрати Dynamic Link library). Kopirati dokument zaglavlja i implementacije.

Unutar BuildOptions projekta treba napraviti: BuildOptions -> SearchDirectories -> Compiler/Linker -> dodati putanju do java/jdk/include i java/jdk/include/win32.

Sljedeći korak je postavljanje odgovarajućeg prevodioca (*CodeBlocks* standardno dolazi s 32-bitnim GNU GCC compiler-om.)

Settings -> Compiler -> Toolchain executables -> namjestiti sve sa MinGW64

Kreiranje dinamičke C biblioteke

The screenshot displays the Code::Blocks IDE interface. The main window shows the source code for `main.cpp` with the following content:

```
1 #include "probajni_ProbaJNI.h"
2
3 JNIEXPORT void JNICALL Java_probajni_ProbaJNI_helloC(JNIEnv *env, jobject javaobj)
4
5
6
7
```

The **Project build options** dialog is open, showing the following configuration:

- Selected compiler: MinGW 64-bit
- Policy: Append target options to project options
- Compiler settings: Compiler, Linker, Resource compiler
- Include paths: `..\..\..\..\Program Files\Java\jdk-15.0.1\include`, `..\..\..\..\Program Files\Java\jdk-15.0.1\include\win32`

The **Build log** at the bottom shows the following output:

```
<Program Files\Java\jdk-15.0.1\include" -I"..\..\..\..\Program Files\Java\jdk-15.0.1\include\win32" obj\Debug\main.o -o bin\Debug\HelloWorld.dll -
luser32
Output file is bin\Debug\HelloWorld.dll with size 86.12 KB
Process terminated with status 0 (0 minute(s), 4 second(s))
0 error(s), 0 warning(s) (0 minute(s), 4 second(s))
```

Kreiranje dinamičke C biblioteke

The screenshot displays the Code::Blocks IDE interface with the 'Global compiler settings' dialog box open. The dialog is titled 'Global compiler settings' and features a sidebar with icons for 'Global compiler settings', 'Profiler settings', and 'Batch builds'. The main area is divided into several sections:

- Selected compiler:** MinGW 64-bit (with buttons for 'Set as default', 'Copy', 'Rename', 'Delete', and 'Reset defaults').
- Compiler's installation directory:** C:\Program Files\mingw-w64\x86_64-8.1.0-win32-seh-rt_v6-rv0\mingw64 (with an 'Auto-detect' button and a note: 'NOTE: All programs must exist either in the "bin" sub-directory of this path, or in any of the "Additional paths"...').
- Program Files:** A table with two columns: 'Program Files' and 'Additional Paths'.

Program Files	Additional Paths
C compiler:	x86_64-w64-mingw32-gcc.exe
C++ compiler:	x86_64-w64-mingw32-g++.exe
Linker for dynamic libs:	x86_64-w64-mingw32-g++.exe
Linker for static libs:	ar.exe
Debugger:	GDB/CDB debugger : Default
Resource compiler:	windres.exe
Make program:	mingw32-make.exe

At the bottom of the dialog are 'OK' and 'Cancel' buttons. The background shows the IDE's workspace with a project named 'HelloWorld' and a file named 'main.cpp'.

Kreiranje dinamičke C biblioteke iz terminala (UNIX)




Konfiguracija iz **praktikuma na fakultetu**:



- 1 putanja i poziv javac: `../../../../../opt/jdk-15.0.2/bin/./javac`
- 2 Kreiranje dinamičke biblioteke: `gcc main.c -I/opt/jdk-15.0.2/include -I/opt/jdk-15.0.2/include/linux -c -fPIC`
`gcc main.o -shared -o main.so`
- 3 Dodati: Run ->VM Options:
`-Djava.library.path=/home/student1/username/NetBeansProjects/ProbaJNI/` (putanja do direktorija projekta, pripazite nalazi li se vaš račun unutar direktorija student1 – modificirajte po potrebi)
- 4 Ime library-a: `liblme.so` (da bi se unutar Jave koristilo ime `lme`)

Izlaz Java programa

```
10 | * @author Matej
11 | */
12 | public class ProbaJNI {
13 |
14 |     public native void helloC();
15 |
16 |     static {
17 |         System.loadLibrary("HelloWorld");
18 |     }
19 |
20 |
21 |     /**
22 |      * @param args the command line arguments
23 |      */
24 |     public static void main(String[] args) {
25 |         // TODO code application logic here
26 |
27 |         new ProbaJNI().helloC();
28 |     }
29 |
```

probajni.ProbaJNI >

Test Results	Output - ProbaJNI (run) %
	compile:
	run:
	Hello World: From CBUILD SUCCESSFUL (total time: 13 seconds)

Instalacija programskog jezika R i odgovarajućeg okruženja

- Instalirati programski jezik R ².
- Opcionalno instalirati okruženje *RStudio*³. Olakšava rad u R -u.
- Instalirati paket *rJava* unutar R -a koristeći naredbu `install.packages("rJava")`.
- Skinuti paket *rJava*⁴.
- *Java* projektu unutar kojega će se pozivati R , dodati biblioteke `rJava\jri\{JRI.jar, JRIEngine.jar, REngine.jar}`.
- U putanju (pod *Windowsom* unutar *Environmental variables* dodati: `rJava\jri\x64`, `rJava\libs\x64`, `R\R-4.0.5\bin\x64` (primjer je za 64-bitnu arhitekturu). Bit će potrebno ponovo pokrenuti *Apache NetBeans*.

²<https://www.r-project.org/>

³<https://www.rstudio.com/>

⁴<https://cran.r-project.org/web/packages/rJava/index.html>

Računanje centralne vrijednosti vektora koristeći R

```
1 package rinjava;
2 import org.rosuda.JRI.Engine;
3 import org.rosuda.JRI.REXP;
4
5 public class RinJava {
6     public static void main(String[] args) {
7         String vektor = "c(1,2,3,4,5)"; //R vektor unutar Java
8         //stringa
9         Engine engine = Engine.getMainEngine(); //dohvacamo R
10        //za racunanje
11        if(engine == null){ //ukoliko R instanca nije pokrenuta
12            engine=new Engine (new String [] {"--vanilla"},
13            false, null); //stvaramo novu instancu
14            if (!engine.waitForR())
15                {
16                    System.out.println ("R se ne moze ucitati");
17                    return; //doslo je do greske
18                }
19        }
20    }
```

Poziv R-funkcija iz Java.

Računanje centralne vrijednosti vektora koristeći R

```
1 engine.eval("rVektor<-" + javaVector); //ucitamo vektor u R
2   . Java reprezentacija tog vektora je string R naredbe.
3   //Racunamo centralnu vrijednost
4   engine.eval("medianVr<-median(vektor)");
5
6   //Dohvacamo centralnu vrijednost
7   double median = engine.eval("medianVr").asDouble();
8
9   //Ispis povratne vrijednosti
10  System.out.println("Centralna vrijednost vektora je" +
11  median);
12  engine.end(); //gasimo R instancu (jako bitno!)
13 }
```

Poziv R-funkcija iz Jave.

- Skinuti najnoviju verziju *SQLite JDBC* (Java database connectivity)⁵.
- Dodati `sqlite-jdbc.jar` kao biblioteku Java aplikaciji koja koristi *SQLite* bazu podataka.
- Skinuti `sqlite-tools`⁶. Omogućava pregledavanje stvorene baze, ali i brzo kreiranje baze iz komandne linije korištenjem naredbe `sqlite3`.

⁵<https://mvnrepository.com/artifact/org.xerial/sqlite-jdbc>

⁶<https://www.sqlite.org/download.html>