**Khởi tạo một chương trình JNI theo các bước sau :**

* 1. Viết Class java cho phép giao tiếp với C/C++.
  2. Biên dịch Class Java thành C/C++ Header.
  3. Chương trình C/C++ include java.h.
  4. Tạo các thủ tục giao tiếp C & Java.
  5. Biên dịch chương trình .cpp thành .dll.
  6. Khai báo các thông số cho chương trình java (Host Environment)
  7. Chạy chương trình Java với các thông số được thiết lập .

**Các kiểu dữ liệu cơ bản :**

* Việc tryền và quản lí các biến trong JNI khá phức tạp . Nguyên nhân là   
  do cấu trúc xây dựng và quản lí biến giữa 2 ngôn ngữ C/C++ và Java   
  không tương đồng .
* Để biểu diễn kiểu dữ liệu chung JNI tạo ra một vài kiểu dữ liệu trung   
  gian như sau :
  + jobject jboolean jbyte
  + jchar jshort jint
  + jlong jstring jfloat
  + jdouble …

1. **Numeric Parameters**

* Giữa Java và C có những kiểu dữ liệu đặc trưng khác nhau. Ví dụ trong C kiểu int có 2 nền tảng là 16bit và 32bit, còn trong java kiểu int luôn là 32bit. Do đó JNI định nghĩa ra các kiểu dữ liệu jint, jlong …
* Sự tương ứng trong kiểu dữ liệu:

|  |  |  |
| --- | --- | --- |
| JAVA programming | C programming | Bytes |
| boolean | jboolean | 1 |
| byte | jbyte | 1 |
| char | jchar | 2 |
| short | jshort | 2 |
| Int | jint | 4 |
| long | jlong | 8 |
| float | jfloat | 4 |
| double | jdouble | 8 |

* Trong file jni.h cũng định nghĩa được các typedef, JNI\_TRUE=1, JNI\_FALSE=0

1. **String Parameters**

* String trong các ngôn ngữ Java là trình tự các điểm mã UTF-16 trong khi ở ngôn ngữ C, String được chấm dứt bởi byte null.
* String trong jstring (JNI), là một chuỗi unicode 16 bit. Trong C String được xây dựng mặc định từ 8 ký tự bit. Vì vậy, để truy cập vào một đối tượng String trong Java thông qua một C/C++, hoặc trả lại một chuỗi C/C++ cho Java, ta cần phải sử dụng chức năng chuyển đổi JNI trong việc thực hiện.
* JNI tạo ra phương thức để thao tác chuyển đổi chuỗi jstring. Jstring không được sử dụng như 1 string bình thường trong C mà phải qua các phương thức trong JNI
* Jstring có các phương thức hỗ trợ căn bản sau:

|  |  |
| --- | --- |
| * **GetStringChars** * **ReleaseStringChars** | Lấy chuỗi String thường |
| * **GetStringUTFChars** * **ReleaseStringUTFChars** | Lấy ra chuỗi String UTF |
| * **GetStringLength** * **GetStringUTFLength** | Trả về độ dài chuỗi String |
| * **NewString** * **NewStringUTF** | Tạo chuỗi String mới |
| * **GetStringCritical** * **ReleaseStringCritical** | Lấy chuỗi String từ con trỏ truyền vào |

1. **Array**

* JNI xử lý *các* mảng nguyên thủy và các mảng đối tượng khác nhau. Mảng nguyên thủy có chứa các yếu tố là các loại nguyên thủy như int và boolean. Mảng đối tượng chứa các yếu tố là các loại object chẳng hạn như trường hợp class và các mảng 2 chiều.
* JNI sử dụng jarray để tao tác với mảng
* Jarray không phải là 1 loại mảng trong C, vì vậy phải sử dụng các phương thức jarray trong JNI thích hợp để truy cập mảng.
* Các phương thức hỗ trợ căn bản:

|  |  |
| --- | --- |
| * **Get<Type>ArrayRegion** * **Set<Type>ArrayRegion** | Lấy ra mảng từ 1 mảng căn bản |
| * **Get<Type>ArrayElements** * **Release<Type>ArrayElement** | Nhận và hủy 1 phần tử mảng |
| * **GetArrayLength** | Trả về số phần tử mảng |
| * **New<Type>Array** | Tạo mảng mới |
| * **GetPrimitiveArrayCritical** * **ReleasePrimitiveArrayCritical** | Nhận và hủy 1 mảng từ biến con trỏ gửi vào |

1. **JNA**

* **JNA**: Java Native Access là một phần mở rộng của Java cho phép sử dụng các API bao gồm các tập tin thư viện động DLL trong Windows.   
  Không giống như JNI, nó không yêu cầu để tạo ra mã bằng cách sử dụng chức năng C. Để sử dụng chúng, chỉ đơn giản sử các tập tin có định nghĩa chúng và tuyên bố tiêu đề của các chức năng này trong một giao diện.
* **JNA** cung cấp các chương trình Java để dễ dàng truy cập đến các thư viện được chia sẽ mà không cần sử dụng JNI . JNA thiết kế nhằm mục đích để cung cấp các truy cập một cách tự nhiên, ít tốn công sức.
* Các thư viện yêu cầu sử dụng
  + import com.sun.jna.Library;
  + import com.sun.jna.Native;
  + import com.sun.jna.Platform;
* Demo

// JNA Demo. Scriptol.com   
 package CInterface;   
 import com.sun.jna.Library;   
 import com.sun.jna.Native;   
 import com.sun.jna.Platform;

public class hello   
 {   
 public static void main(String[] args)   
 {   
 String mytext = "Hello World!";   
 if (args.length != 1)   
 {   
 System.err.println("You can enter you own text between quotes...");   
 System.err.println("Syntax: java -jar /jna/dist/demo.jar \"myowntext\"");   
 }   
 else   
 mytext = args[0];

// Library is c for unix and msvcrt for windows   
 String libName = "c";   
 if (System.getProperty("os.name").contains("Windows"))   
 {   
 libName = "msvcrt";   
 }

// Loading dynamically the library   
 CInterface demo = (CInterface) Native.loadLibrary(libName, CInterface.class);   
 demo.puts(mytext);   
 }   
 }