A Study of Unspecified Cases in the JNI Specification

Sungjae Hwang
sjhwang87@kaist.ac.kr
KAIST
South Korea

Sungho Lee
eshaj@cnu.ac.kr
CNU
South Korea

Jihoon Kim
kjh618@kaist.ac.kr
KAIST
South Korea

Sukyoung Ryu
sryu.cs@kaist.ac.kr
KAIST
South Korea

February 5, 2021

1 List of Segmentation Faults

1.1 Missing Type Check

Call methods with a wrong receiver object:
Calling a private function with java object that does not consist of the private function through `call<type>method`, `call<type>methodV`, `call<type>methodA`.

Define classes with a non classloader object:
Passing non classloader object when calling `DefineClass`.

Call reflect methods with a non reflect type object:
Passing non `java.lang.reflect.Method` or `java.lang.reflect.Constructor` when calling `FromReflectedMethod`, and `FromReflectedField`.

Release unrelated strings:
Release a string by calling `releaseCharArrayElements` which is not obtained by `getCharArrayElements`.

1.2 Missing Null Check

Call methods with null arguments:
Missing null check of jvalue when calling `Call<type>methodA`, `CallNonvirtual<type>MethodA`, `CallStatic<type>methodA`.

Obtain/Set static fields with a null field ID:
Missing null check of static jfieldID when calling `getStatic<type>Field`, `setStatic<type>field`

Obtain strings with a null destination buffer:
Missing null check of destination buffer when calling `GetStringUTFRegion`, `GetStringRegion`

Generate new strings with a null source buffer:
Missing null check of source string when calling `NewString`

Call methods with a null method ID:
Missing null check of jmethodID when calling `call<type>method`, `call<type>methodV`, `call<type>methodA`

Define classes with a name as null value:
Missing null check of name of the class when calling `DefineClass`

Obtain fields and methods ID with a signature as null value:
Missing null check of field signature when calling `GetFieldID`, `GetStaticFieldID`, and missing null check of method signature when calling `GetStaticMethodID`, `GetMethodID`.

Obtain fields with a null receiver object:
Missing null check of jobject when calling `GetObjectField`.

Obtain field/method IDs and class from a null object:
Missing null check of jobject when calling `FromReflectedField`, `GetObjectClass`, `FromReflectedMethod`

2 Differences between JVMs

2.1 Missing Type Check

Define classes with a non classloader object:
Passing non-classloader object when calling `DefineClass`

Obtain/update non-static fields with a static field ID:
Passing non-static field ID when calling `GetStatic<type>Field`,and `SetStatic<type>Field`.

Obtain/update static fields with a non-static field ID

Passing static field ID when calling `Get<type>Field`,and `Set<type>Field`.

Obtain classes with a bad class descriptor
Passing field signature as a class name when calling `FindClass`.

Throw exceptions with non-throwable objects
Passing non-throwable object when calling `ThrowNew`.

Call a private function with a receiver object that does not consist of the private function
Call Java function with wrong type of method ID. e.x. Calling `CallBooleanMethod` with jmethod ID of Int return type function.

Create Object with array classes
Passing array class when calling `NewObject`.

Create Object with a non-constructor method ID
Passing non-constructor method ID when calling `NewObject`.

Obtain field IDs with not subtype of java/lang/reflect/Field object
Passing non-java.lang.reflect.field type object when calling `FromReflectedField`.

Call methods with a wrong type of receiver
Calling a private function with java object that does not consist of the private function through `call<type>method`, `call<type>methodV`, `call<type>methodA`.

Release unrelated array elements
Release a string by calling releaseCharArrayElements which is not obtained by getCharArrayElements.

2.2 Missing Null Check

Call methods with a null method ID
Passing null jmethodID when calling `Call<type>Method`.

Create Objects with a null method ID
Passing null jmethodID when calling `NewObject`.

Obtain fields with a null receiver object
Passing null jboject when calling `Get<type>Field`.

Call methods with a null receiver object
Second argument of `Call<type>Method` is null.

Obtain field IDs with a null reflected object
Second argument of `FromReflectedField` is null.

Obtain method IDs with a null reflected object
Second argument of `FromReflectedMethod` is null.

Obtain field IDs with a field signature as null value
Forth argument of `GetFieldID` is null.

Obtain field IDs with a field name as null value
Third argument of `GetFieldID` is null.

Obtain method IDs with a method signature as null value
Forth argument of `GetMethodID` is null.

Obtain method IDs with a method name as null value
Third argument of `GetMethodID` is null.

Release null string Third argument of `ReleaseStringChars` is null.

Obtain object from a non-existing local frame on the stack
Get local reference object when there is no local frame on stack by calling `PopLocalFrame`

2.3 Missing Garbage Check

Obtain a reference type of deleted objects
Passing deleted object as second argument of `GetObjectRefType`.

Store deleted objects into an array element
Passing deleted object as forth argument of `SetObjectArrayElement`.

2.4 Missing Negative Integer Check

Create negative capacity of a local reference frame
Passing negative number as second argument of `EnsureLocalCapacity`.

Access negative index of array elements
Passing negative number as third argument of `GetByteArrayRegion`, and `SetCharArrayRegion`.