FFI Internal Documentation

Eliot Miranda
2007

This document describes the Alien FFI, a minimal foreign function interface for Newspeak and/or Squeak on IA32 (Intel x86) platforms, from the implementor's perspective. Please read the external documentation for an overview before reading this.

Implementation Architecture

The system has two main components. In Smalltalk the Alien class hierarchy provides the bulk of the system to the client. Aliens are byte data directly interpreted by various primitives. The primitives all reside in a Squeak plugin (currently external, but could be made internal for improved performance) "IA32ABI" (IA32ABI.bundle IA32ABI.dll IA32ABI.so).

The plugin is composed of a set of primitives written in Slang (Smalltalk class name IA32ABIPlugin) and a small amount of support machinery (5 functions) written in C.

Wrangling Aliens

An Alien is at least 8 bytes in length. Its first 4 bytes are a size field which defines the kind of Alien and its bounds:

sizeField > 0 Direct Alien
The data follows the 4 byte sizeField and is of length sizeField byte.

sizeField < 0 Indirect Alien
A pointer occupies the 4 bytes following the sizeField (the Alien's Smalltak size is 8) which points to data on the C heap (typically obtained by calloc(3)). The data is assumed to be -sizeField bytes long.

sizeField = 0 Pointer Alien
A pointer occupies the 4 bytes following the sizeField (the Alien's Smalltak size is 8) which points to data on the C heap. The Alien has no bounds.

Class Alien is registered with the VM in the specialObjectsArray to allow the VM to identify Aliens and reject non-Aliens (see SystemDictionary>>recreateSpecialObjectsArray). In Slang access to Aliens is via support functions in IA32ABIPlugin private-support:

IA32ABIPlugin private-support
sizeField: rcvr
```
"Answer the first field of rcvr which is assumed to be an Alien of at least 8 bytes"
self inline: true.
^self longAt: rcvr + BaseHeaderSize
```

startOfData: rcvr "<Alien oop> ^<Integer>"
```
"Answer the start of rcvr's data. For direct aliens this is the address of
 the second field. For indirect and pointer aliens it is what the second field points to."
self inline: true.
```

```
        (self sizeField: rcvr) > 0
                ifTrue: [^rcvr + BaseHeaderSize + BytesPerOop]
                ifFalse: [^self longAt: rcvr + BaseHeaderSize + BytesPerOop]

startOfData: rcvr "<Alien oop>" withSize: sizeField "<Integer> ^<Integer>"
        "Answer the start of rcvr's data.  For direct aliens this is the address of
         the second field.  For indirect and pointer aliens it is what the second field points to."
        self inline: true.
        sizeField > 0
                ifTrue: [^rcvr + BaseHeaderSize + BytesPerOop]
                ifFalse: [^self longAt: rcvr + BaseHeaderSize + BytesPerOop]

isAlien: anOop
        self export: true.
        ^interpreterProxy
                includesBehavior: (interpreterProxy fetchClassOf: anOop)
                ThatOf: interpreterProxy classAlien
```

In the support machinery equivalent code is defined via C macros:

```
#define isIndirect(alien) (sizeField(alien) < 0)
#define startOfParameterData(alien) (isIndirect(alien)  \
                                     ? *(void **)dataPtr(alien)  \
                                     :  (void  *)dataPtr(alien))
#define isIndirectSize(size) ((size) < 0)
#define startOfDataWithSize(alien,size) (isIndirectSize(size)   \
                                 ? *(void **)dataPtr(alien)      \
                                 :  (void  *)dataPtr(alien))
```

All access primitives are defined in Slang. e.g.

IA32ABIPlugin primitives-accessing

```
primAddressField
        "Answer the unsigned 32-bit integer comprising the address field (the second 32-bit
field)."
        "<Alien> primAddressField ^<Integer>
                <primitive: 'primAddressField' error: errorCode module: 'IA32ABI'>"
        | rcvr value valueOop |
        self export: true.
        rcvr := interpreterProxy stackValue: 0.
        value := self longAt: rcvr + BaseHeaderSize + BytesPerOop.
        valueOop := interpreterProxy positive32BitIntegerFor: value.
        ^interpreterProxy pop: 1 thenPush: valueOop
```

Many of the accessing primitives use a support function to bounds-check an access to an Alien. Since pointer Aliens are without bounds the function succeeds for zero size.

IA32ABIPlugin private-support
index: byteIndex **length:** length **inRange:** rcvr
	"Answer if the indices byteIndex to byteIndex + length - 1 are valid zero-relative indices into the rcvr."
	| dataSize |
	self inline: true.
	dataSize := self sizeField: rcvr.
	^self
		cCode: 'dataSize == 0 || (unsigned long)byteIndex <= abs(dataSize)'
		inSmalltalk: [dataSize = 0 or: [byteIndex > 0 and: [byteIndex + length <= dataSize abs]]]

Call-outs
Here-ish

Coda
ASHE: You still don't know what you're dealing with do you? Perfect organism. Its structural perfection is matched only by its hostility.
RIPLEY: You admire it.
ASHE: I admire its purity, its sense of survival; unclouded by conscience, remorse, or delusions of morality.
RIPLEY: I've heard enough and I'm asking you to pull the plug.
ASHE: One more word..... I can't lie to you about your chances, but... you have my sympathies.
Alien, dir. Ridley Scott, 1979