Системы верификации байткода в JVM

Николай Иготти

Функциональность верификатора

- Требования к .class-файлу чётко описаны стандартом Java – верификатор проверяет это соответствие
- Является важной частью системы безопасности
- Предоставляет гарантии исполняющей компоненте
- В общем случае верификатор должен решать NP-полную задачу
- Реальные решения – компромисс
- Доказывать или проверять?

Фазы работы верификатора

- Проверка формата классфайла (длина, magic, базовая структура)
- Линковка класса (пул констант, иерархия наследования, использование final)
- Линковка метода (верность инструкций байткода, обработка исключений, обработка jsr/ret, *семантическая корректность*)
- Вызов метода или первое исполнение (наличие, сигнатурная совместимость, контроль доступа)

Формат классфайла

u4 MAGIC (== 0xCAFEBABE)
u2 MINOR VERSION
u2 MAJOR VERSION
u2 constant_pool_count
cp_info constant_pool[constant_pool_count-1]
u2 access_flags
u2 this_class
u2 super_class
u2 interfaces_count
u2 interfaces[interfaces_count]
u2 fields_count
field_info fields[fields_count]
u2 methods_count
method_info methods[methods_count]
u2 attributes_count
attribute_info attributes[attributes_count]

Пул констант

u1 tag	u1 info[ ]

Тип	Значение
CONSTANT_Class	7
CONSTANT_Fieldref	9
CONSTANT_Methodref	10
CONSTANT_InterfaceMethodref	11
CONSTANT_String	8
CONSTANT_Integer	3
CONSTANT_Float	4
CONSTANT_Long	5
CONSTANT_Double	6
CONSTANT_NameAndType	12
CONSTANT_Utf8	1

Описание метода

- Доступ (маска ACC_PUBLIC , ACC_PRIVATE ACC_PROTECTED, ACC_STATIC, ACC_FINAL, ACC_SYNCHRONIZED, ACC_NATIVE , ACC_ABSTRACT ACC_STRICT)
- Сигнатура (параметры, возвращаемое значение) – строки Utf8, например *[[IL*
- Список атрибутов (включая предопределённые: Code, ConstantValue, Exceptions, InnerClasses, Synthetic и т.п.)
- Code содержит информацию о количестве локалов, необходимом стеке, обработчиках исключений, байткод метода (4 байта))

Семантические инварианты проверяемые верификатором

- *Корректность стековых операций (у Java нетипизированный стек)*
- *Совместимость типов при записи в поля класса*
- *Доступ к локалам (локалы тоже нетипизированные)*
- *Использование методов в соответствии с сигнатурой (количество, тип)*
- *Доступ к методам и полям других классов*
- *Исполнение try/catch/finally блоков*