How to Implement A Simple Dalvik Virtual Machine

Agenda

- Java Virtual Machine (JVM)
  - Java Virtual Machine and its instructions
  - Implement a Simple JVM
- Dalvik Virtual Machine (DVM)
  - Dalvik Virtual Machine and its instructions
  - Implement a Simple DVM
- References

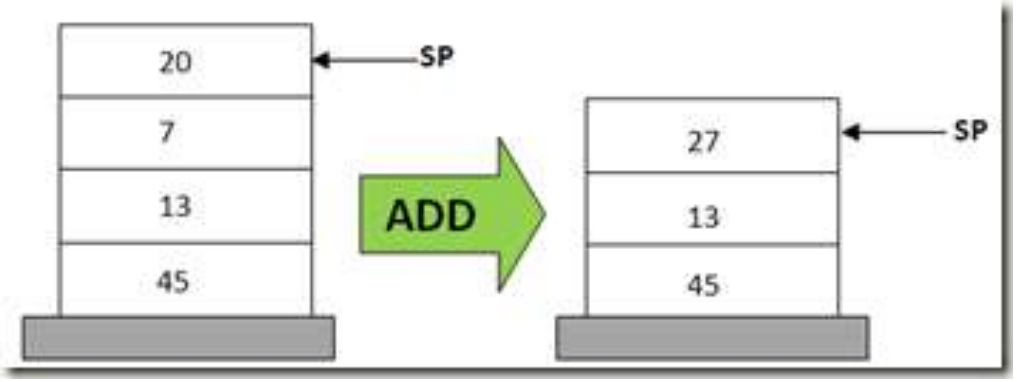
Java Virtual Machine

Java Virtual Machine Overview

- Java Virtual Machine
  - JVM Model
  - Java ByteCode
  - Java ByteCode instructions
- How to make a Java VM
  - A Simple Java Virtual Machine
  - Experiment

Java Virtual Machine

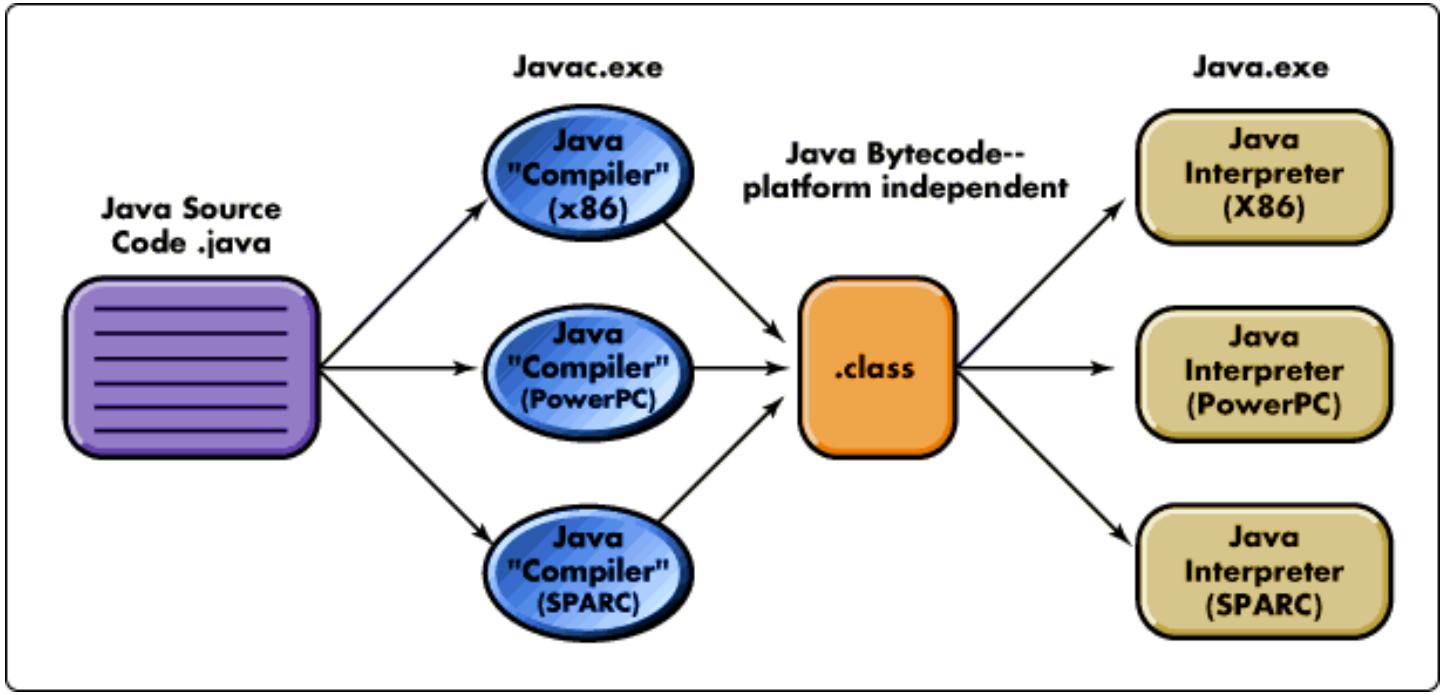
- Stack-based (Last-In First-Out) Virtual Machine
- Computation in Stack
- Load Java **ByteCode** to execute program



Lines	Stack-based VM Pseudo Code
0	POP 20
1	POP 7
2	ADD 20, 7, result
3	PUSH result

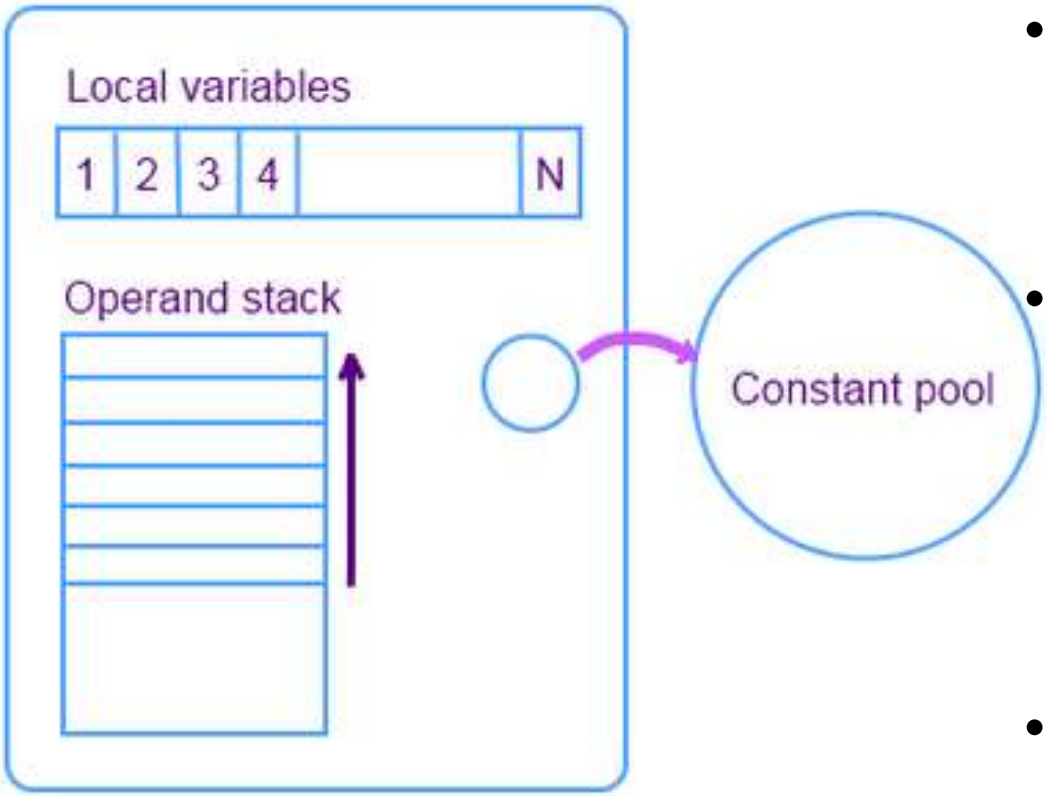
http://www.codeproject.com/Articles/461052/Stack-based-vs-Register-based-Virtual-Machine-Arch

Java Source to ByteCode



http://javabook1.blogspot.tw/2013/07/introduction-to-java.html

JVM Model



- Local Variables:
  - place the method input parameters
- Operand Stack:
  - Computation Area
  - Put Instruction Operands and Return address
- Constant Pool
  - Put Constant Data

Java ByteCode

- What is ByteCode ?
  - also known as **p-code** (portable code), is a form of instruction set designed for efficient execution by **a software interpreter**.

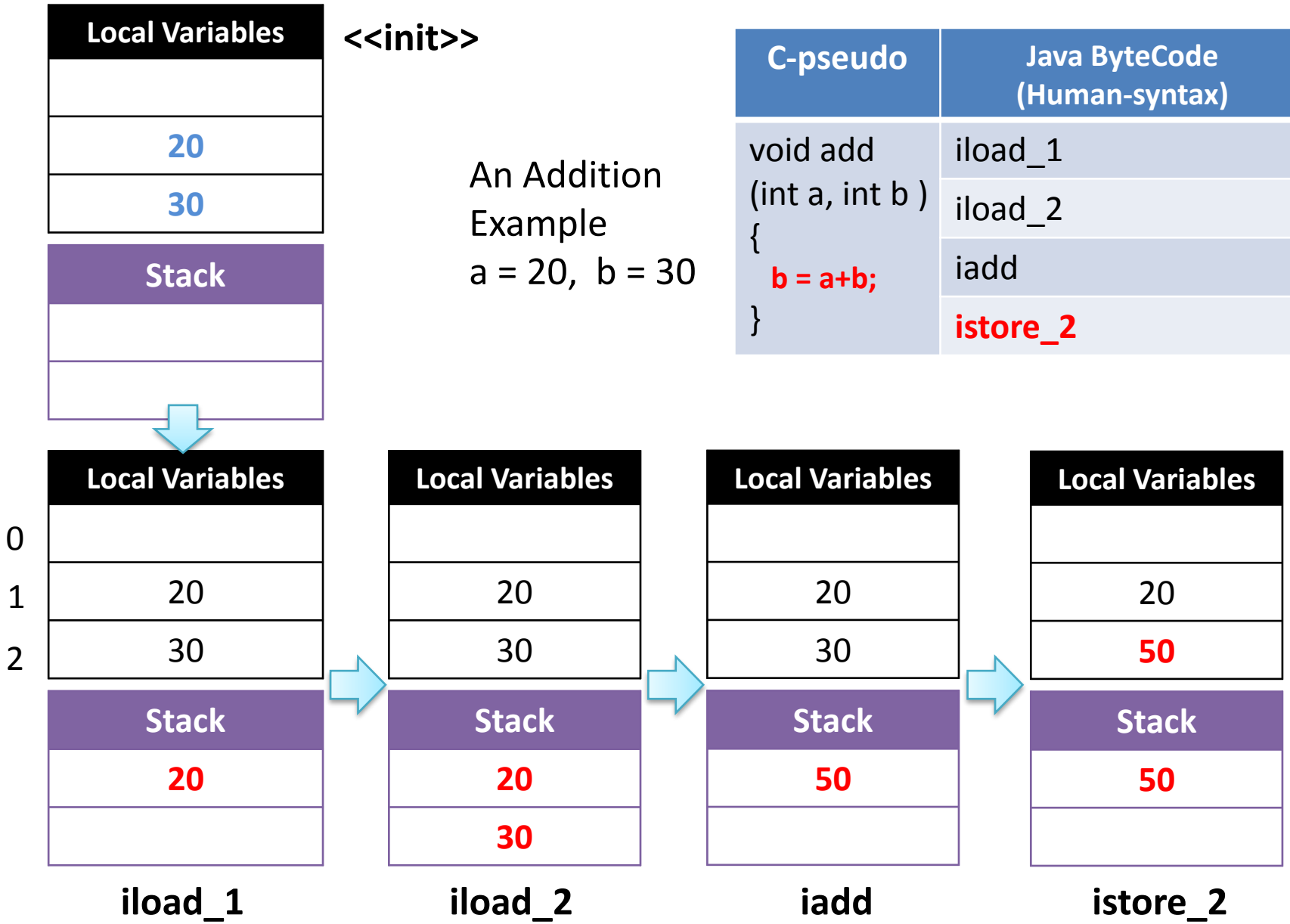
An Java Addition Example a = 20, b = 30

C-pseudo	X86 ASM	Java ByteCode (Human-syntax)	Java ByteCode binary

int add
(int a, int b)
{
return a+b;
}
	mov eax, byte [ebp-4]	iload_1	0x1a

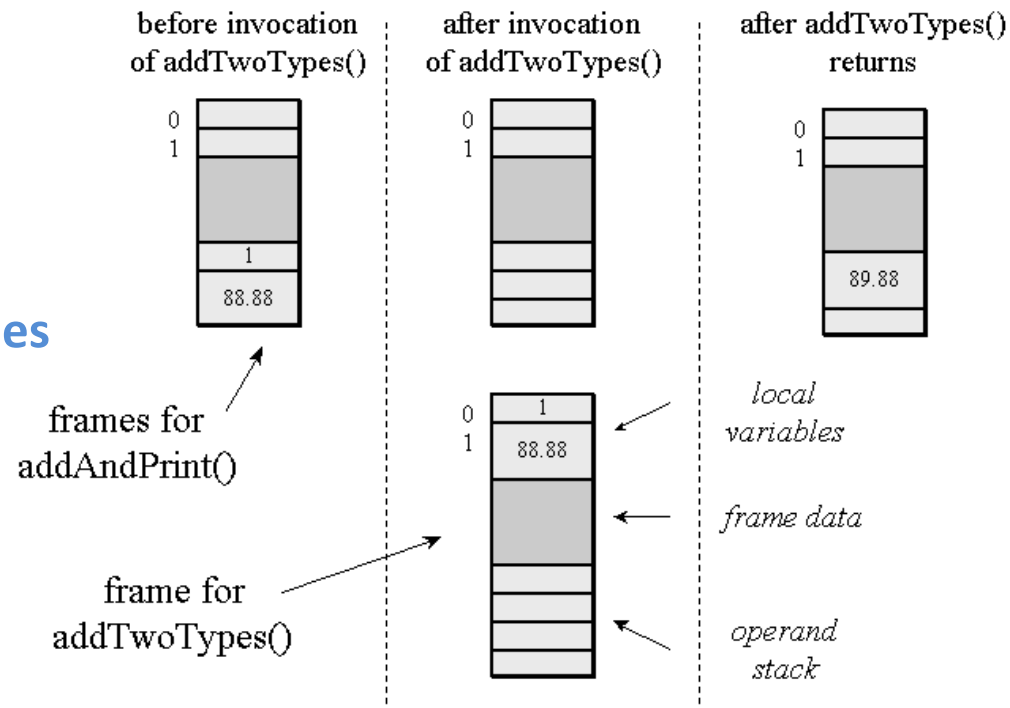
	mov edx, byte [ebp-8]	iload_2	0x1b
	add eax, edx	iadd	0x60
	ret	ireturn	0x3e

A Java Addition Example



More Java ByteCode Example

```
class Example3c {
   public static void addAndPrint() {
      double result = addTwoTypes
                              (1, 88.88);
      System.out.println(result);
   }
   public static double addTwoTypes
   (int i, double d) {
      return i + d;
   }
}
```



Inside the Java Virtual Machine, 2000, Bill Venners

Java Bytecode instructions (Partials)

Mnemonic	Opcode	Stack
iadd	0x60	Pop value1, Pop value2 result = value1 + value2 Push result
isub	0x64	Pop value1, Pop value2 result = value1 - value2 Push result
idiv	0x6C	Pop value1, Pop value2 result = value2 / value1 Push result
imul	0x68	Pop value1, Pop value2 result = value1 * value2 Push result
irem	0x70	Pop value1, Pop value2 result = value2 % value1 Push result

http://en.wikipedia.org/wiki/Java_bytecode_instruction_listings

How to make a Java Virtual Machine

- At least to know about Java Class File
  - Wikipedia
    - http://en.wikipedia.org/wiki/Java_bytecode
    - http://en.wikipedia.org/wiki/Java_class_file
  - the Java Specification
    - http://docs.oracle.com/javase/6/docs/index.html

Java Class File

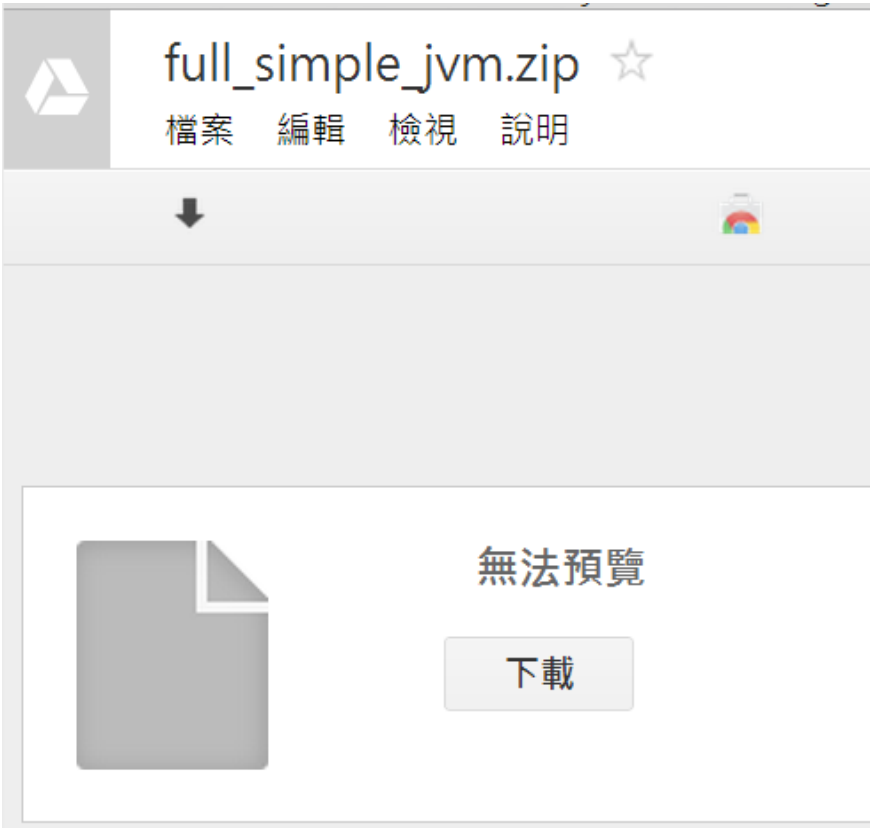
Java Class File Structure	
Magic Number:	0xCAFEBABE
Version of Class File Format:	the minor and major versions of the class file
Constant Pool:	Pool of constants for the class
Access Flags:	for example whether the class is abstract, static, etc.
This Class:	The name of the current class
Super Class:	The name of the super class
Interfaces:	Any interfaces in the class
Fields:	Any fields in the class
Methods:	Any methods in the class
Attributes:	Any attributes of the class (for example the name of the sourcefile, etc.)

Java Class File Structure

```
struct Class_File_Format {
  u4 magic_number;

  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];
}
```

Download Simple JVM

- goo.gl/FA3fwx



Simple JVM Source Code Structure

simple_jvm ▸

說明(H)

共用對象 ▾ 燒錄 新增資料夾

A – H (4)

- Foo.class
- Foo
- foo_dasm
- Foo_dasm2

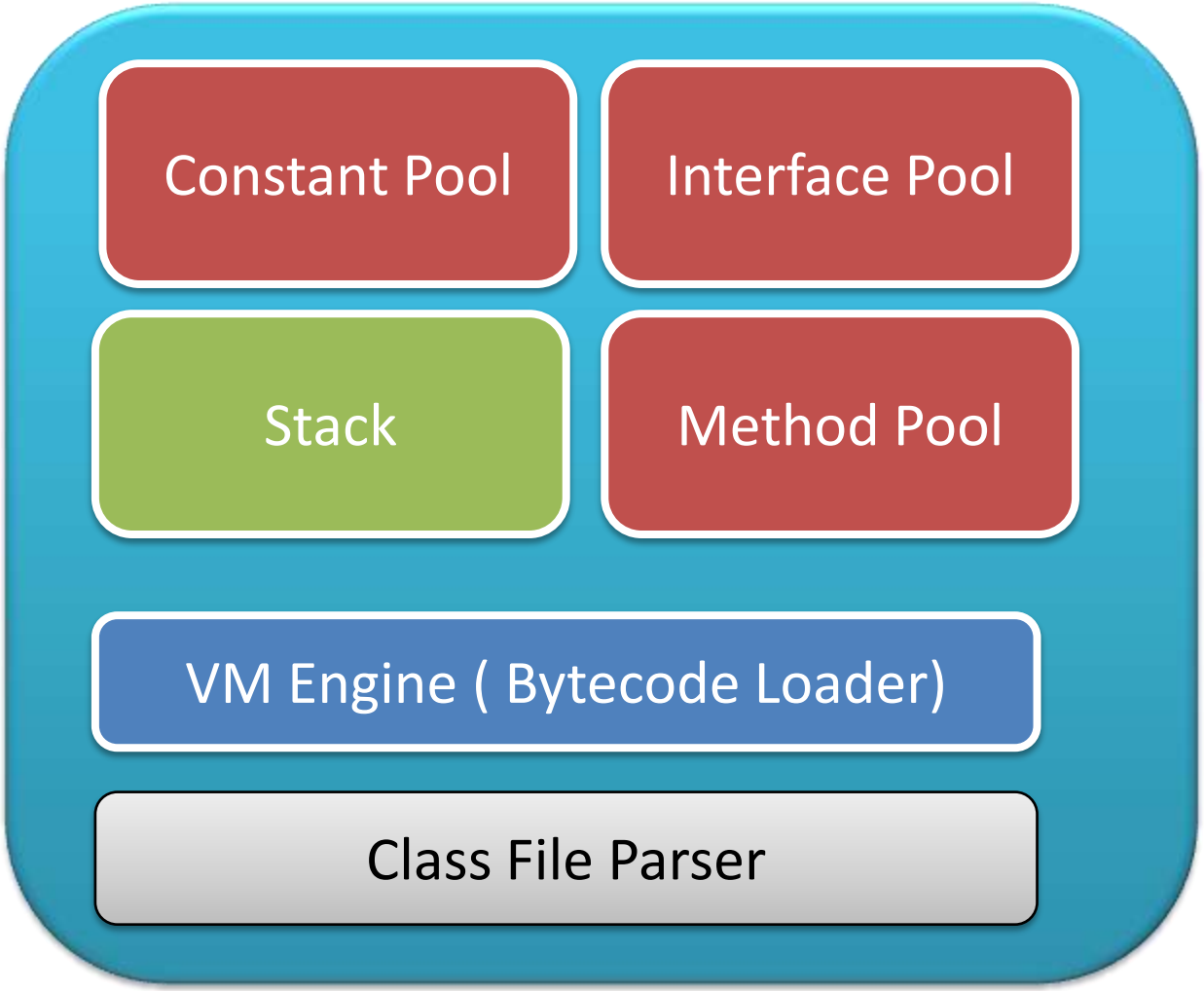
I – P (5)

- mingw
- log
- log.tx
- log
- make

Q – Z (12)

- simple_jvm
- simple_jvm
- simple_jvm_bytecodes
- simple_jvm_class_parser
- simple_jvm_constant_pool_parser
- simple_jvm_debug
- simple_jvm_field_pool_parser
- simple_jvm_free_pool
- simple_jvm_interface_pool_parser
- simple_jvm_main
- simple_jvm_method_pool_parser
- simple_jvm_stack

Simple JVM


Constant Pool
Interface Pool
Stack
Method Pool
VM Engine (Bytecode Loader)
Class File Parser

Compile Simple JVM

```
D:\simple_jvm>make

D:\simple_jvm>echo off
"Make Simpe Java VM"
"Make simple_jvm"
"Make simple_jvm successful"

D:\simple_jvm>
```

Test Foo

```
D:\simple_jvm>java Foo
HelloWorld
5 + 10 = 15
5 * 10 = 50
5 - 10 = -5
5 / 10 = 0
d = 15 + 50 + -5 + 0 = 62
Foo Test By WJY

D:\simple_jvm>
```

Java Foo

```
method attributes_count = 1
method tmp->attribute_name_index = 21
method tmp->attribute_length = 29
method attributes_count = 1
method tmp->attribute_name_index = 21
method tmp->attribute_length = 389
-------------------------------------
Execute Simple JVM
-------------------------------------
-------------------------------------
find and execute <init> method
-------------------------------------
HelloWorld
5 + 10 = 15
5 * 10 = 50
5 - 10 = -5
5 / 10 = 0
d = 15 + 50 + -5 + 0 = 62
Foo Test By WJY
-------------------------------------
Terminate Simple JVM
-------------------------------------

D:\simple_jvm>
```

Simple JVM Foo

```
byteCode byteCodes[] = {
    { "aload_0"        , 0x2A, 1,  op_aload_0        },
    { "bipush"         , 0x10, 2,  op_bipush         },
    { "dup"            , 0x59, 1,  op_dup            },
    { "getstatic"      , 0xB2, 3,  op_getstatic      },
    { "iadd"           , 0x60, 1,  op_iadd           },
    { "iconst_0"       , 0x03, 1,  op_iconst_0       },
    { "iconst_1"       , 0x04, 1,  op_iconst_1       },
    { "iconst_2"       , 0x05, 1,  op_iconst_2       },
    { "iconst_3"       , 0x06, 1,  op_iconst_3       },
    { "iconst_4"       , 0x07, 1,  op_iconst_4       },
    { "iconst_5"       , 0x08, 1,  op_iconst_5       },
    { "idiv"           , 0x6C, 1,  op_idiv           },
    { "imul"           , 0x68, 1,  op_imul           },
    { "invokespecial"  , 0xB7, 3,  op_invokespecial  },
    { "invokevirtual"  , 0xB6, 3,  op_invokevirtual  },
    { "iload"          , 0x15, 2,  op_iload          },
    { "iload_1"        , 0x1B, 1,  op_iload_1        },
    { "iload_2"        , 0x1C, 1,  op_iload_2        },
    { "iload_3"        , 0x1D, 1,  op_iload_3        },
    { "irem"           , 0x70, 1,  op_irem           },
    { "istore"         , 0x36, 2,  op_istore         },
    { "istore_1"       , 0x3C, 1,  op_istore_1       },
    { "istore_2"       , 0x3D, 1,  op_istore_2       },
    { "istore_3"       , 0x3E, 1,  op_istore_3       },
    { "isub"           , 0x64, 1,  op_isub           },
    { "ldc"            , 0x12, 2,  op_ldc            },
    { "new"            , 0xBB, 3,  op_new            },
    { "return"         , 0xB1, 1,  op_return         }
};
```

Simple JVM Instruction Table :

simple_jvm_bytecodes.c

iadd : simple_jvm_bytecodes.c

```c
// iadd
int op_iadd( unsigned char **opCode, StackFrame *stack, SimpleConstantPool *p ) {
    int value1 = popInt(stack);
    int value2 = popInt(stack);
    int result = 0;
    result = value1 + value2;
#if SIMPLE_JVM_DEBUG
    printf("iadd: %d + %d = %d\n",value1, value2, result);
#endif
    pushInt(stack, result);
    *opCode = *opCode + 1;
    return 0;
}
```

iadd	0x60	Pop value1, Pop value2 result = value1 + value2 Push result

imul: simple_jvm_bytecodes.c

```c
// imul
int op_imul( unsigned char **opCode, StackFrame *stack, SimpleConstantPool *p ) {
    int value1 = popInt(stack);
    int value2 = popInt(stack);
    int result = 0;
    result = value1 * value2;
    pushInt(stack, result);
    *opCode = *opCode + 1;
    return 0;
}
```

imul	0x68	Pop value1, Pop value2 result = value1 * value2 Push result

Experiment: add irem instruction into Simple JVM

irem	0x70	Pop value1, Pop value2 result = value2 % value1 Push result

goo.gl/xIMuym

Execution Result:

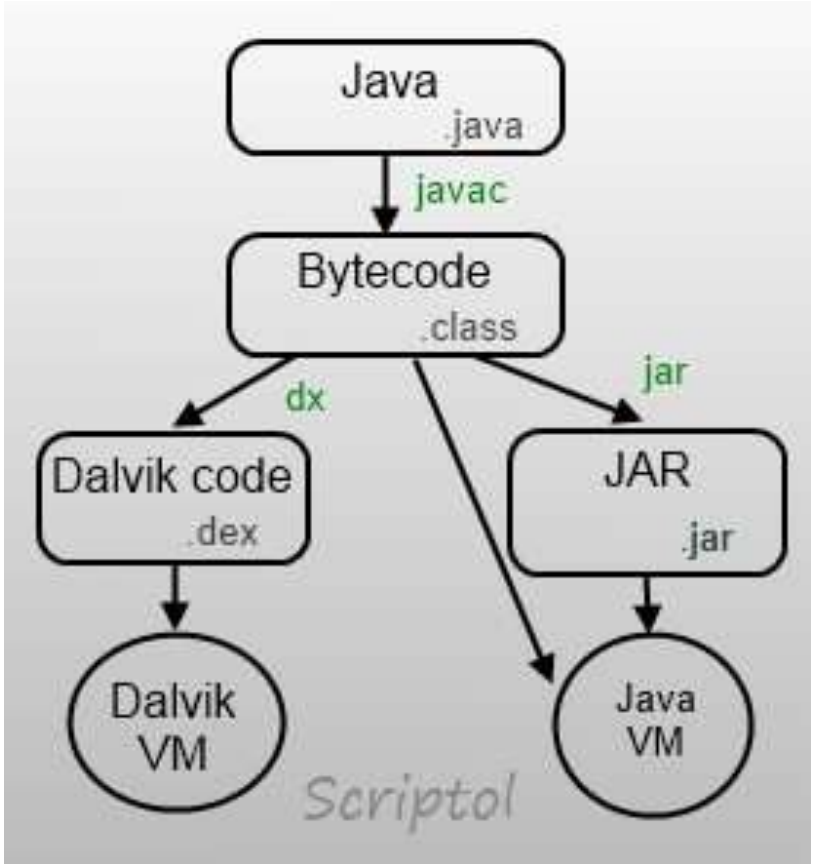
```
D:\simple_jvm>simple_jvm
method attributes_count = 1
method tmp->attribute_name_index = 22
method tmp->attribute_length = 29
method attributes_count = 1
method tmp->attribute_name_index = 22
method tmp->attribute_length = 441
-------------------------------------
Execute Simple JVM
-------------------------------------
-------------------------------------
find and execute <init> method
-------------------------------------
HelloWorld
5 + 10 = 15
5 * 10 = 50
5 - 10 = -5
5 / 10 = 0
d = 15 + 50 + -5 + 0 = 62
Foo Test By WJY
62 % 5 = 2
-------------------------------------
Terminate Simple JVM
-------------------------------------

D:\simple_jvm>
```

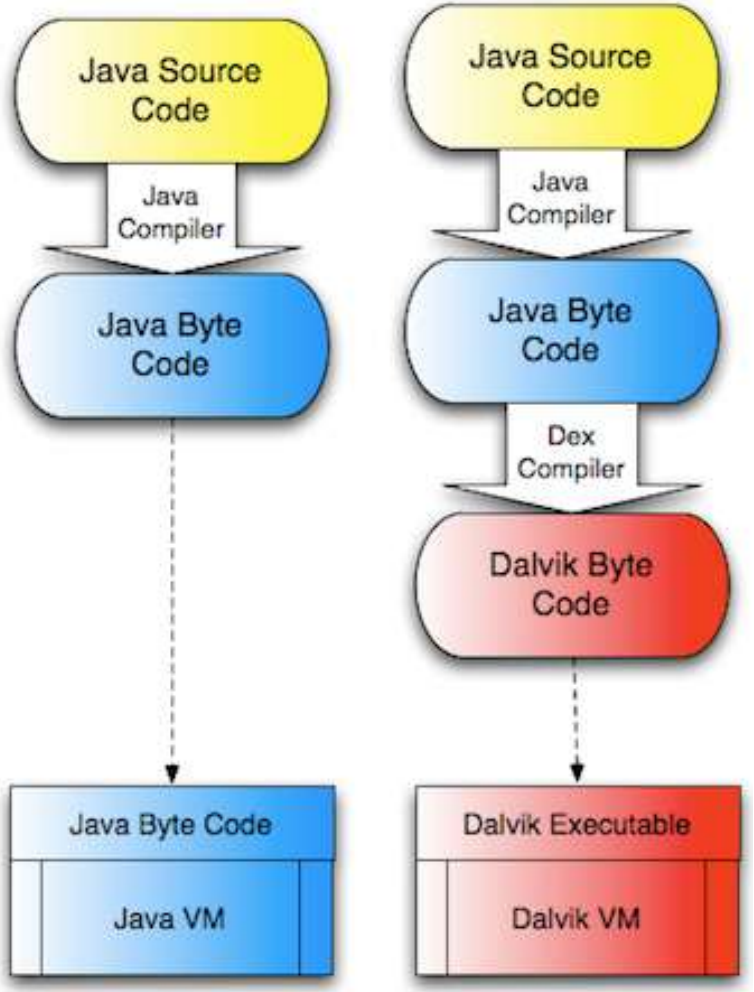
Dalvik Virtual Machine

Dalvik Virtual Machine Overview

- Java Translation for JVM and DVM
- Hello World on Dalvik VM
- DVM ByteCode
- DVM ByteCode Interpreter Generation on Android Open Source
- Dex File Header
- An Simple Dalvik Virtual Machine

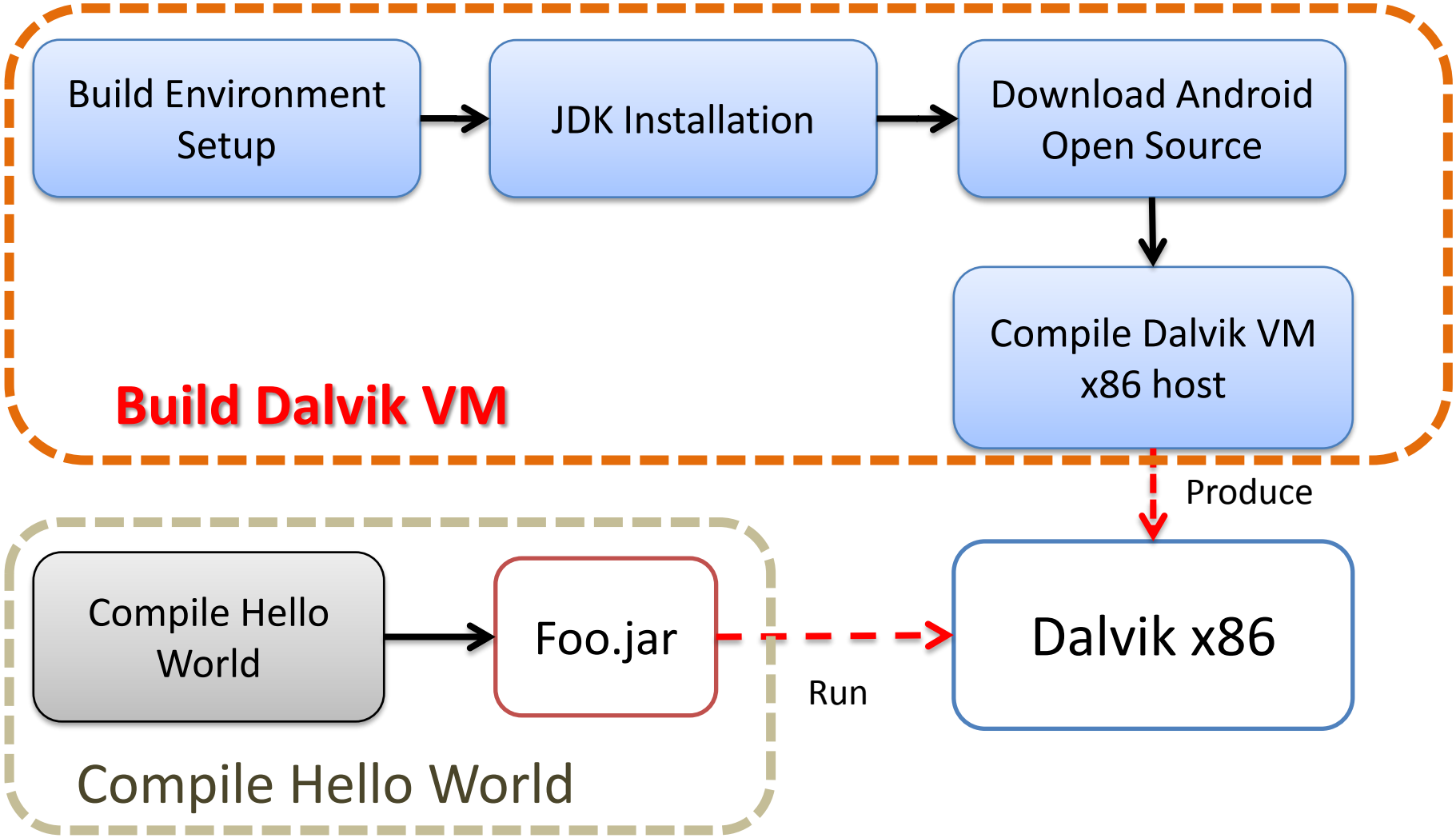
Java Translation for JVM and DVM





http://www.codeproject.com/Articles/461052/Stack-based-vs-Register-based-Virtual-Machine-Arch

Hello World on Dalvik VM Roadmap



Android Open Source Build Setup

- Ubuntu 12.04
  - Virtual Box
- sudo **apt-get install** git gnupg flex bison gperf build-essential zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev libx11-dev:i386 libreadline6-dev:i386 **libgl1-mesa-dri:i386** libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc zlib1g-dev:i386
- 如果發生衝突使用 **libgl1-mesa-glx:i386**

Installing required packages (Ubuntu 12.04)

You will need a 64-bit version of Ubuntu. Ubuntu 12.04 is recommended. Building using an older version of Ubuntu is not supported on master or recent releases.

```
$ sudo apt-get install git gnupg flex bison gperf build-essential \
  zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev \
  libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \
  libgl1-mesa-dev g++-multilib mingw32 tofrodos \
  python-markdown libxml2-utils xsltproc zlib1g-dev:i386
$ sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1 /usr/lib/i386-linux-gnu/libGL.so
```

Android Open Source Initializing a Build Environment

http://source.android.com/source/initializing.html

Build Setup Result

```
libc6-dev 被設定為手動安裝。
有些套件無法安裝。這可能意謂著您的要求難以解決，或是若您使用的是
unstable 發行版，可能有些必要的套件尚未建立，或是被移出 Incoming 了。
以下的資訊或許有助於解決當前的情況：

下列的套件有未滿足的相依關係：
 libgl1-mesa-glx:i386 : 相依關係: libglapi-mesa:i386 (= 8.0.4-0ubuntu0.6)
                        推薦: libgl1-mesa-dri:i386 (>= 7.2)
E: 無法修正問題，您保留 (hold) 了損毀的套件。
anr2@anr2:~$ sudo apt-get install git gnupg flex bison gperf build-essential   z
ip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev   libx11-dev:i386 libre
adline6-dev:i386 libgl1-mesa-dri:i386   libgl1-mesa-dev g++-multilib mingw32 tof
rodos   python-markdown libxml2-utils xsltproc zlib1g-dev:i386
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
zip 已經是最新版本了。
zip 被設定為手動安裝。
build-essential 已經是最新版本了。
gnupg 已經是最新版本了。
libc6-dev 已經是最新版本了。
libc6-dev 被設定為手動安裝。
以下套件為自動安裝，並且已經無用：
  x11-apps x11-session-utils x11-xfs-utils xinit libfs6 thunderbird-globalmenu
使用 'apt-get autoremove' 來將其移除。
下列的額外套件將被安裝：
  g++-4.6-multilib gcc-4.6-base:i386 gcc-4.6-multilib gcc-multilib git-man
  lib32gcc1 lib32gomp1 lib32quadmath0 lib32stdc++6 libbison-dev libc6:i386
  libc6-dev:i386 libc6-dev-i386 libc6-i386 libdrm-dev libdrm-intel1:i386
  libdrm-nouveau1a:i386 libdrm-radeon1:i386 libdrm2:i386 liberror-perl
  libexpat1:i386 libffi6:i386 libfl-dev libgcc1:i386 libgpm2:i386 libkms1
  libllvm3.0:i386 libncurses5:i386 libpciaccess0:i386 libpthread-stubs0
  libpthread-stubs0:i386 libpthread-stubs0-dev libpthread-stubs0-dev:i386
  libreadline6:i386 libstdc++6:i386 libtinfo-dev:i386 libtinfo5:i386
  libx11-6:i386 libx11-dev libx11-doc libxau-dev libxau-dev:i386 libxau6:i386
  libxcb1:i386 libxcb1-dev libxcb1-dev:i386 libxdmcp-dev libxdmcp-dev:i386
  libxdmcp6:i386 libxext-dev linux-libc-dev:i386 m4 mesa-common-dev
  mingw32-binutils mingw32-runtime x11proto-input-dev x11proto-kb-dev
  x11proto-xext-dev xorg-sgml-doctools xtrans-dev zlib1g:i386
```

JDK Installation on Ubuntu

- sudo add-apt-repository ppa:webupd8team/java
- sudo apt-get update
- sudo apt-get install oracle-java6-installer

Installing the JDK

The Sun JDK is no longer in Ubuntu's main package repository. In order to download it, you need to add the appropriate repository and indicate to the system which JDK should be used.

Java 6: for Gingerbread and newer

```
$ sudo add-apt-repository "deb http://archive.canonical.com/ lucid partner"
$ sudo apt-get update
$ sudo apt-get install sun-java6-jdk
```

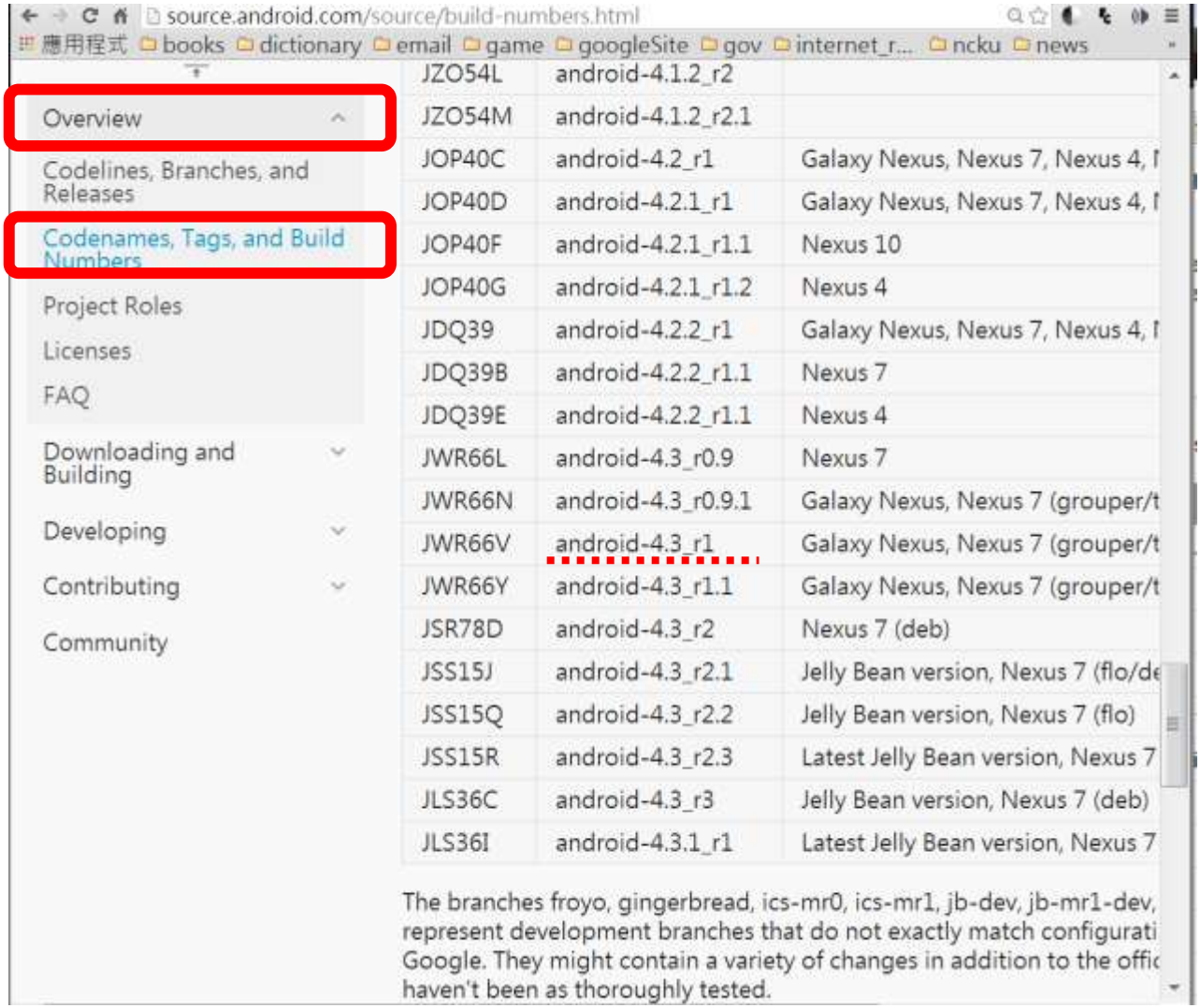
Android Open Source Initializing a Build Environment
http://source.android.com/source/initializing.html

Download Android Open Source(1)

- cd ~
- mkdir android_source
- cd android_source
- mkdir bin
- curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > repo
- chmod a+x repo
- cd ..

Download Android Open Source(2)

- Check android release Tag



Download Android Open Source(3)

- mkdir test & cd test
- mkdir bin & cd bin
- curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > repo
- chmod 777 repo
- cd ..
- mkdir **android-4.3_r1**
- cd **android-4.3_r1**
- ../bin/repo init -u https://android.googlesource.com/platform/manifest -b **android-4.3_r1**
  - Initial android-4.3_r1
- repo sync
  - Download Android Open Source

Download Android Open Source Result

```
 * [new tag]         android-4.3_r2 -> android-4.3_r2
 * [new tag]         android-4.3_r2.1_ -> android-4.3_r2.1_
 * [new tag]         android-4.3_r2.1__ -> android-4.3_r2.1__
 * [new tag]         android-4.3_r2.2 -> android-4.3_r2.2
 * [new tag]         android-4.3_r2.3 -> android-4.3_r2.3
 * [new tag]         android-4.3_r2_ -> android-4.3_r2_
 * [new tag]         android-4.3_r3 -> android-4.3_r3
 * [new tag]         android-4.3_r3.1 -> android-4.3_r3.1
 * [new tag]         android-cts-2.2_r8 -> android-cts-2.2_r8
 * [new tag]         android-cts-2.3_r10 -> android-cts-2.3_r10
 * [new tag]         android-cts-2.3_r11 -> android-cts-2.3_r11
 * [new tag]         android-cts-2.3_r12 -> android-cts-2.3_r12
 * [new tag]         android-cts-4.0.3_r1 -> android-cts-4.0.3_r1
 * [new tag]         android-cts-4.0.3_r2 -> android-cts-4.0.3_r2
 * [new tag]         android-cts-4.0_r1 -> android-cts-4.0_r1
 * [new tag]         android-cts-4.1_r1 -> android-cts-4.1_r1
 * [new tag]         android-cts-4.1_r2 -> android-cts-4.1_r2
 * [new tag]         android-cts-4.2_r2 -> android-cts-4.2_r2
 * [new tag]         android-cts-verifier-4.0.3_r1 -> android-cts-verifier-4.0.3_r1
 * [new tag]         android-cts-verifier-4.0_r1 -> android-cts-verifier-4.0_r1
 * [new tag]         android-sdk-4.0.3-tools_r1 -> android-sdk-4.0.3-tools_r1
 * [new tag]         android-sdk-4.0.3_r1 -> android-sdk-4.0.3_r1
 * [new tag]         android-sdk-adt_r16.0.1 -> android-sdk-adt_r16.0.1
 * [new tag]         android-sdk-adt_r20 -> android-sdk-adt_r20
 * [new tag]         android-sdk-support_r11 -> android-sdk-support_r11

Your Name  [anr2]:
Your Email [anr2@anr2.(none)]:

Your identity is: anr2 <anr2@anr2.(none)>
is this correct [y/N]? y

Testing colorized output (for 'repo diff', 'repo status'):
  black    red      green    yellow   blue     magenta  cyan     white
  bold     dim      ul       reverse
Enable color display in this user account (y/N)? y

repo has been initialized in /home/anr2/android_source/android-4.3_r1
anr2@anr2:~/android_source/android-4.3_r1$
```

Repo Init

```
 * [new tag]         android-2.3.6_r0.9 -> android-2.3.6_r0.9
 * [new tag]         android-2.3.5_r1 -> android-2.3.5_r1
 * [new tag]         android-2.3.4_r1 -> android-2.3.4_r1
 * [new tag]         android-2.3.4_r0.9 -> android-2.3.4_r0.9
 * [new tag]         android-2.3.3_r1.1 -> android-2.3.3_r1.1
 * [new tag]         android-2.3.3_r1 -> android-2.3.3_r1
 * [new tag]         android-2.3.2_r1 -> android-2.3.2_r1
 * [new tag]         android-2.3.1_r1 -> android-2.3.1_r1
 * [new tag]         android-2.2_r1.3 -> android-2.2_r1.3
 * [new tag]         android-2.2_r1.2 -> android-2.2_r1.2
 * [new tag]         android-2.2_r1.1 -> android-2.2_r1.1
 * [new tag]         android-2.2_r1 -> android-2.2_r1
 * [new tag]         android-2.2.3_r2.1 -> android-2.2.3_r2.1
 * [new tag]         android-2.2.3_r2 -> android-2.2.3_r2
 * [new tag]         android-2.2.3_r1 -> android-2.2.3_r1
 * [new tag]         android-2.2.2_r1 -> android-2.2.2_r1
 * [new tag]         android-2.2.1_r2 -> android-2.2.1_r2
 * [new tag]         android-2.2.1_r1 -> android-2.2.1_r1
 * [new tag]         android-2.1_r2.1s -> android-2.1_r2.1s
 * [new tag]         android-2.1_r2.1p2 -> android-2.1_r2.1p2
 * [new tag]         android-2.1_r2.1p -> android-2.1_r2.1p
 * [new tag]         android-2.1_r2 -> android-2.1_r2
 * [new tag]         android-2.1_r1 -> android-2.1_r1
 * [new tag]         android-2.0_r1 -> android-2.0_r1
 * [new tag]         android-2.0.1_r1 -> android-2.0.1_r1
 * [new tag]         android-1.6_r2 -> android-1.6_r2
 * [new tag]         android-1.6_r1.5 -> android-1.6_r1.5
 * [new tag]         android-1.6_r1.4 -> android-1.6_r1.4
 * [new tag]         android-1.6_r1.3 -> android-1.6_r1.3
 * [new tag]         android-1.6_r1.2 -> android-1.6_r1.2
 * [new tag]         android-1.6_r1.1 -> android-1.6_r1.1
 * [new tag]         android-1.6_r1 -> android-1.6_r1
remote: Counting objects: 12613, done
remote: Finding sources: 100% (4079/4079)
remote: Getting sizes: 100% (696/696)
remote: Compressing objects: 100% (4236885/4236885)
  6  154M    6  9.8M    0     0   834k      0  0:03:10  0:00:12  0:02:5
  6  154M    6 10.5M    0     0   828k      0  0:03:11  0:00:13  0:02:5
Receiving objects:  99% (4039/4079), 899.72 KiB | 832 KiB/s
```

Repo Sync

Compile Dalvik VM x86

- source build/envsetup.sh
- lunch 2
- make dalvikvm dalvik-host core ext dexopt framework android.policy services

```
cd android-4.3_r1
source build/envsetup.sh
lunch 2
make dalvikvm core ext dexopt framework android.policy services
cd ..
```

make_dvm.sh

Compile Dalvik VM x86 Result

```
target  C++: libdvm <= dalvik/vm/Exception.cpp
target  C++: libdvm <= dalvik/vm/Hash.cpp
target  C++: libdvm <= dalvik/vm/Init.cpp
dalvik/vm/Init.cpp: In function 'void blockSignals()':
dalvik/vm/Init.cpp:1331:9
-set-variable]
dalvik/vm/Init.cpp: In fu
dalvik/vm/Init.cpp:1740:8
sed-but-set-variable]
dalvik/vm/Init.cpp:1740:1
used-but-set-variable]
dalvik/vm/Init.cpp:1740:3
used-but-set-variable]
dalvik/vm/Init.cpp:1741:8
d-but-set-variable]
dalvik/vm/Init.cpp:1741:1
ed-but-set-variable]
dalvik/vm/Init.cpp:1741:2
ed-but-set-variable]
target  C++: libdvm <= da
target  C++: libdvm <= da
target  C++: libdvm <= da
target  C++: libdvm <= da
```

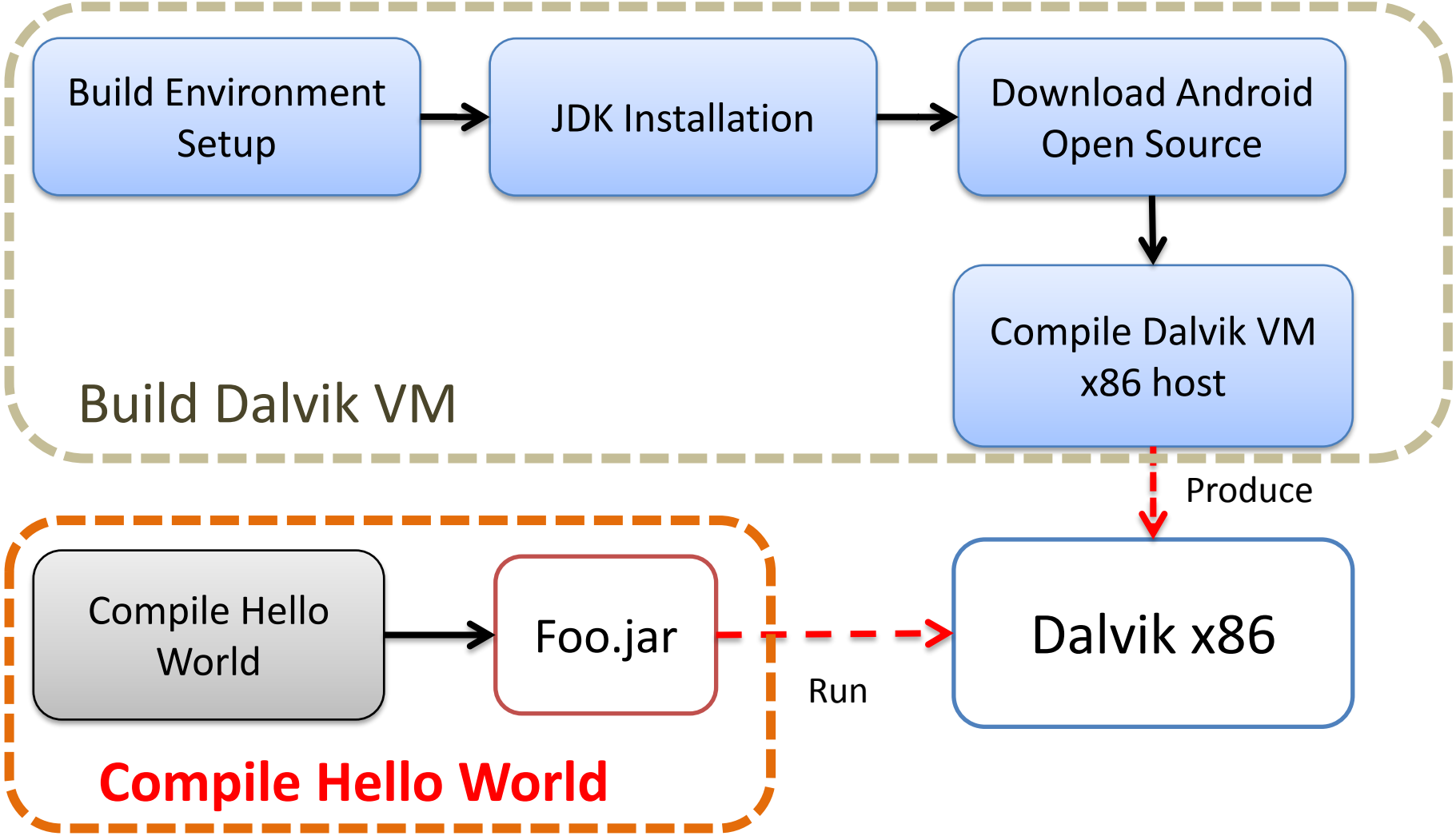
```
target  C++: libdvm <= dalvik/vm/native/java_lang_System.cpp
target  C++: libdvm <= dalvik/vm/native/java_lang_Throwable.cpp
target  C++: libdvm <= dalvik/vm/native/java_lang_VMClassLoader.cpp
target  C++: libdvm <= dalvik/vm/native/java_lang_VMThread.cpp
target  C++: libdvm <= dalvik/vm/native/java_lang_reflect_AccessibleObject.cpp
target  C++: libdvm <= dalvik/vm/native/java_lang_reflect_Array.cpp
target  C++: libdvm <= dalvik/vm/native/java_lang_reflect_Constructor.cpp
target  C++: libdvm <= dalvik/vm/native/java_lang_reflect_Field.cpp
target  C++: libdvm <= dalvik/vm/native/java_lang_reflect_Method.cpp
target  C++: libdvm <= dalvik/vm/native/java_lang_reflect_Proxy.cpp
target  C++
pp
target  C++
et.cpp
target  C++
.cpp
target  C++
rnal.cpp
target  C++
target  C++
target  C++
target  C++
target  C++
```

```
Install: out/target/product/generic_x86/system/framework/android.policy.jar
logtags: out/target/common/obj/JAVA_LIBRARIES/services_intermediates/src/com/androi
LogTags.logtags
logtags: out/target/common/obj/JAVA_LIBRARIES/services_intermediates/src/com/androi
/EventLogTags.logtags
target Java: services (out/target/common/obj/JAVA_LIBRARIES/services_intermediates/
Note: Some input files use or override a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Copying: out/target/common/obj/JAVA_LIBRARIES/services_intermediates/classes-jarjar
Copying: out/target/common/obj/JAVA_LIBRARIES/services_intermediates/emma_out/lib/c
Copying: out/target/common/obj/JAVA_LIBRARIES/services_intermediates/classes.jar
Copying: out/target/common/obj/JAVA_LIBRARIES/services_intermediates/noproguard.cla
target Dex: services
Copying: out/target/common/obj/JAVA_LIBRARIES/services_intermediates/noproguard.cla
target Jar: services (out/target/common/obj/JAVA_LIBRARIES/services_intermediates/j
Dexpreopt Boot Jar: out/target/product/generic_x86/dex_bootjars/system/framework/se
Processing target/product/generic_x86/dex_bootjars/system/framework/services.jar
Done!
Install: out/target/product/generic_x86/system/framework/services.odex
Install: out/target/product/generic_x86/system/framework/services.jar
anr2@anr2:~/android_source$ ls
```

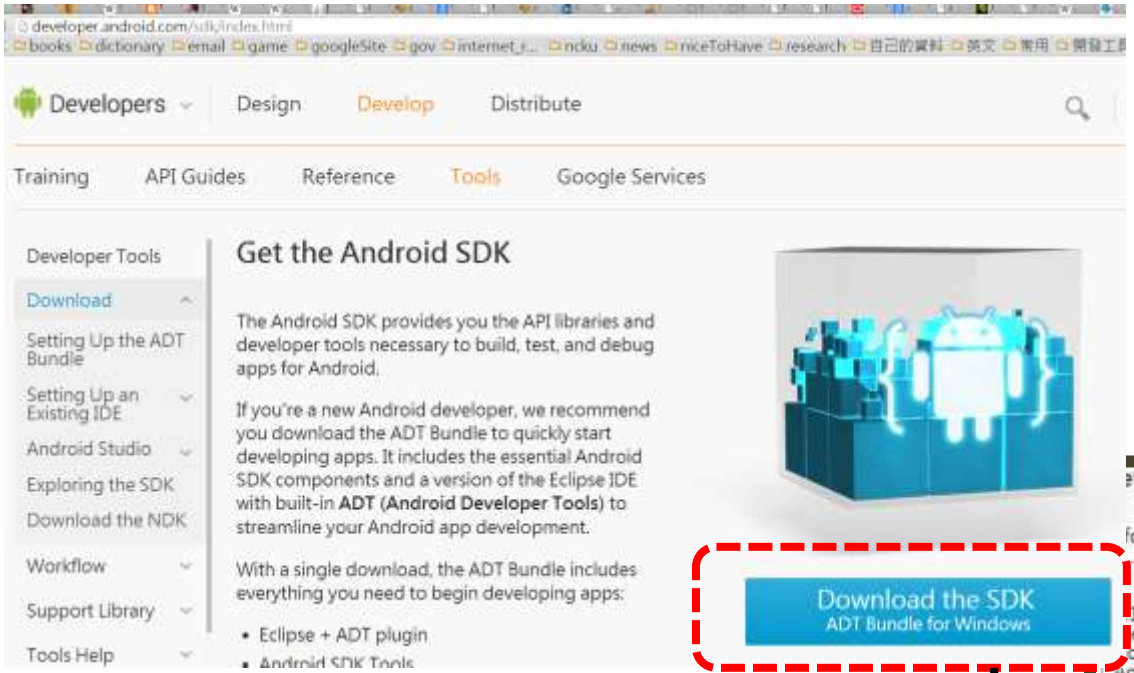
Setup DalvikVM x86

- mkdir -p dalvik-x86-android-4.3
- mkdir -p dalvik-x86-android-4.3/tmp/dalvik-cache
- cp -r **android-4.3_r1**/out/target/product/generic_x86/system/ dalvik-x86-android-4.3/system/
- cp -r **android-4.3_r1**/out/host/linux-x86/bin dalvik-x86-android-4.3/
- cp -r **android-4.3_r1**/out/host/linux-x86/lib dalvik-x86-android-4.3/
- cp -r **android-4.3_r1**/out/host/linux-x86/usr dalvik-x86-android-4.3/system/

Hello World on Dalvik VM Roadmap

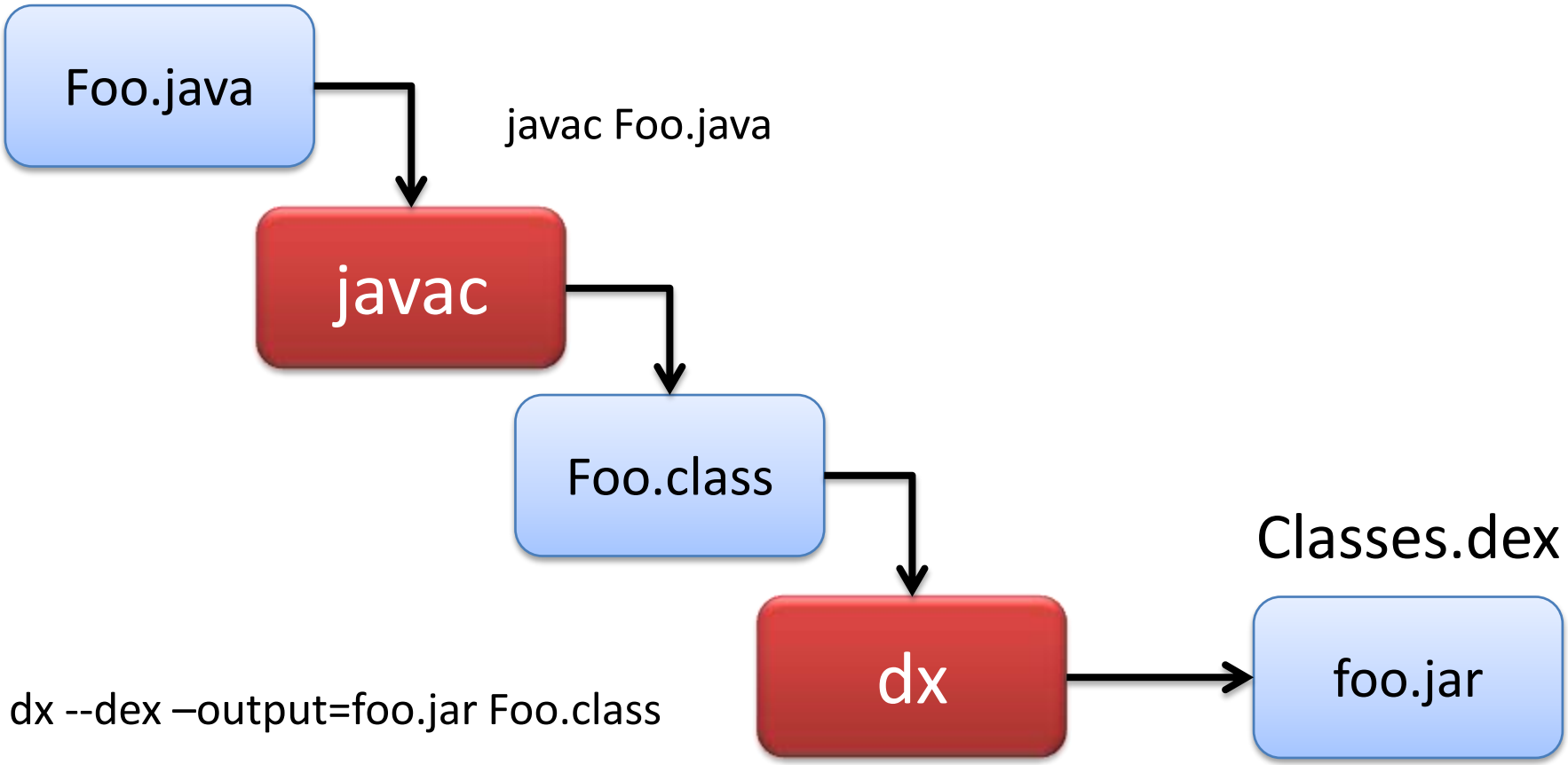


Download ADT (Android Development Tools) for Compile Hello World



http://developer.android.com/sdk/index.html#download

Compile Hello World to DEX



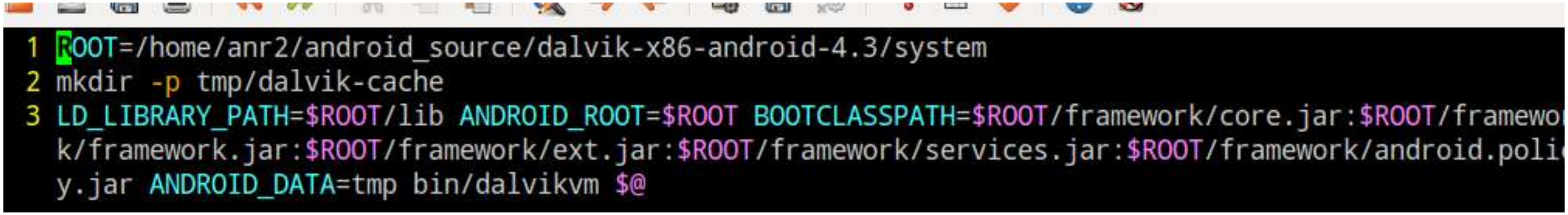
Hello World

- Foo1.java

```
Foo1 {
    public static void main ( String args[] ) {
     System.out.println("Hello World");
    }
}
```

- javac Foo1.java
- dx --dex --output=foo1.jar Foo1.class

Run Hello World on DalvikVM x86



```
ROOT=/home/anr2/android_source/dalvik-x86-android-4.3/system
mkdir -p tmp/dalvik-cache
LD_LIBRARY_PATH=$ROOT/lib ANDROID_ROOT=$ROOT BOOTCLASSPATH=$ROOT/framework/core.jar:$ROOT/framewo
k/framework.jar:$ROOT/framework/ext.jar:$ROOT/framework/services.jar:$ROOT/framework/android.poli
y.jar ANDROID_DATA=tmp bin/dalvikvm $@
```

run_dvm2.sh

$@ 是 bash script 的 parameters

./run_dvm2.sh –cp **foo1.jar** Foo

```
anr2@anr2:~/android_source/dalvik-x86-android-4.3$ ./run_dvm_hello_world.sh
I/dalvikvm( 5203): DexOpt: mismatch dep name: '/home/anr2/android_source/dal
E/dalvikvm( 5203): /home/anr2/android_source/dalvik-x86-android-4.3/system/f
I/dalvikvm( 5203): Zip is good, but no classes.dex inside, and no valid .ode
I/dalvikvm( 5203): DexOpt: mismatch dep name: '/home/anr2/android_source/dal
E/dalvikvm( 5203): /home/anr2/android_source/dalvik-x86-android-4.3/system/f
I/dalvikvm( 5203): Zip is good, but no classes.dex inside, and no valid .ode
I/dalvikvm( 5203): DexOpt: mismatch dep name: '/home/anr2/android_source/dal
E/dalvikvm( 5203): /home/anr2/android_source/dalvik-x86-android-4.3/system/f
I/dalvikvm( 5203): Zip is good, but no classes.dex inside, and no valid .ode
I/dalvikvm( 5203): DexOpt: mismatch dep name: '/home/anr2/android_source/dal
E/dalvikvm( 5203): /home/anr2/android_source/dalvik-x86-android-4.3/system/f
I/dalvikvm( 5203): Zip is good, but no classes.dex inside, and no valid .ode
Hello World
anr2@anr2:~/android_source/dalvik-x86-android-4.3$
```

Dalvik VM and ByteCode

- Register-based, 32bits
- Instructions Fetch Unit : 16 bits
  - Byte code store as binary
- Constant pools
  - String, Type, Field, Method, Class
- Human-syntax and mnemonics

Insturction Suffix	
-wide(64bits OpCodes)	-char
-boolean	-short
-byte	-int
-long	-float
-object	-string
-class	-void

Dalvik ByteCode Human-syntax

- Example "move-wide/from16 vAA, vBBBB":
  - Opcode : “move" move a register's value).
  - "wide" is the name suffix
    - it operates on wide (64 bit) data.
  - "from16" is the opcode suffix
    - 16-bit register reference as a source.
  - "vAA" is the **destination register**
    - v0 – v255.
  - "vBBBB" is the **source register**
    - v0 – v65535.

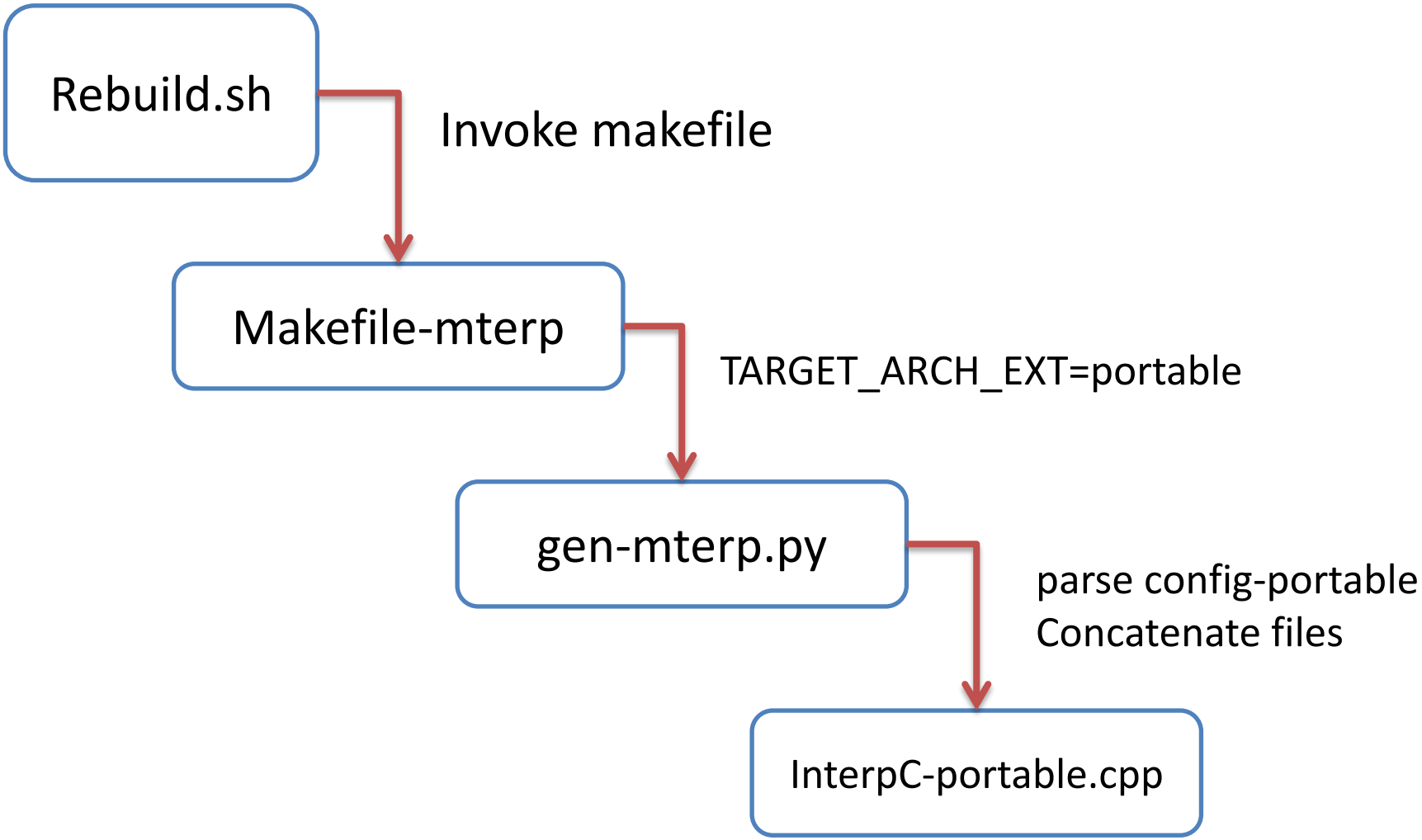
Dalvik ByteCode Example

OpCode	suffix1	Suffix2	destination	source
move	wide	from16	vAA	vBBBB
const		4	v6	int #0
double-to-int			v0	v0
invoke-virtual			method@0002	{v3,v4}
const-string			string@0005	v4
mul-int			v3	v0,v1
add-int		2addr	v2	v2,v3

DVM ByteCode Interpreter Generation on AOSP

- How to generate the **InterpC-portable.cpp**
  - rebuild.sh TARGET_ARCH=portable
  - parse Makefile-mterp
  - gen-mterp.py TARGET_ARCH=portable
  - parse config-portable
  - concatenate cpp files to one files
    - InterpC-portable.cpp

Dalvik Mterp Generation flow



Dex Header

Magic – 8 bytes – "dex\n035\0"
Checksum – 4 bytes – Adler32 checksum from bytes offset 12 and on
Signature – 20 bytes – SHA-1 of bytes from 32 on
File Size – 4 bytes – Exactly what it sounds like, the file size
Header Size – 4 bytes – Will always be "70"
Endian Tag – 8 bytes – Will always be "78563412"
Zeros – 8 bytes – Exactly that, eight bytes of zeros
Map Offset – 4 bytes – Leads to below, need more research on this though
String Table Size – 4 bytes – Size of the string's table
String Table Offset – 4 bytes – Offset to the string table
TypeTable Size – 4 bytes – Size of the type's table
Type Table Offset – 4 bytes – Offset to the type table
Prototype Table Size – 4 bytes – Size of the prototype's table
Prototype Table Offset – 4 bytes – Offset to the prototype table
Field Table Size – 4 bytes – Size of the field's table
Field Table Offset – 4 bytes – Offset to the field table
Method Table Size – 4 bytes – Size of the method's table
Method Table Offset – 4 bytes – Offset to the method table
Class Table Size – 4 bytes – Size of the class's table
Class Table Offset – 4 bytes – Offset to the class table

http://www.strazzere.com/blog/2008/11/updated-dalvik-vm-dex-file-format/

Dex Translation Example

Java source code	Dalvik instructions
`static byte foo(int x) {`	*parameter* `x = v2`
	`const/16 v0 1000`
`  if(x > 1000) {`	`if-le v2 v0 +9`
	`rem-int/lit16 v0 v2 1000`
`  byte y = foo(x % 1000);`	`invoke-static v0` $@m_0$
	`move-result v0`
`  return y;`	`return v0`
`  }`	`const/4 v0 2`
`  byte [] data = {7, 9};`	`new-array v0 v0` $@c_0$
	`fill-array-data v0 +8`
`  byte z = data[x % 2];`	`rem-int/lit8 v1 v2 2`
	`aget-byte v0 v0 v1`
`  return z;`	`goto -11`
`}`	`[0:  7]`
	`[1:  9]`

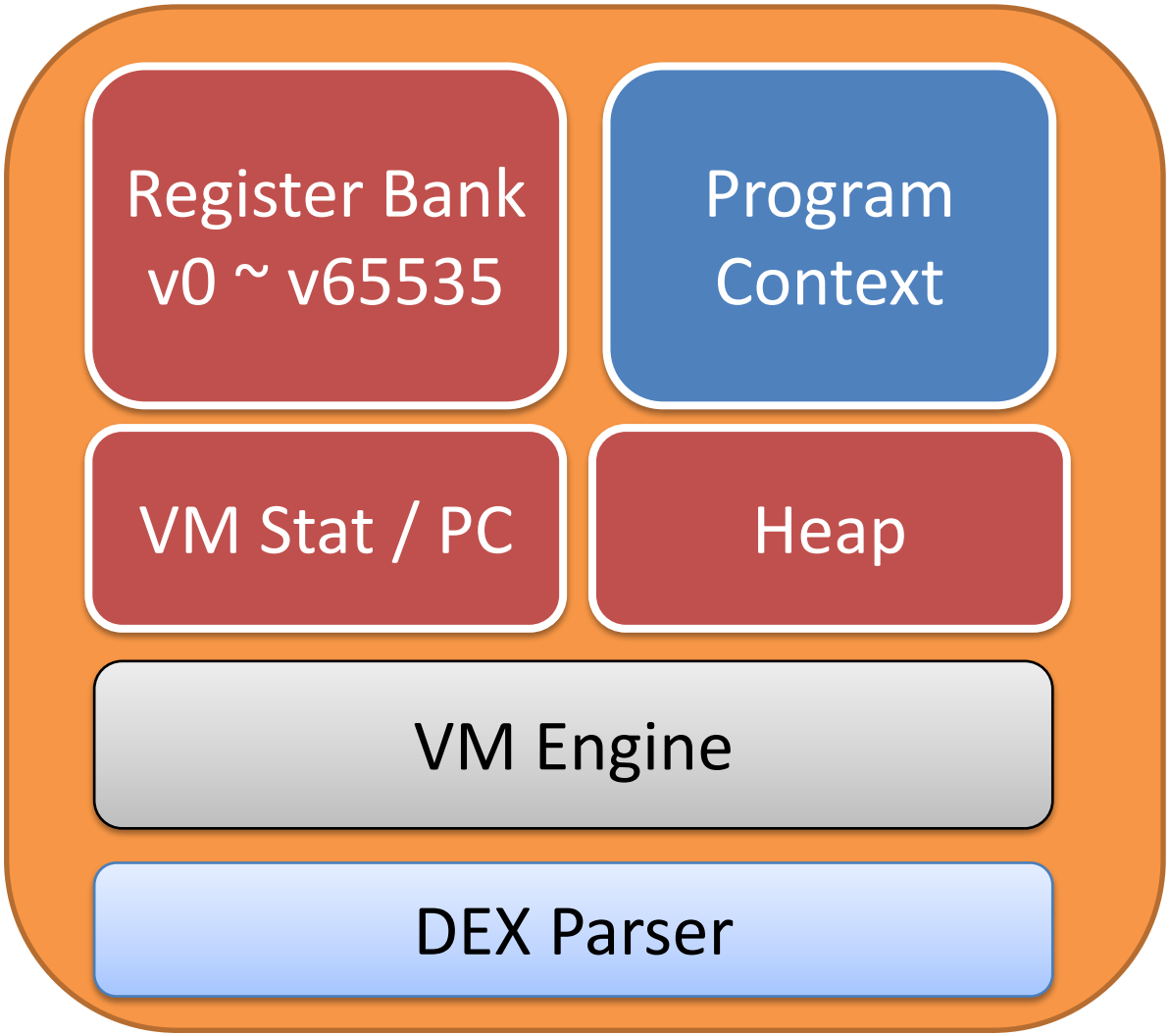
$@c_0$ = byte array $@m_0$ = foo()

SymDroid: Symbolic Execution for Dalvik Bytecode- Technical Report CS-TR-5022, July 2012
Jinseong Jeon, Kristopher K. Micinski, Je rey S. Foster
Department of Computer Science, University of Maryland, College Park

Dalvik ByteCode Example 2

Dalvik instructions
parameter x = v2
const/16 v0 1000
if-le v2 v0 +9
rem-int/lit16 v0 v2 1000
invoke-static v0 @m_0
move-result v0
return v0
const/4 v0 2
new-array v0 v0 @c_0
fill-array-data v0 +8
rem-int/lit8 v1 v2 2
aget-byte v0 v0 v1
goto -11
[0: 7]
[1: 9]

A Simple Dalvik Virtual Machine



```
yteCode byteCodes[] = {
    { "move-result-wide"   , 0x0B, 2,  op_move_result_wide },
    { "move-result-object", 0x0C, 2,  op_move_result_object }
    { "return-void"        , 0x0e, 2,  op_return_void },
    { "const/4"            , 0x12, 2,  op_const_4 },
    { "const/16"           , 0x13, 4,  op_const_16 },
    { "const-wide/high16"  , 0x19, 4,  op_const_wide_high16 },
    { "const-string"       , 0x1a, 4,  op_const_string },
    { "new-instance"       , 0x22, 4,  op_new_instance },
    { "sget-object"        , 0x62, 4,  op_sget_object },
    { "invoke-virtual"     , 0x6e, 6,  op_invoke_virtual },
    { "invoke-direct"      , 0x70, 6,  op_invoke_direct },
    { "invoke-static"      , 0x71, 6,  op_invoke_static },
    { "double-to-int"      , 0x8a, 2,  op_double_to_int},
    { "add-int"            , 0x90, 4,  op_add_int },
    { "sub-int"            , 0x91, 4,  op_sub_int },
    { "mul-int"            , 0x92, 4,  op_mul_int },
    { "div-int"            , 0x93, 4,  op_div_int },
    { "add-int/2addr"      , 0xb0, 2,  op_add_int_2addr},
    { "add-double/2addr"   , 0xcb, 2,  op_add_double_2addr},
    { "mul-double/2addr"   , 0xcd, 2,  op_mul_double_2addr},
    { "div-int/lit8"       , 0xdb, 4,  op_div_int_lit8 }
};
static byteCode_size = sizeof(byteCodes)/ sizeof(byteCode);
```

Simple DVM Instruction Table :

simple_dvm_bytecodes.c

add-int implementation

```
// 0x90 add-int vx,vy vz
// Calculates vy+vz and puts the result into vx.
// 9000 0203 - add-int v0, v2, v3
// Adds v3 to v2 and puts the result into v0.
int op_add_int( DexFileFormat *dex, simple_dalvik_vm *vm, u1 *ptr, int *pc )
{
    int reg_idx_vx = 0;
    int reg_idx_vy = 0;
    int reg_idx_vz = 0;
    int x = 0, y =0 , z =0;
    reg_idx_vx = ptr[*pc+1];
    reg_idx_vy = ptr[*pc+2];
    reg_idx_vz = ptr[*pc+3];

    if ( is_verbose() ) {
        printf("add-int v%d, v%d, v%d\n", reg_idx_vx, reg_idx_vy,
                reg_idx_vz);
    }
    // x = y + z
    load_reg_to( vm,reg_idx_vy, (unsigned char*)&y);
    load_reg_to( vm,reg_idx_vz, (unsigned char*)&z);
    x = y + z;
    store_to_reg(vm,reg_idx_vx, (unsigned char*)&x);
    *pc = *pc + 4;
    return 0;

}
```

An Simple Dalvik VM Experiment

```
Execute Simple Dalvik Virtual Machine
-------------------------------------
get random number = 0.349712
HelloWorld
--------------------
initial value
random number x : 15
x = 15
y = 6345
c = 0
d = 23456
f = 0
--------------------
c = x + y = 15 + 6345 = 6360
d = d + c = 29816
x = c/2 = 3180
c = x * y = 3180 * 6345 = 20177100
d = d + c = 20206916
x = c/2 = 10088550
c = x - y = 10088550 - 6345 = 10082205
d = d + c = 30289121
x = c/2 = 5041102
c = x / y = 5041102 / 6345 = 794
d = d + c = 30289915
f = 30289915 + 5041102 + 6345 + 794 = 35338156
Foo Test By WJY
-------------------------------------
Stop    Simple Dalvik Virtual Machine
-------------------------------------
```

goo.gl/J5VFQV

1. **make_simple_dvm**
2. **simple_dvm Foo1.dex**

References

- Android Open Source
  - http://source.android.com/index.html
- Android XRef
  - http://androidxref.com/
- Java ByteCodes Fundamentals
  - http://arhipov.blogspot.tw/2011/01/java-bytecode-fundamentals.html
- Java ByteCode Instruction listings
  - http://en.wikipedia.org/wiki/Java_bytecode_instruction_listings
- Dalvik Wiki
  - http://en.wikipedia.org/wiki/Dalvik_(software)