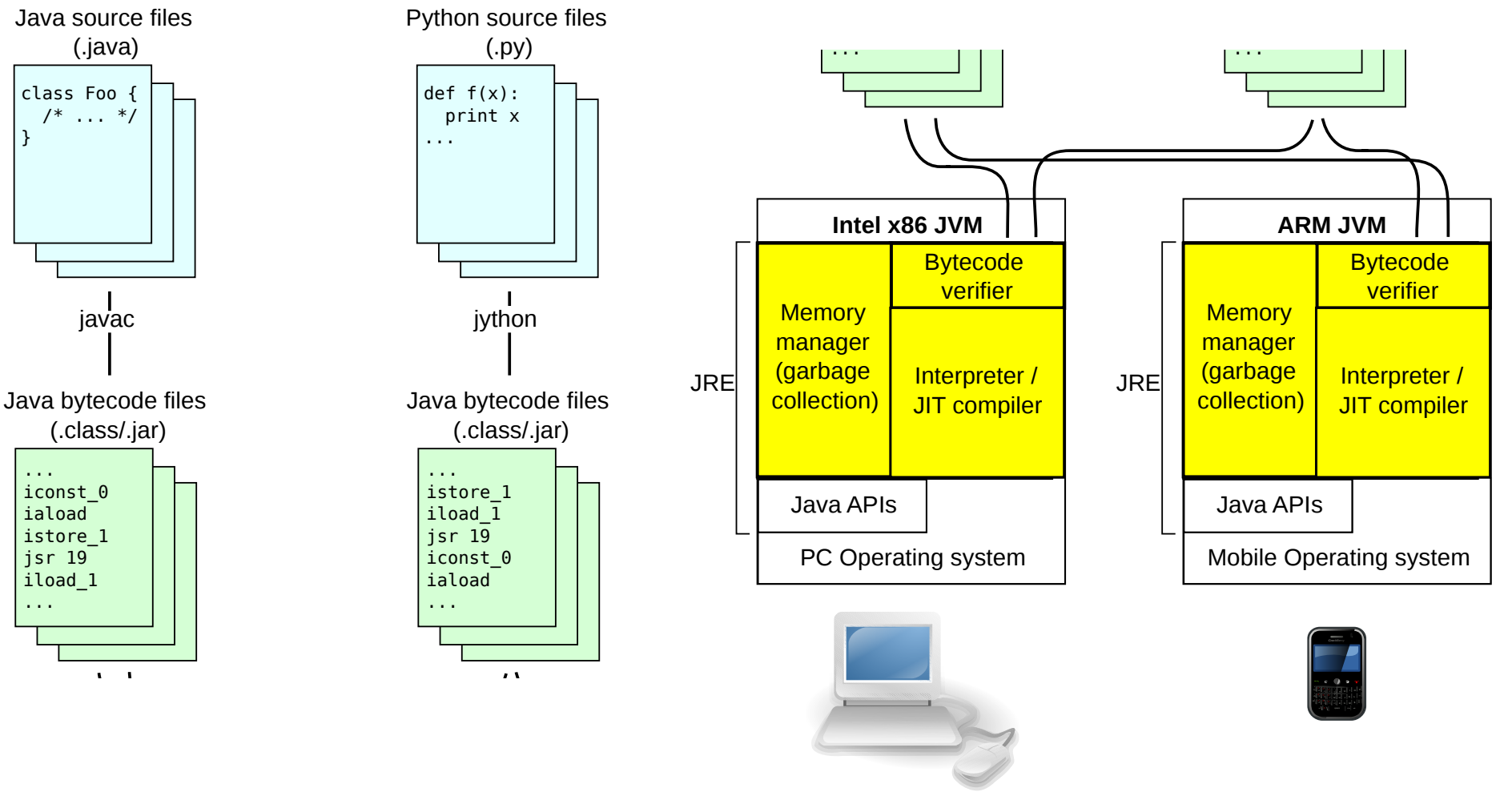
Introduction (0A)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

JVM (Java Virtual Machine)



http://en.wikipedia.org/

Java Bytecode

A **Java virtual machine (JVM)**
a process virtual machine that executes Java bytecode.

The code execution component of the Java platform.

Java bytecode
the instruction set of the Java virtual machine.

for a instruction (**opcode**), 1 ~ 2 bytes
for passing parameters, 0+ bytes

The 256 possible byte-long opcodes
- 198 are currently in use
- 51 are reserved for future use
- 3 are set aside as permanently unimplemented

an **opcode (operation code)**
the portion of a machine language instruction
that specifies the operation to be performed.

http://en.wikipedia.org/

Java Bytecode Instructions

Instructions fall into a number of broad groups:

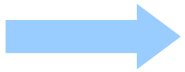
Load and store	(e.g. aload_0, istore)
Arithmetic and logic	(e.g. ladd, fcmpl)
Type conversion	(e.g. i2b, d2i)
Object creation and manipulation	(new, putfield)
Operand stack management	(e.g. swap, dup2)
Control transfer	(e.g. ifeq, goto)
Method invocation and return	(e.g. invokespecial, areturn)

Prefix/Suffix	Operand Type
i	integer
l	long
s	short
b	byte
c	character
f	float
d	double
z	boolean
a	reference

http://en.wikipedia.org/

javac

```
outer:
for (int i = 2; i < 1000; i++) {
    for (int j = 2; j < i; j++) {
        if (i % j == 0)
            continue outer;
    }
    System.out.println (i);
}
```



javac

```
0:   iconst_2
1:   istore_1
2:   iload_1
3:   sipush  1000
6:   if_icmpge       44
9:   iconst_2
10:  istore_2
11:  iload_2
12:  iload_1
13:  if_icmpge       31
16:  iload_1
17:  iload_2
18:  irem
19:  ifne    25
22:  goto    38
25:  iinc    2, 1
28:  goto    11
31:  getstatic       #84; /
34:  iload_1
35:  invokevirtual   #85; /
38:  iinc    1, 1
41:  goto    2
44:  return
```

http://en.wikipedia.org/

Machine Code

Machine code or **machine language**
a set of instructions executed directly by a computer's central processing unit (CPU).

a load, a jump, or an ALU operation on a unit of data in a CPU register or memory. Every program directly executed by a CPU is made up of a series of such instructions.

http://en.wikipedia.org/

MIPS Machine Code Examples

```
   6      5     5     5     5      6 bits
[  op  |  rs |  rt |  rd |shamt| funct]  R-type
[  op  |  rs |  rt | address/immediate]  I-type
[  op  |        target address        ]  J-type
```

rs, *rt*, and *rd* indicate register operands; *shamt* gives a shift amount; and the *address* or *immediate* fields contain an operand directly.

For example adding the registers 1 and 2 and placing the result in register 6 is encoded:

```
[  op  |  rs |  rt |  rd |shamt| funct]
    0     1     2     6     0     32     decimal
 000000 00001 00010 00110 00000 100000   binary
```

Load a value into register 8, taken from the memory cell 68 cells after the location listed in register 3:

```
[  op  |  rs |  rt | address/immediate]
   35     3     8           68          decimal
 100011 00011 01000 00000 00001 000100   binary
```

Jumping to the address 1024:

```
[  op  |        target address        ]
    2                1024              decimal
 000010 00000 00000 00000 10000 000000   binary
```

http://en.wikipedia.org/

Intel Assembly Language Examples

The Intel opcode 10110000 (`B0`) copies an 8-bit value into the *AL* register, while 10110001 (`B1`) moves it into *CL* and 10110010 (`B2`) does so into *DL*. Assembly language examples for these follow.[6]

```
MOV AL, 1h        ; Load AL with immediate value 1
MOV CL, 2h        ; Load CL with immediate value 2
MOV DL, 3h        ; Load DL with immediate value 3
```

The syntax of MOV can also be more complex as the following examples show.[7]

```
MOV EAX, [EBX]    ; Move the 4 bytes in memory at the address contained in EBX into EAX
MOV [ESI+EAX], CL ; Move the contents of CL into the byte at address ESI+EAX
```

http://en.wikipedia.org/

Java Class File

Java class file
a file with the .class filename extension
containing a Java bytecode
produced by Java compiler

from Java programming language source files (.java files) containing Java classes. If a source file has more than one class, each class is compiled into a separate class file.

http://en.wikipedia.org/

Java Platform

API

JVM

Java Edition

Java EE

Java SE

Java ME

Java Card

Java FX

JVM
Java ME VM
Java Card VM

Java Programs

Java Applications

Java Applet

Java Sublet

Java Server Page

Java Beans

Android Application

JDK & JRE

Java Development Kit

Java Runtime Environment

References

[1] Java in a nutshell, 4th ed, David Flanagan
[2] An Introduction to Object-Oriented Programming with Java, C. Thomas, Wu
[3] Power Java, I. K. Chun (in Korean)