**Java Virtual Machine**

Java consists of:

* + Programming language specification.
  + Application Program Interface.
  + Virtual Machine Specification.

**Figure 1: Java virtual machine**

**Figure 2: Host system**

**The Java Development Environment:**

**Figure 3: The Java Development Environment**

**Operating System operations**

* Interrupt Driven by hardware.
* Software error or request, creates exception or trap
  + Division by zero, request for operating System service.
  + Other process problems include infinite loop, processes modifying each other or the operating system.
* Dual-mode operation allows OS to protect itself and other system components.
  + User mode and kernel mode
* Mode bit :

1 -> User mode

0 -> kernel mode

* + provided by hardware
  + Provides ability to distinguish when system is running user code or kernel code.
* Some instructions designated as privileged, only executable in kernel mode.
* System call changes mode to kernel, return from call resets it to user

**Transition from User to Kernel Mode**

* Timer to prevent infinite loop/process, hopping resources.
  + Set interrupt after specific period
    - Operating system decrements counter
    - When counter zero generate an interrupt
  + Set up before scheduling process to regain control or terminate program that exceeds allocate time

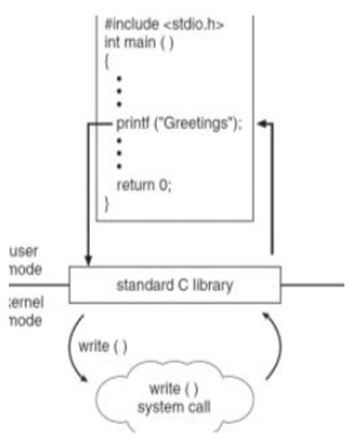
**Figure 4: The Transition from User to Kernel Mode**

**System Calls**

* Provide the interface between a process and the operating system.
* These calls are generally available as a routine written in C and C++, although low-level tasks, may need to be written using assembly-language instructions.
* Mostly accessed by programs via a high-level Application Program Interface (API) rather than direct system call use.
* Three common APIs are Win32 API for windows, POSIX API for UNIX, Linux, and JAVA API for the Java Virtual Machine (JVM)

**Standard C library Example:**

* C program invoking printf() library call, which calls write() system call



**System Call Parameter Passing**

* Three general methods used to pass parameters to the Operating System OS
  + **Simplest**: pass the parameters in registers.
  + **Block**: parameters stored in a block or table, in memory, and address of block passed as a parameter in a register
  + **Stack**: parameters placed, or pushed, onto stack by the program and popped off the stack by the Operating System
* Block and Stack methods do not limit the number or length of parameters being passed.

**Types of System Calls**

1. **Process Control:** Load, execute, create, terminate, get/set process attributes, wait for time, allocate, free memory.
2. **File Management:** Create file, delete file, open, close, read, write, reposition, get/set file attributes.
3. **Device Management:** Request device, release device, read, writes, reposition, get/set device attributes.
4. **Information Maintenance:** Get/set time or date, get/set process, file, or device attributes
5. **Communication:** Create, delete connection, send, receive messages, transfer status information, attach remote devices

**System Programs**

* **System Programs**: provide a convenient environment for program development and execution. it can be divided into:

1. **File Management**: create Delete, Copy, Rename, Print and generally manipulated files and directories.
2. **Status Information:** Some ask the system for info-date, time, and amount of available memory, disk space, and number of users. Other provide detailed performance, logging, and debugging information, Typically these programs format and print the output to the terminal or other output devices.
3. **File Modification:** Text editors to create and modify files. Special commands to search contents of files or perform transformations of the text.
4. **Programming-language support:** Compilers, assemblers, debuggers and interpreters.
5. **Program loading and execution:** Absolute loaders, relocatable loaders, linkage editors for higher-level and machine language.
6. **Communications:** provide the mechanism for creating a connection among processes, users, and computer systems. Allow users to send message to one another’s screen, browse web pages, send electronic-mail messages, transfer files from one machine to another.