

Contact Information

Publications@ipcg.com

ipCapital Group, Inc.
400 Cornerstone Drive, Suite 325
Williston, VT 05495
United States of America
(802) 872-3200

TITLE

Virtual Machine Debugging Module

ABSTRACT

A Debugging Module, within the operating system, runs a virtual machine on which the user may run unstable code for testing purposes without risking harm to the system.

1. BACKGROUND

Problem or Opportunity

Testing code is an inevitable part of any software development process. When testing recently developed and previously untested code there is a potential for something to go wrong. If the code being tested has access to important system files, there is a danger that these system files could be irreparably damaged. Even if the code does not access important system files, a crashing code could necessitate a reboot of the machine, wasting time and frustrating the tester. A virtual machine running an instance of an Operating System could be used to provide a safe testing environment.

However, booting a virtual machine solely for this purpose may only load a new instance of the Operating System itself, without any specific user processes. Because of this, the user may be required to restart all user processes, including background applications and development software, to produce an accurate testing environment. A more complete system for testing code under real system conditions is necessary.

Background Publications

Previous publications have attempted to address the opportunity for code testing using virtual machines. However, the existing publications do not describe a system in which a local environment is recreated for code testing in a “real world” scenario.

US Patent Number 7228532 describes a system for verifying code with a platform-independent virtual machine. The platform-independent code is transferred to the virtual

machine for verifications and code cleanup. This system does not test the code within an operating system and is not able to recreate the local environment.

US Patent Number 6839647 describes a system for testing code in a java virtual machine environment. A Java virtual machine acts as a “harness” for a second virtual machine running within the Java virtual machine. The “harness” allows for safer testing of code, however, the system does not allow for the simple recreation of the local environment for testing purposes. In addition, the invention uses two virtual machines which may be processor intensive.

2. SUMMARY OF INVENTION

Invention Summary

In this invention, a Debugging Module, embedded within the operating system, runs a Virtual Machine specifically for testing code. The Virtual Machine allows the software developer to test the code on a virtual version of the Operating System. This way, the code runs as if it were on a real machine without the risk of damaging important system files and without the hassle of potential reboots.

A virtual machine is a software simulation of a computing machine that runs on real computing machine. A virtual machine can run an operating system and applications. However, a virtual machine does not have the ability to access or edit the real machine outside of the virtual machine.

When a user desires to test a piece of code, the user initiates the “testing mode” of the Debugging Module. When in testing mode, the Debugging Module loads a duplicate instance of the currently running OS into the Virtual Machine, and switches context from the main OS to the Virtual Machine. The Debugging module then automatically copies the user session and physical memory contents over to the Virtual Machines Virtual Memory, creating an identical copy of the main OS running on the Virtual Machine, with all user processes preserved. This allows the user to seamlessly transfer all operations to the Virtual Machine without the need to reload development software. Within the Virtual Machine, the code can be executed safely.

If the test code requires access to a file or multiple files stored on the user machine, the Debugging Module can copy file(s) to the Debugging Module’s virtual memory for use by the Virtual Machine, so that the code can be tested without altering the original files.

If at any point while running the test code, the test code crashes, no files on the User Machine will have been altered, nor will the OS on the User Machine crash. If the Virtual OS running on the Virtual Machine crashes, the Virtual Machine can be restarted to continue testing the code without needing to restart the User Machine.

Unique Concepts

The unique concept of the present invention is the ability to quickly test code in an environment similar to the environment encountered during normal operation. The local operating environment is recreated on a virtual machine in order to test the code. This invention can establish this testing environment quickly, without unnecessary hassle to the user.

3. DESCRIPTION OF THE INVENTION

Figure 1 depicts the structure of the Virtual Machine Debugging Module System.

The User Machine is a computing device such as a desktop computer, laptop computer, or mobile device.

The OS is an operating system, such as Linux, that manages hardware, applications, and user interactions.

The Local Storage is the internal data storage on the user machine such as a hard drive.

The OS System Files are all the files belonging to the OS installation.

The Debugging Module is the module running and managing the Virtual Machine used for testing code. The Debugging Module also manages the Virtual Machines requests for access to files in Local Storage.

The Virtual Machine is a virtual machine that simulates the existence of another computer, and is capable of running a second instance of an Operating System. During testing mode, all processes are run on the virtual machine. The Virtual Machine does not have direct access to any files stored in Local Storage.

The Virtual OS is another instance of the OS running as a process on the Virtual Machine inside the Debugging Module. This Virtual OS can be used to run code without risk to the User Machine and OS.

The Virtual Memory is simulated memory for the Virtual Machine.

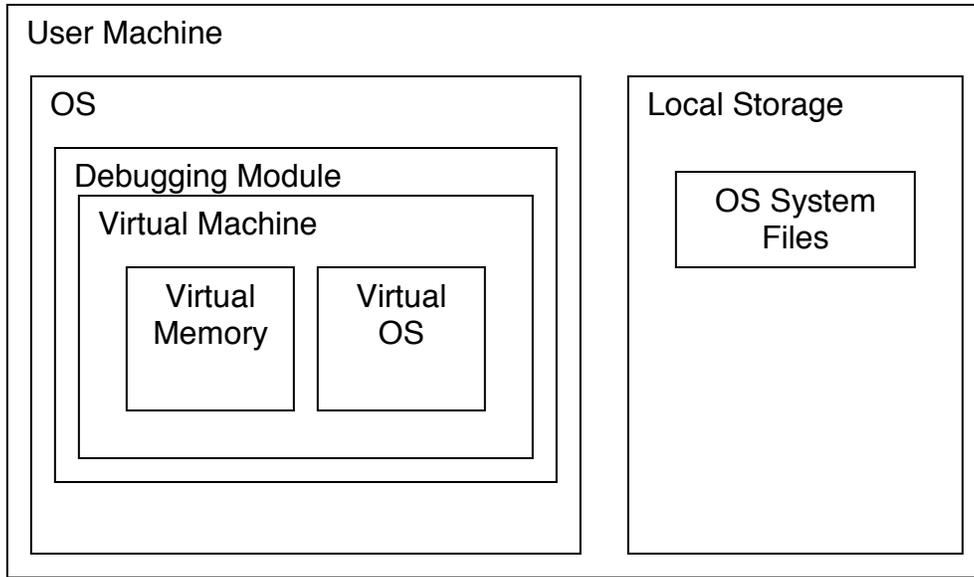


Figure 1. Diagram of the Virtual Machine Code Testing System.

Figure 2 depicts the method for testing code using the Virtual Machine.

In step 1, the user initiates testing mode in the Debugging Module.

In step 2, the Debugging Module automatically transfers all session information to the Virtual Machine, including open applications, and all memory is loaded into the virtual memory.

In step 3, The User executes the test code through the Virtual OS running on the Virtual Machine.

In (optional) step 4, the test code process requests a file from Local Storage.

In (optional) step 5, the Debugging Module makes a copy of the requested file in the Virtual Memory of the Virtual Machine.

In (optional) step 6, the test code accesses any necessary files from the Virtual Memory and continues to run.

In step 7, if the test code crashes, the method proceeds to step 8. Otherwise the method proceeds to step 9.

In step 8, the user closes or restarts the Virtual Machine within the Debugging Module, and continues either code development or code testing without any harm to the User Machine.

In step 9, the code executes successfully without any changes to the User Machine.