Cambridge International AS & A Level Computer Science

9608
For examination from 2017

Topic 3.4.2 Virtual machine

Contents

Introduction

How to use this guide

The aim of this guide is to facilitate your teaching of Cambridge International AS and A Level Computer Science topic 3.4.2 Virtual machine, part of the advanced theory topic 3.4 System software. The guidance and activities in this resource are designed to help teachers devise programmes of study which provide teaching time devoted to theory work as well as activities that consolidate learning.

Section 1 lists some key terms used in this topic and their definitions. Section 2 introduces virtual machines, gives examples of how they are used and lists some benefits and limitations. Section 3 lists some online resources that you or your learners may find useful. Section 4 gives ideas for class and homework activities.

Learning objectives

Using this document should help you guide learners in the following syllabus learning objectives:

- show understanding of the concept of a virtual machine
- give examples of the role of virtual machines
- show understanding of the benefits and limitations of virtual machines.

Prior knowledge

Before you begin teaching this topic you should:

- understand what virtual machines are and what they do, including the difference between system and process virtual machines
- be familiar with different ways in which virtual machines are used
- be familiar with how the Java Virtual Machine works
- understand some benefits and limitations of virtual machines.

1. Key terms

| Word/phrase	Meaning
byte code	intermediate code that a virtual machine converts into machine code
compiler	a program that converts source code into machine code
emulator	software that runs a virtual machine
guest	the emulated system/program
host	the computer that runs the emulated virtual machine
machine code	code written as instructions that the processor understands
process virtual machine	an emulation of a single process
source code	code written in a high-level language
system virtual machine	an emulation of a complete computer system/operating system
virtualisation	the process of creating and running a virtual machine
virtual machine	software emulation of another computer/program/operating system

---

2. Theory

2.1 What is a virtual machine?

Computers vary in the hardware they use, and run different programs and operating systems. Sometimes we might want to have the functionality and capability of another computer system running on our own computer. It may be that the hardware for the other system no longer exists or is unavailable, or we may wish to have several versions of a computer running simultaneously on one hardware system.

We can use software to emulate the other computer system so that our computer can behave like that system. This emulation is known as a **virtual machine**.

The process of creating and running a virtual machine is known as **virtualisation**. The computer that runs the virtual machine is known as the **host**. The virtualised system is known as a **guest**. Virtualisation software, known as an **emulator**, creates a virtual implementation of the guest system on the host computer, providing the host with the functions and programs of the guest. The emulator translates the instructions of the guest computer so that the host computer can understand them.

The emulator also provides a bridge that allows the virtual machine to use the user interface, via the host operating system (I/O – input/output), allowing the user to interact with the software running on the virtual machine.
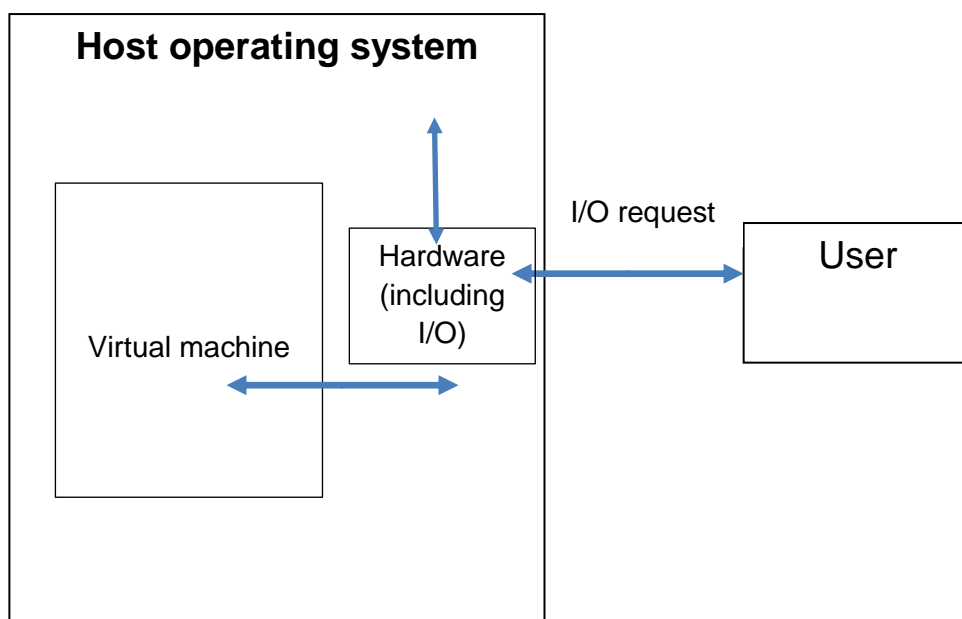
## 2.2 Types of virtual machine

2.2.1 System virtual machines

System virtual machines emulate the software of another computer system completely. They allow the host computer to have all the software functionality of the guest system. System virtual machines run the operating system of the guest system within the operating system of the host,

allowing programs supported by the guest operating system to be run on the host computer. As a result, system virtual machines are often used to allow the host computer to run an operating system written for another hardware platform.

Users running Apple operating systems often use a Microsoft Windows virtual machine to run the Windows operating system on Apple computers. Similarly, users of Apple- or Windows-based computers often run Linux as a virtual machine. The system virtual machine allows full use of the guest computer operating system and the programs that run on it.

Another use of system virtual machines is to provide access to programs that current operating systems no longer support. Many old DOS and Windows 3.1, 95 and 98 programs are not supported in Windows 8 and 10. Through using a system virtual machine for these old operating systems, such programs can still be run.

System virtual machines often require substantial processing power and memory. This is inevitable, as many modern operating systems require fast processors and plenty of memory. When emulating these operating systems, the more capable a host computer is, the faster and more capable the guest operating system will be. A host computer with limited resources may still be able to run a system virtual machine but it may have degraded performance.
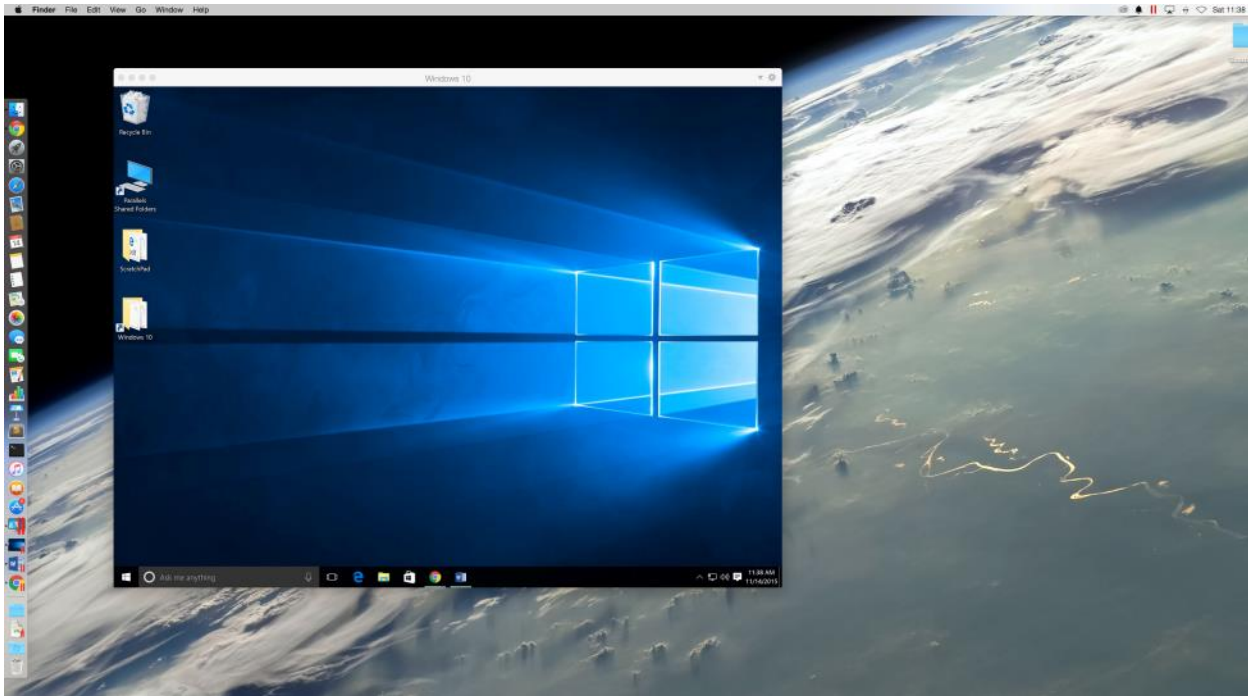

A Windows 10 system virtual machine running in an Apple OS X environment.

2.2.2 Process virtual machines

Process virtual machines emulate a single program. They are often used to support a programming language, allowing programs created in that language to run on a variety of systems. They are frequently used to emulate old video game systems and arcade games.
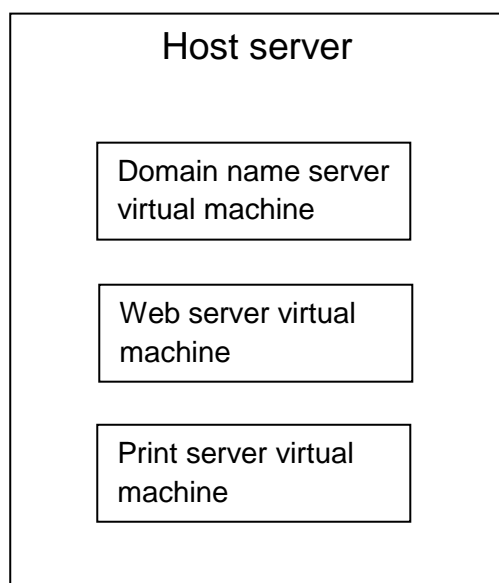
2.3 Virtual machines within virtual machines

It is possible to run one virtual machine within another. For example, a user wanting to play an emulated old arcade game could run the arcade game process virtual machine, within a Windows system virtual machine, on an Apple-based computer. The host computer's operating system provides an interface to the guest virtual machine, which in turn provides an interface to the process virtual machine within it.
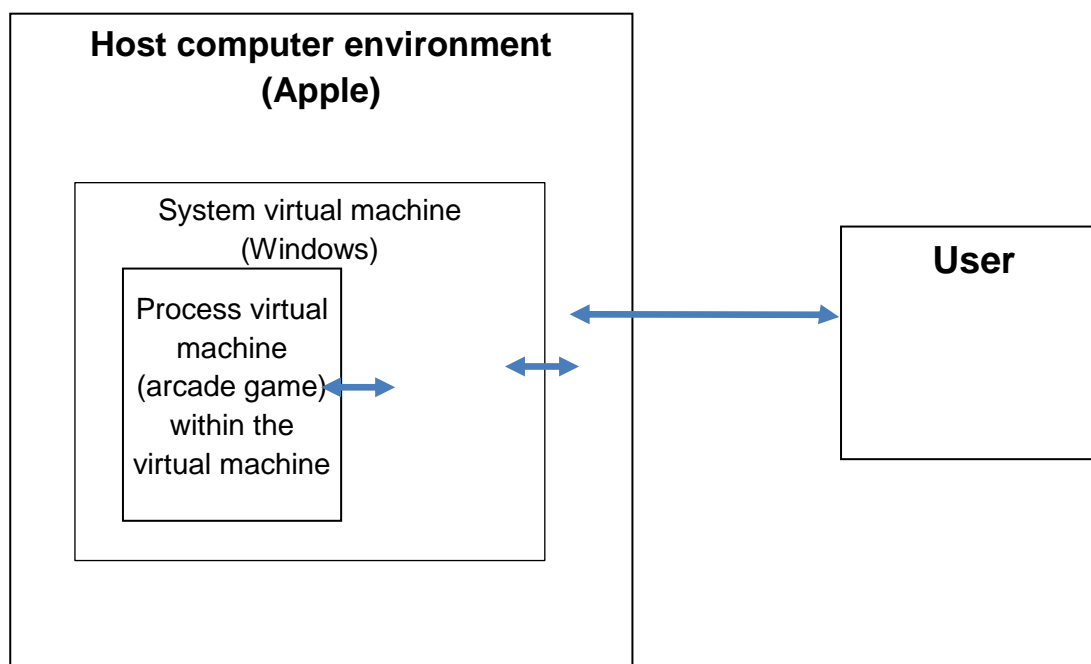
The number of virtual machines within virtual machines is limited only by the hardware's performance and available memory of the host computer.

## 2.4 Running virtual machines simultaneously

If a host system is powerful enough, several virtual machines can be run side-by-side on one computer. A good example of this is the increasing use of virtual machines to run several virtual servers, such as a domain name server, web server and print server, on one physical server. The host server maintains the virtual machines and allows them to act as though they are all separate computers.

2.5 The Java Virtual Machine

Many programs are written in high-level languages such as Visual Basic, C++, Python and Java. Before these programs can be run they need compiling. A **compiler** takes the high-level language **source code** and converts it into **machine code**. This machine code is a set of instructions specific to the processor architecture that the compiler is running on. As a result, this code can only be run on processors that share the same instruction set.

The Java compiler does not convert a high-level language into machine code, but into an intermediate language called **byte code**. This still needs translating into machine code before it can be run on a processor.

The Java Virtual Machine is a virtual machine that reads and translates byte code into different machine codes for many different processor architectures and operating systems. This means that Java programs can be run on any computer that has this virtual machine, including desktop PCs, laptops, tablets, smartphones and smart TVs.
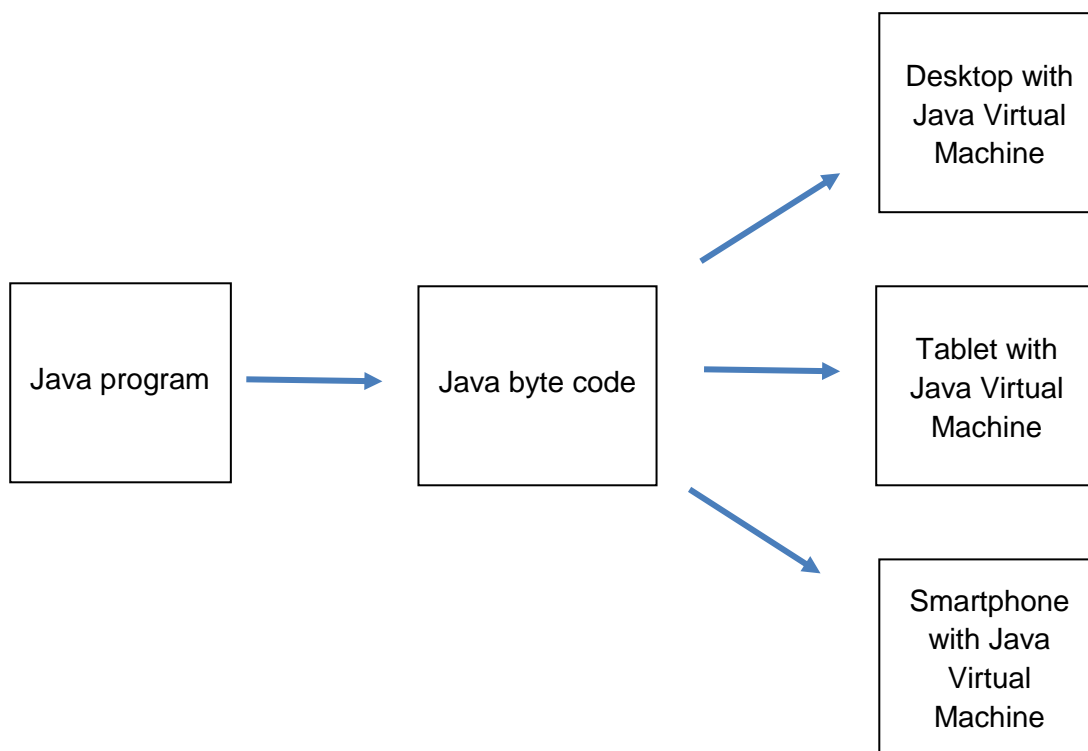
2.6 Benefits and limitations of virtual machines

Using virtual machines brings several benefits:

- They allow modern systems to use programs that run on obsolete or unsupported platforms
- They allow platform-independent programs to run on multiple platforms
- They allow one computer to behave as several computers simultaneously
- They can be easily backed up, copied and re-installed. This has particular benefits in corporate environments where network downtime can lead to loss of revenue. The virtual machines can be installed on another computer and be up and running again quickly.

Virtual machines also have limitations:

- They emulate software, not hardware. Users trying to get the full functionality from software for another platform may find that they still cannot use the software correctly. For example, an emulation of a driving video game that requires a steering wheel might not be able to operate the game correctly
- Emulation requires processing power and memory. Running a virtual machine may take a substantial amount of these resources, resulting in reduced performance from the host computer
- If the host system lacks sufficient resources, the virtual machine will also suffer from poor performance and be difficult to use.

---

3. Online resources

The following are useful resources for understanding virtual machines.

> The content of websites is dynamic and constantly changing. Schools are strongly advised to check each site for content and accessibility prior to using it with learners. Cambridge International Examinations is not responsible for the accuracy or content of information contained in these sites. The inclusion of a link to an external website should not be understood to be an endorsement of that website or the site's owners (or their products/services).

3.1 Websites

BBC Bitesize GCSE Computer Science – basic description of the types of virtual machine; the previous page describes the hierarchy of software languages and codes:

www.bbc.co.uk/education/guides/zgmpr82/revision/3

BBC Bitesize Higher Computing Science – virtual machines and emulators:

www.bbc.co.uk/education/guides/zsnbr82/revision/15

Introduction to virtual machines from National Instruments, including the benefits of virtualisation, particularly for engineering:

www.ni.com/white-paper/8708/en/

Description of the Java Virtual Machine:

www.careerbless.com/java/basics/JavaArchitecture.php

Slides from Michigan Technology University – useful mainly for diagrams of system and process virtual machines:

www.cs.mtu.edu/~zlwang/virtualmachines/slides/Introduction.ppt

4. Class and homework activities

4.1 Quiz

1. A virtual machine is:

    a. a hardware emulation of another system

    b. a software emulation of another system or program*

    c. a method of backing up data

2. Virtualisation software provides:

    a. a bridge between the host computer's hardware and the emulated software*

    b. a bridge between the guest computer's hardware and software

    c. a bridge between the host computer's operating system and the emulated software

3. A system virtual machine emulates:

    a. a single program

    b. the hardware and software of a guest system

    c. the operating system of a guest system*

4. A process virtual machine:

    a. emulates a single program*

    b. tells the operating system which processes to handle next

    c. emulates the operating system of a guest system

5. The Java Virtual Machine translates:

    a. source code into object code

    b. source code into byte code

    c. byte code into machine code*

4.2 Homework questions

1. Give three situations where a user might wish to use a virtual machine

2. Describe three advantages of using virtual machines

3. Explain how the use of virtual machines may allow software to run on multiple platforms without alteration

4. Describe a situation where using virtual machines might not be a good idea

5. A network manager is considering using virtual machines to host his network – explain the benefits of doing this