(Approx. 1116 words)

## Safe Experimenting

By Dick Maybach, Brookdale Computer User Group

www.bcug.com

n2nd (at) att.net

At the dawn of the personal computer age, life was simpler and more fun. Malware didn’t exist, nor did the Internet, and the most valuable thing on our PC was the BASIC program Hunt the Wumpus. We continually tried new software (usually discarding it immediately). Now our PCs contain vital data, such as family photos, financial records, tax returns, and email history, which makes many reluctant to experiment. While the Internet is full of free and cheap software, much of it is tainted, and we are hesitant to take a chance with anything. Moreover, modern operating systems are complex, making tinkering with their organizations hazardous. As a result, we are sitting in a huge virtual library, but afraid to take a book off the shelf.

You can restore the adventure to PCs by setting up an environment, separate from the one presently on your machine, where you can experiment safely. However, remember that an effective backup discipline is always your last and best defense. Let’s examine three such environments, virtual machines (VMs), dual-booting, and separate hardware.

No matter which environment you choose, you will need an operating system for it. If you use Windows, you have to purchase a separate copy, as the Microsoft license allows Windows to be installed in only one environment. Windows 10 is available (from Amazon) for as little as $50, which lets you achieve greatly increased security and yet stay in familiar surroundings. You also could use Linux, which opens up a whole new world of open-source software and which is generally malware-free, but the environment change may be traumatic.

The easiest separate environment to set up is a virtual machine, such as Oracle’s VirtualBox, but it requires competent hardware, at least eight Gbytes of RAM (16 is better), and 30 to 50 Gbytes of available disk space. When the VM is running, your hardware is supporting two environments, the one on your PC (called the host) and the one on the VM (called the guest). As a result, the guest environment may be noticeably slow, but less so if your hardware supports virtual environments. The key features on the CPU are VT-x on Intel and AMD-V on AMD processors, and these are now common, even on laptops. Be sure to check your VM documentation, as these features may be disabled in your BIOS.



Figure 1. VirtualBox Manager.

In operation, a VM looks like an application to the host; see Figure 1, which shows the VirtualBox manager. You use a virtual manager to add, delete, and configure VMs, and this PC has four, Windows 7, FreeDOS, Tails, and Windows 10. The figure also shows a summary of the VM running Windows 10.

The VM snapshot feature is useful for experimenters. Making a snapshot is equivalent to cloning the environment, and if the current experiment isn’t successful, you can restore things with a click or two. Also, since VMs are just files on the host when you back up the host, you also back up the VMs.



Figure 2. Host Desktop with a VM Running.

Figure 2 shows Windows 10 running in a VM on a Linux host. As you can see Windows has access to the Internet. Note also the file-manager window, which is looking at a directory on the host. Both Linux and Windows can access files in this directory, making it easy for the two to exchange information. You can also copy and paste between the two. However, these features require that you install the Guest Extensions to VirtualBox (see its documentation).

Before VMs became available, I used dual-booting for experimenting. This has the advantage of making all the resources of the host machine available to both environments; using VMs of course means that resources are shared between the host and the guest. The drawback is that setting up dual-booting requires some expertise and adds some risk. Here are the steps.

* Back up the system.
* Defrag the operating system to ensure that nothing is stored at the high addresses.
* Shrink the partition to make space for a second one above it. The second partition should contain at least 100 Gbytes. If you are short of space on your disk, you’ll have to install a second one.
* Install the second OS in the second partition.

This involves more risk than installing an application, so do your homework before attempting it. You also must be careful to back up the second environment separately.



Figure 3. Raspberry Pi Desktop.

The last and safest method of obtaining a test environment is to use a separate PC. Many of us have old, unused machines, making this approach very cheap indeed. Its main disadvantage is the space occupied. If you don’t have an unused PC or are short on space, consider a Raspberry Pi; it is model 4 that has as much power as a PC of not that many years ago; see Figure 3. If you share your PC display, keyboard, and mouse with the Raspberry, it uses almost no space. A KVM (Keyboard Video, Mouse) switch will allow you to do the sharing conveniently. Alternatively, you can set up a remote desktop to access the Pi from your PC, making the former appear as an application on the latter. It doesn’t even have to be in the same room; all both need is a connection to your home network. If you haven’t used a Raspberry Pi, you should first read the introductory material on its website, <https://www.raspberrypi.org/>. Setting one up is quite different than getting started with a new PC. Instead of a hard disk, it uses a microSD card, which you’ll buy separately and on which you must install the operating system that you’ll download from the Raspberry Pi website. The OS is a Linux variant, which probably involves yet more study, but the whole idea of experimenting is to learn.

Once you have hardware for your test environment, you’ll need an operating system. A VM and dual-booting give you the most flexibility, as you can use anything your host PC supports. With a Raspberry Pi, you’ll be running Linux. Your options on a second PC depend on its age; older units may not support Windows 10 for example. You might also consider switching to Linux, as many distributions support older hardware. It also has thousands of free applications available.

Regardless of how you choose to do your experimenting, continue to exercise care if you transfer files to your home PC, as they can carry malware. Also, when you use virtual machines and dual-booting, you are not completely isolated from your home environment. Cross-contamination, while unlikely, is not impossible.