Exercise 28.1 Making Sure **KVM** is Properly Set up

1. First check that you have hardware virtualization available and enabled:

   ```
   $ grep -e vmx -e svm /proc/cpuinfo
   ```

   where `vmx` is for **INTEL** CPUs and `svm` for **AMD**. If you do not see either one of these:

   - If you are on a physical machine, maybe you can fix this. Reboot your machine and see if you can turn on virtualization in the BIOS settings. Note that some IT personnel may make this impossible for "security" reasons, so try to get that policy changed.
   - You are on a virtual machine running under a hypervisor, and you do not have nested virtualization operable.

2. If you for either of these reasons you do not have hardware virtualization you **may** be able to run **virt-manager**, but with weak performance.
3. You need all relevant packages installed on your system. One can work hard to construct an exact list. However, exact names and requirements change with time, and most enterprise distributions ship with all (or almost all) of the software you need.
4. The easiest and best procedure is to run the script we have already supplied to you:

   ```
   $ ./ready-for.sh --install LFS301
   ```

   where we have done the hard work.
   Alternatively, on **RPM** systems you can do some overkill with:

   ```
   $ sudo yum|dnf|zypper install kvm* qemu* libvirt*
   ```

   It is not a large amount of storage space to do it this way.
   On **Debian** package based systems including **Ubuntu** you will have to do the equivalent with your favorite package installing procedure.

- Do not run **libvirtd** at the same time as another hypervisor as dire consequences are likely to arise. This can easily include crashing your system and doing damage to any virtual machines being used.
- We recommend both **stopping** and **disabling** your other hypervisor as in:

  ```
  $ sudo systemctl stop vmware
  $ sudo systemctl disable vmware
  ```

  or

  ```
  $ sudo systemctl stop vboxdrv
  $ sudo systemctl disable vboxdrv
  ```

