BUILDING A DOCKER-BASED VIRTUAL MACHINE

**The instructions below were used on a Mac, but a Windows-based installation should be quite similar*

CONTAINER SETUP

Download the Docker Desktop app from the Docker web site. Install and configure it with whatever options you like (e.g., max memory for the VM). For the sake of convenience, you may just want to use the various defaults, with no explicit configuration.

Start the Docker Desktop app, if necessary.

Now create a folder called `Docker` somewhere in your local file system. The exact location doesn't matter.

Copy the simple `Dockerfile` provided on the class website into this folder (no other files should be placed in the Docker folder)

***IMPORTANT**: Please take note of the following:*

1. *The file must be called `Dockerfile`, NOT `Dockerfile.txt`. Sometimes the .txt extension is added automatically when you download or save a text file. The `Docker` application will not read a file called `Dockerfile.txt` and your installation will fail. Keep in mind that File Managers often hide file extensions by default, so you should check the name of the file from the command line (i.e., by using the "ls" command on Mac/Linux, or with "dir" on Windows. If you see `Dockerfile.txt`, please rename the file to `Dockerfile` before proceeding with the next step.*

2. *If you are working on a Window's computer, you may want to copy and paste the Dockerfile content into a new text document (e.g., using notepad) and then save that file locally. This is because the config file was created on a Mac laptop that uses different end-of-line characters than the ones used on Windows. The result is that Unix/Linux text files may not always be read properly by Windows-based applications. This may give odd docker errors – about invalid continuation lines - when you try to do the build. Saving the file locally will ensure that the proper end-of-line characters are used.*

3. *When transferring the config file to your computer, make sure that you actually download the file (e.g., by using whatever download or save option your operating system provides). Do not, for example, drag the config file from a web browser and drop it on your desktop. This may just create a "link" to the config file, rather than a copy of it. Trying to then execute docker commands on a link file will almost certainly fail.*

Use the relevant *Terminal* app to open a command line shell and navigate to the Docker folder that you just created. If you now use the 'ls' command (or 'dir' on Windows) to display the contents of the folder, you should see the Dockerfile listed.

Enter the following from the command line:

```
docker build -t comp6411_image .
```

(Note: the trailing period '.' is required)

This will pass the Dockerfile contents to the Docker Desktop server (as noted above, you must start the Docker Desktop app before running the docker commands). Docker will create a new image, labeled with the tag "comp6411_image", using the "gcc" base image from the docker online repository. The RUN command in the Dockerfile will install the base packages and perform any other actions required, including adding some additional language packages. This process will take several minutes to complete. Just sit back and let everything run.

If all goes well, you will eventually get a command prompt again. We will now create an interactive container from the downloaded image, mounting the relevant folders from the local file system. This container will be built from the `comp6411_image` and will be called `comp6411`.

We simply do the following:

1. Identify a "target" folder from the host file system where your source files will be stored (you can eventually create as many sub-folders within this parent folder as you like). For example, the full path to this folder might be something like:

   `/Users/myname/Documents/comp6411  (where myname is your user name)`

2. Specify a path location that will be used to mount this folder inside the Linux distribution. For example, something like the following works just fine:

   `/root/comp6411`

3. From the Terminal app, enter the command below - **as a single line** - substituting the appropriate values for the "source" path and "target" path (as specified in the previous steps)

docker container create -i -t --name comp6411 --mount
type=bind,source="/Users/myname/Documents/comp6411",target=/root/comp6411 comp6411_image

(note: if your path name contains spaces, you must enclose the full path in quotes (e.g., "/this/is/my path")

At this stage you should now have a valid Docker container that may be used throughout the semester (the dashboard in the Docker Desktop app should list the new image and container). The preceding steps do not need to be repeated.

RUNNING YOUR DOCKER CONTAINER

You can now start the container from the command line, as required:

```
docker container start –a –i comp6411
```

This will give you a command line shell inside the Linux distribution. Your source folder from the host system will be mounted at the target location specified in the command above (e.g., /root/comp6411).

(Note: Typing "exit" at the command line will terminate your session.)

You can now compile your source code from the Linux command line. Woo hoo!