tips.txt

- When working out of the VM, avoid using spaces in file/folder names.
  If you transfer files/folders to the VM, make sure you remove all
  spaces; and then transfer them.
    - Avoid using spaces because they need to be escaped with a
      backslash (\ ). Dealing with this is cumbersome and tedious.

- The VM does not support scrolling. If the output for some command
  gets cutoff (i.e. dmesg), just redirect it to a text file and then
  use nano to view the text file (i.e. dmesg > file.txt).
    - Alternatively, the keys "page up" and "page down" also work.

- Transferring files between host machine and virtual machine (VM):
    - If you are using Windows, please use winscp to transfer files
      between the machines. winscp is very easy to use and there are
      instructions on how to set it up.
    - If you are using a *nix based operating system, please use scp
      and sftp to transfer files to and from the virtual machine. If
      you prefer a GUI, you can use Cyberduck.
        - Transferring files from host to VM:
            scp -P 2222 FILE osc@127.0.0.1:~   # For Files
            scp -P 2222 -r FOLDER osc@127.0.0.1:~ # For Folders
            # Note: Use this command on your HOST's terminal
            # Note: The password is "osc", without the quotes
        - Transferring files from VM to host:
            sftp -P 2222 osc@127.0.0.1
            # Use `cd` to navigate the directories inside the VM
            # Use `ls` to list all files and folders
            # Use `get FILE` to retrieve the FILE
            # Use `get -r FOLDER` to retrieve a directory/folder
            # Note: Use this command on your HOST's terminal
            # Note: The password is "osc", without the quotes

- Accessing the VM from the host machine is possible via `ssh`.
    - From a terminal, type the following:
        ssh -p 2222 osc@127.0.0.1
        # Note: Use this command on your HOST's terminal
        # Note: The password is "osc", without the quotes
    - This will enable scrolling, copy/pasting, and a variety of other
      functions.

- If you make one really good Makefile, you can use it over and over
  again for subsequent projects, assignments, labs, etc.
    - Design your Makefile to be adaptable (i.e. Easily mutable)