CS:APP2e Unix FAQ

Here are some questions we get a lot from students in our 15-213 class.

- **Q:** *How come the Unix shell can't find my program "foo"? I know it's there in the current directory!*
  A: You need to tell the shell to look for your program in the current directory (denoted by a "."). You can do this *explicitly* by typing the full pathname,

  ```
  linux> ./foo
  ```

  or *implicitly* by adding a dot (.) at the end of your search path (see next question).

- **Q:** *How do I add a dot to the end of my search path?*
  A: It depends on what shell you are using. If you are using `tcsh`, add the following line to your `~/.tcshrc` file:

  ```
  set path=($path .)
  ```

  If `csh`, add the following line to your `~/.cshrc` file:

  ```
  set path=($path .)
  ```

  If `bash`, add the following line to your `~/.bashrc` file:

  ```
  PATH=${PATH}:.
  ```

  If `sh`, add the following line to your `~/.profile` file:

  ```
  PATH=${PATH}:.
  ```

  If `ksh`, add the following line to your `~/.kshrc` file:

  ```
  PATH=${PATH}:.
  ```

  The change to your path will go into effect the next time you run a new shell process, for example, when you log in. If you don't want to wait, you can type the appropriate line directory to the shell. For example, if you are running tcsh on a Linux box, type:

  ```
  linux> set path = ($path .)
  ```

  Caution: Never add a dot to the beginning or the middle of your path. It will make you vulnerable to an exploit where the attacker places a bogus set-userid enabled binary in your home directory.

- **Q:** *So how do I know what shell I am running?*
  A: Type "`echo $SHELL`". For example:

  ```
  linux> echo $SHELL
  /usr/local/bin/tcsh
  ```