
IT 1100 : Introduction to Operating Systems

Chapter 11 Modifying the Environment

Commands Covered

“The shell maintains a body of information during our shell session called the environment. Data stored in the environment is used by programs to determine facts about the system’s configuration. While most programs use configuration files to store program settings, some programs will also look for values stored in the environment to adjust their behavior. ”

Commands Covered

- `printenv` - print part or all of the environment variables
 - `set` - prints both shell and environment variables
 - `export` - export environment to subsequently executed programs
 - `alias` - create an alias for a command or set of commands
-

System Config vs Single User Config

System configuration files are located in the `/etc` directory

User configuration files are located in users home directory

In the textbook chapter 11 still uses `nano` for its examples - you can follow the examples as written and get some `nano` experience or you can use `vim`.

The Environment

The shell stores two basic types of data in the environment, though, with `bash`, the types are largely indistinguishable. They are environment variables and shell variables. Shell variables are bits of data placed there by `bash`, and environment variables are basically everything else. In addition to variables, the shell also stores some programmatic data, namely aliases and shell functions.

Environment Variables

An environment is an area that the shell builds every time that it starts a session. The environment creates variables that will hold different system properties.

- `set` (shows both shell and env)
 - `printenv` (will give you a listing of currently defined environment variables)
 - `echo $PATH`; (What is the path variable?)
-

Environment Variable Types

These are summarized in Table 11-1 in the text.

- `DISPLAY`
- `EDITOR`
- `SHELL`
- `HOME`
- `LANG`
- `PATH`
- `PS1`
- `PWD`

- `TERM`
- `USER`

We can use the echo command to display an environment variable but using a \$:

- `echo $PWD`
 - `echo $USER`
-

Aliases

- See aliases by entering `alias` without anything else
 - `alias p='pwd ; ls -CF'`
 - `alias foo='echo hello'`
-

Establishing the Environment

When we log on to the system, the bash program starts, and reads a series of configuration scripts called startup files, which define the default environment shared by all users.

Types of sessions

- A login shell session is one in which we are prompted for our username and password; when we start a virtual console session, for example.
 - A non-login shell session typically occurs when we launch a terminal session in the GUI .
-

Startup files for login shells

- `/etc/profile` # global for all users
 - `~/.bash_profile` # individual users startup, can override above
 - `~/.bash_login` # read if above is not found
 - `~/.profile` # if neither of 2 above is found, read this (default ubuntu)
 - `~/.bash_logout` #run these commands when logging out
-

Startup files for non-login shells

- `/etc/bash.bashrc` # global for all users
 - `~/.bashrc` # personal user, override above
 - Look in these files to see what we have
-

More Shell stuff

To change environment variables:

- `PATH=$PATH:/getstuff/bin ; export PATH` (put in `.bashrc` file)
 - Setting your prompt [site](#)
 - More on this in next slides
-

After changing bash files

- Either logout/login
 - `source .bashrc`
-

Textbook Time

- [WES-11 Modifying the Environment](#)