The Bourne Again shell (bash) is based off the Bourne shell (sh) written by Steve Bourne in the AT&T laboratory. Other shells that are extended or compatible or similar to bash are dash, korn, and POSIX shell.

A shell runs startup files to initialize itself when it starts. The type of files depends on whether the shell is a login shell, interactive shell or noninteractive shell.

Login Shell:

The first shell that displays a prompt.

/etc/profile – first command executed, establishes systemwide default characteristics for running bash, then followed by ~/.bash\_profile, to logout, ~/.bash\_logout

Interactive (non-login) shell:

Inherit from the files used to start up login shell

Noninteractive shell:

Whatever that is in the previous two, not used in this shell, but if this shell is forked by a login shell then it inherits from the startup files.

.(dot) runs startup file in current shell

Bash uses symbols – (, ), [, ], and $

Same with standard output, shell will direct standard error to the screen unless redirected. System buffers standard output, but not standard error.

File descriptors is a place where a program sends its output to and get its input from. When a program starts there are 3 files descriptors: 0, 1, 2 – standard input, output, error. Use numbers with < and > for redirection. &> means redirecting standard output and error to a single file.

Writing and executing shell scripts:

Shell scripts are files that hold an executable command. Usually they are not executable, so use chmod to change the permissions.

#! Specifies a shell, so #!/bin/bash means that bash should run the script.

# begins a comment.

You can use bash to execute a script, but it’ll run more slowly than if you just gave the execution permission to yourself.

You must separate commands:

; and /n can separate commands. /n will initiate execution, ; will just separate them and create a sequence of commands. Whitespaces are not necessary.

| and & separate and do sth else. Doesn’t start execution but does change some aspect of how the command functions. & causes the shell to execute in the background. Pipelining is sequential, $ x | y | z means that z won’t finish till y is done, and y won’t finish till x is done. $ d & e & f runs d and e in the background and f in the foreground.

&& and || are Boolean operators. Example:

Mkdir bkup && cp -r src bkup – make a directory called bkup and if that succeeds copy contents of src into bkup.

Mkdir bkup || echo “mkdir failed” >> /tmp/log – if mkdir fails execute echo command.

Stack manipulation: dirs, pushd, popd – display, push, pop.

Parameters and variables:

Variables are only available in the shell they are created in by default, unless you use export, then it goes from shell variable to environment variable. Syntax: VARIABLE=variable (no whitespace allowed)

$ person=max

$ echo $person – max

$ echo “$person” – max

$ echo ‘$person’ – person

$ echo \$person – person

To remove a variable use unset, to remove value but not variable, assign null to variable. Use readonly to make variable constant, it must have a value assigned to it beforehand, cannot remove readonly attribute (+r not allowed). Use declare if u need to use -r, -x, etc. with variables.

Keyword variables:

HOME, PATH, MAIL, PS1 (User Prompt primary), PS2 (User Prompt secondary), PS3(Menu Prompt), PS4 (Debugging Prompt), IFS (Separates Input Fields), CDPATH, etc.

Locale:

Specifies the way locale-aware programs display certain kinds of data. Localization and internalization go hand in hand. Internalization is the process of making software portable to multiple locale, and localization is the process of adapting software so that it meets the language, cultural and other specific requirements of the locale environment.

Process Structure:

Parents, children and root

Parent process forks(spawns) a child process, which can then fork more children

Fork() – system call

Init daemon – the ancestor of all processes, a spontaneous process

Every process has an ID called PID, stays the same as long as process exists