Unit: IV
Shell Introduction and Shell Scripting
Lecture: 4
Shell Variables and System Calls

Variables are used to store data and configuration options. In Linux shell scripting we are using two types of variables:

- System Defined Variables and
- User Defined Variables

A variable in a shell script is a means of referencing a numeric or character value.

A shell script doesn't require you to declare a type for your variables.

System Defined Variables: These are the variables which are created and maintained by Linux (operating system) itself. Generally, these variables are defined in CAPITAL LETTERS.

To view the system defined variables, use the command "$set".

Some of the system defined variables are:

System Defined Variables	Meaning
BASH=/bin/bash	Shell Name
BASH_VERSION=4.1.2(1)	Bash Version
COLUMNS=80	No. of columns for our screen
HOME=/home/linuxtechi	Home Directory of the User
LINES=25	No. of columns for our screen
LOGNAME=LinuxTechi	LinuxTechi Our logging name
OSTYPE=Linux	OS type
PATH=/usr/bin:/sbin:/bin:/usr/sbin	Path Settings
PS1=[\u@\h \W]\$	Prompt Settings
PWD=/home/linuxtechi	Current Working Directory
SHELL=/bin/bash	Shell Name
USERNAME=linuxtechi	User name who is currently login to system

To print the value of above variables, use echo command:

#echo $HOME

#echo $USERNAME

User Defined Variables: These variables are defined by users. A shell script allows us to set and use our own variables within the script. Setting variables allows you to temporarily store data and use it throughout the script, making the shell script more like a real computer program.

User variables can be any text string of up to 20 letters, digits or an underscore character.

User variables are case sensitive, so the variable Var1 is different from the variable var1.

Values are assigned to user variables using an equal sign. No spaces can appear between the variable, the equal sign, and the value.

e.g.

var1 = 10

var2 = -57

var3 = testing

The shell script automatically determines the data type used for the variable value. Variables defined within the shell script maintain their values throughout the life of the shell script but are deleted when the shell script completes.

Just like system variables, user variables can be referenced using the dollar sign.

System Calls: A system call is the programmatic way in which a computer program requests a service from the kernel of the operating system it is executed on. A system call is a way for programs to interact with the operating system.

A computer program makes a system call when it makes a request to the operating system's kernel. System call provides the services of the operating system to the user programs via Application Program Interface (API). It provides an interface between a process and operating system to allow user-level processes to request services of the operating system.

System calls are the only entry points into the kernel system. All programs needing resources must use system calls.

Services Provided by System Calls:

- Process creation and management
- Main memory management
- File Access, Directory and File system management
- Device handling (I/O)
- Protection
- Networking, etc.

Types of System Calls:

There are five different categories of system calls:

- **Process Control:** end, abort, create, terminate, allocate and free memory.
- **File Management:** create, open, close, delete, read file etc.
- Device management
- Information maintenance
- Communication

Examples of Windows and Unix System Calls –

	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitlializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()