MODULE 7 :
Developer's Tools

Environment Variables - Basics

- Environment Variables are variables that store information about the system
- They can be used to store data, set configuration options and customize the shell environment under Linux
- Can be divided into two types:
  - System Environment Variables
  - Local variables

Environment Variables - Basics

System Variables:

- **Standard Names**
  - Used by the Shell
  - Normally they are All Caps
  - More can be added by the users for their usage

- **Local Variables**
  - User selected names
  - Local to a shell (not passed to children shells or programs)
  - Convention is to avoid all caps to differentiate them

Environment Variables - Usage

- Examples of use of Environment Variables (not a full list) :
  - Configure look and feel of shell such as colors and bash prompt
  - Time zone, host name,…
  - Search path for executables, or any types of files
  - Default values for some system configurations
  - Some configuration options for specific programs

Process Environment

- Linux does not maintain or store a global set of environment variable for the system
- Each running program (process) will have its own environment settings
- This means different processes may have different environment settings
- The environment settings for each running process in the system can be listed by viewing the file /proc//environ
    - Where pid is the Process ID

Process Environment

Process receive their environments settings by:

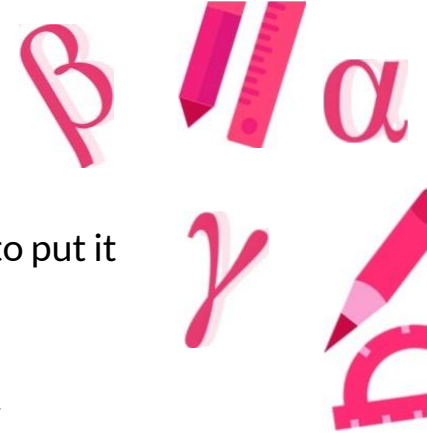
- **By inheritance**
  - Each process will have a parent process that started it
  - The child process inherits the environment settings of its parent process
  - that each program (process) that is started inside the shell, is a child of that shell, hence processes started from the shell, inherit the shell environment settings
  - a non-login shell is a child of a login shell, hence it inherits its environment settings at startup
  - local variables are not inherited to child shells or processes

Process Environment

Process receive their environments settings by:

- **By Startup Scripts**
  - Some programs source some scripts at startup
  - These scripts may include some environment settings that is added to the process settings inherited from its parent
  - We have already discussed this for login/non-login shell startup
  - Login Shells /etc/profile ~/.bash-profile or ~/.bash-login or ~/.profile
  - Non-Login Shells /etc/.bashrc or /etc/bash.bashrc ~/.bashrc
  - GUI Applications (applications started from the GUI) ~/.xinitrc

/etc/profile



- To add settings that will apply to all shells, and all users… we need to put it in /etc/profile
- In most distributions, it is preferred not to edit /etc/profile directly
- To enable that, /etc/profile has a loop that sources all scripts with extentions *.sh in the folder /etc/profile.d
- Accordingly, all we need to do is to put our settings in a new script file inside this folder and call it something.sh then make it executable
- Our script will be called from /etc/profile and hence our settings will be read by login shells, and inherited by non-login shells

export

- To Set a local variable in the shell $ My_Var=5 This way My_Var will not be inherited to any child or process of the current shell
- To Convert it into an Environment Variable $ export My_Var This way My_Var will be inherited to any child shell or process of current shell
- To bring it back to be just a local variable $ export -n My_Var
- To reset an Environment variable $ export My_Var=
- To Completely remove the variable $ unset My_Var

List Environment variables

- set
- Printenv
- env

Common Environment variables: PATH

- It is a list of directories separated by a colon ":" /home/tom/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
- This list represents the search path for commands and binaries, when you issue a command
- To show the current search path $ echo $PATH
- To add a folder to the end of the path $ export PATH=$PATH:/usr/bin
- To add a folder to the beginning of the path $ export PATH=/bin:${PATH}

Common Environment variables: PS1

- Responsible for setting the shell prompt
  - \u →username
  - \h →hostname
  - \W →current working directory

- Example $ export PS1=[\u@\h \W]\$

Common Environment variables: SHELL

- Contains the path to the login shell

- Example:
  - $ echo $SHELL /bin/bash

Common Environment Variables:

- EDITOR
- TERM
- HOME
- HOSTNAME

programs from command line: C

- Create files using any editor (vi, gedit etc..) with .c extension
- Compile by following command:
  - gcc filename.c –o output
  - where <output> is the output filename
- run the code by following command:
  - ./output

programs from command line: C++

- Create files using any editor (vi, gedit etc..) with .C++ extension
- Compile by following command:
  - g++ filename.c++ –o output
  - where <output> is the output filename
- run the code by following command:
  - ./output

programs from command line: java

- Create files using any editor (vi, gedit etc..) with .java extension
- Compile by following command:
  - javac filename.java
- run the code by following command:
  - java filename

Thank You!