Environment variables

Unix

What is Environment variable?

- **Environment variables** allow you to customize how the system works and the behavior of the applications on the system.
- For **example**, the **environment variable** can store information about the present working directory or the **path** to executable files, or the user’s home directory.
- They are called “**environment variables**” because most of them affect the way your **Unix** shell works for you.
- You can display a single **variable** with a simple echo command i.e. to display value of PATH env variable you can type command ***echo $PATH***
- You can get the entire list of env variable with the help of command ***env***.

How to set environmental variables

- You can set env variable as following:
  - **export** *name of env variable =value*

```
E.g.  export HOME=/home/trainee
$ cd
$ pwd
   /home/trainee
```

- You can remove value using **unset** command

```
$ unset HOME
$ cd /tmp
$ cd
   -bash: cd: HOME not set
```

- Environment variables can be changed or unset, but they can also be augmented. If your search path is set up in /etc/profile, you can redefine it in your local .profile or add to it with a command like this:

```
PATH=/bin:$PATH:/apps/bin
```

Customizing Your Working Environment

- When you log in, your system reads the initialization files and uses the environment variables to configure your system. By setting the environment variables, you can customize your system to make it easier and more efficient to do your work.
- You can modify the **environment variables** in your system **initialization files** to control and adjust many aspects of your working environment.

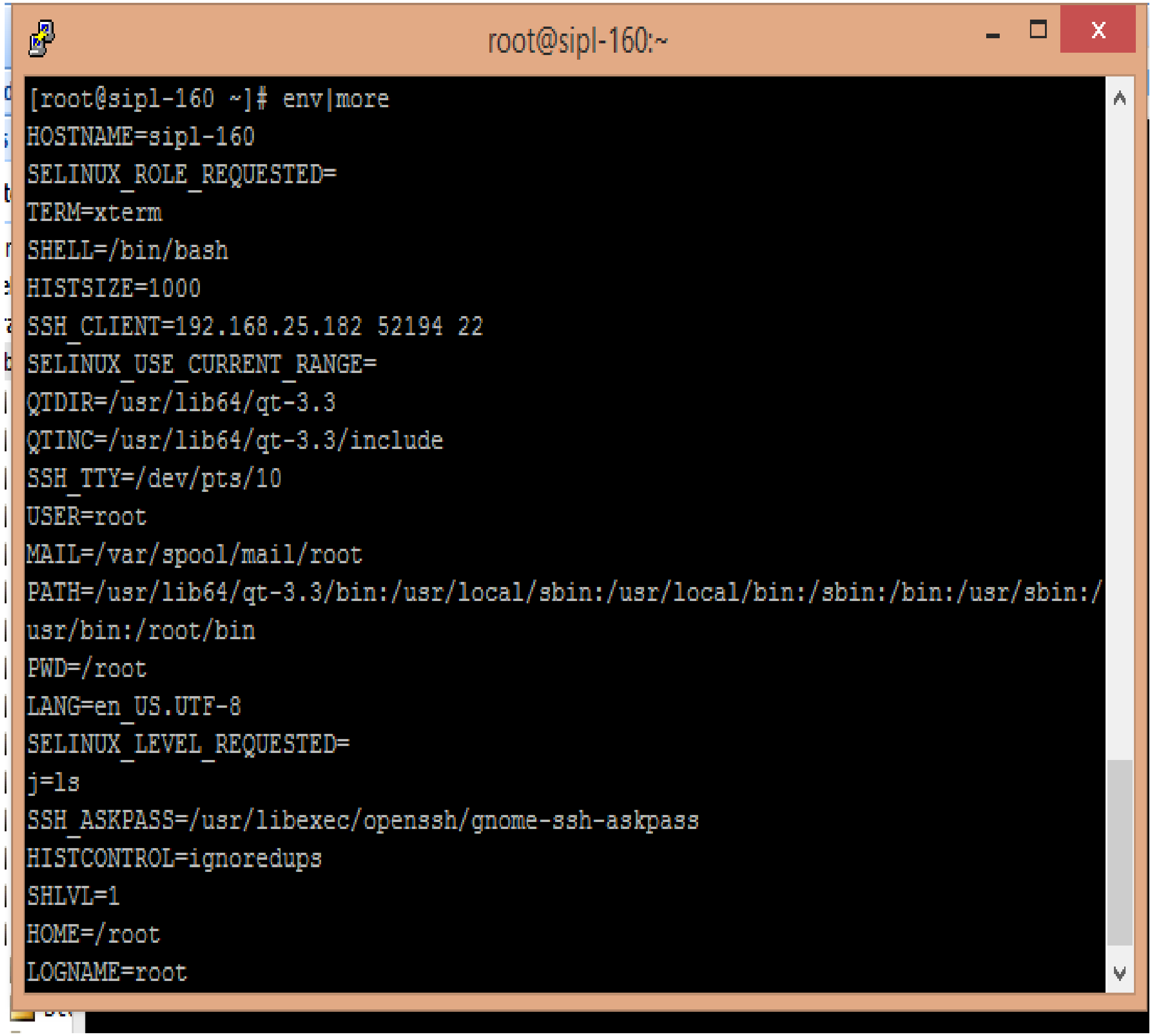
Modifying Initialization Files

- The Bourne shell is the default shell for the most of Unix operating environment, but you can also use the C shell or Korn shell. Each of these shells has its own initialization file (or files).
- To determine your default shell (your **login shell**), Type **echo $SHELL**.
- Refer to the following list to identify your default shell.
    - /bin/sh – Bourne shell
    - /bin/bash – Bourne Again shell
    - /bin/csh – C shell
    - /bin/ksh – Korn shell
    - /bin/tcsh – TC shell
    - /bin/zsh – Z shell
- Regardless of the shell you are using, when you first log in your system generally runs the system profile file, /etc/profile. This file is generally owned by the system administrator and is readable (but not writable) by all users.

- After your system executes the system profile, it runs the **user profile**. The user profile is one (or more) initialization files that define your working environment. For example, if you're in the CDE environment your system checks this file (or set of files) each time you start a new terminal or window.
- Depending on which shell is set up as your default, your user profile can be one of the following:
    1. .profile (for the Bourne and Korn shells)
    2. .bash_profile (for the Bourne Again shell)
    3. .login and .cshrc (for the C shell)
    4. .tcshrc and .cshrc (for the TC shell)
    5. .zlogin and .zshrc (for the Z shell)
- Your user profile file(s) is located in your home directory and enables you to configure your working environment to your preference.

Setting Environment Variables

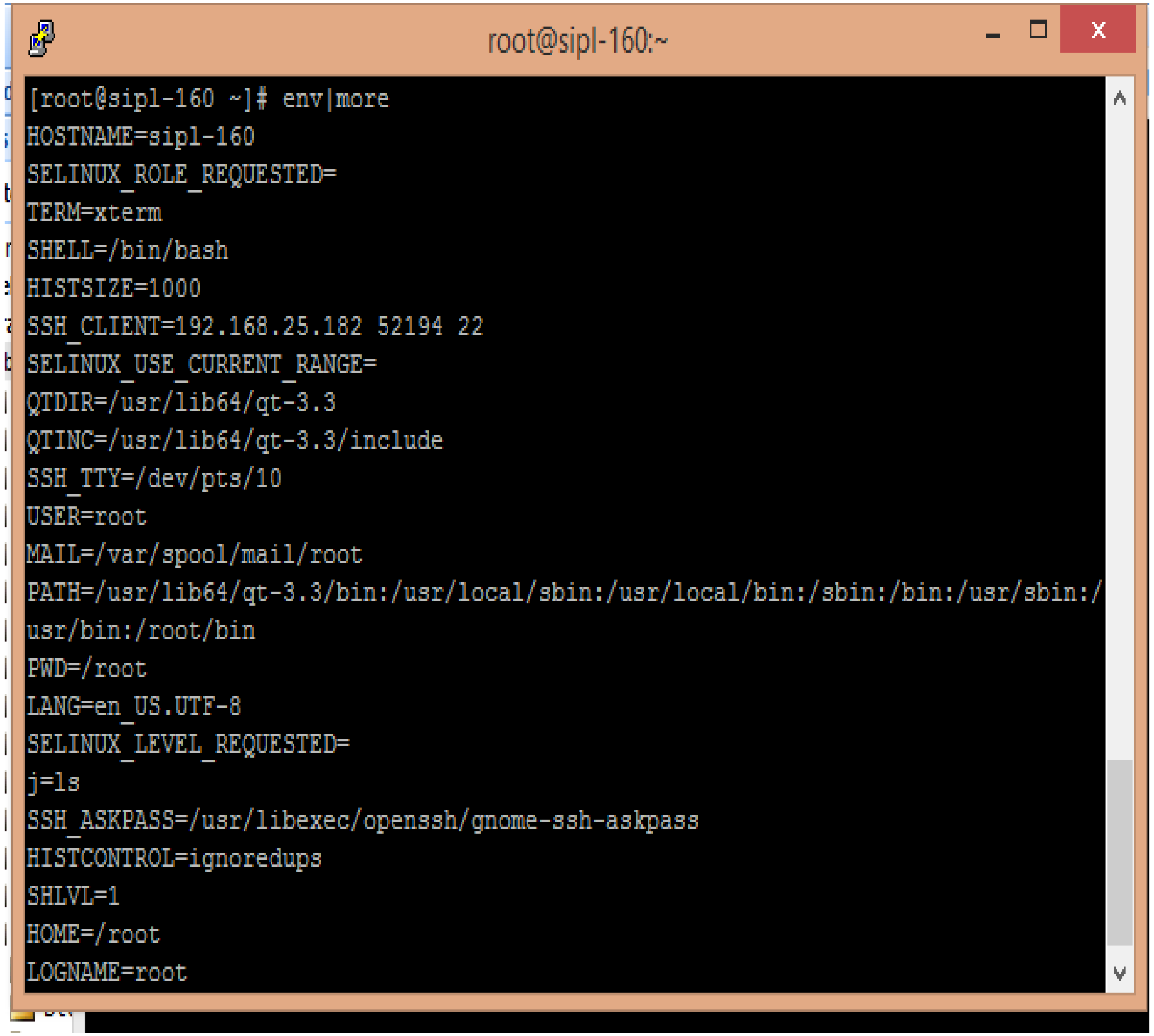
- Your system sets up your system environment by using a set of specifications that are defined in the initialization files.
- If you want to temporarily modify your environment for the current work session, you can issue commands directly at the command prompt.
- However, if you want to modify your working environment on a more permanent basis, you can store “permanent” environment variables in the appropriate user profile files.
- To display the environment variables that are currently set on your system, use the env command.



```
[root@sipl-160 ~]# env|more
HOSTNAME=sipl-160
SELINUX_ROLE_REQUESTED=
TERM=xterm
SHELL=/bin/bash
HISTSIZE=1000
SSH_CLIENT=192.168.25.182 52194 22
SELINUX_USE_CURRENT_RANGE=
QTDIR=/usr/lib64/qt-3.3
QTINC=/usr/lib64/qt-3.3/include
SSH_TTY=/dev/pts/10
USER=root
MAIL=/var/spool/mail/root
PATH=/usr/lib64/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/
usr/bin:/root/bin
PWD=/root
LANG=en_US.UTF-8
SELINUX_LEVEL_REQUESTED=
j=ls
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
HISTCONTROL=ignoredups
SHLVL=1
HOME=/root
LOGNAME=root
```

Setting Environment Variables

- Your system sets up your system environment by using a set of specifications that are defined in the initialization files.
- If you want to temporarily modify your environment for the current work session, you can issue commands directly at the command prompt.
- However, if you want to modify your working environment on a more permanent basis, you can store “permanent” environment variables in the appropriate user profile files.
- To display the environment variables that are currently set on your system, use the env command.



```
[root@sipl-160 ~]# env|more
HOSTNAME=sipl-160
SELINUX_ROLE_REQUESTED=
TERM=xterm
SHELL=/bin/bash
HISTSIZE=1000
SSH_CLIENT=192.168.25.182 52194 22
SELINUX_USE_CURRENT_RANGE=
QTDIR=/usr/lib64/qt-3.3
QTINC=/usr/lib64/qt-3.3/include
SSH_TTY=/dev/pts/10
USER=root
MAIL=/var/spool/mail/root
PATH=/usr/lib64/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/
usr/bin:/root/bin
PWD=/root
LANG=en_US.UTF-8
SELINUX_LEVEL_REQUESTED=
j=ls
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
HISTCONTROL=ignoredups
SHLVL=1
HOME=/root
LOGNAME=root
```

User Profile

- The following is a partial list of environment variables that you can include in your user profile. Your current shell determines the syntax for defining environment variables.
  - HISTORY – Sets the number of commands available to the history command.
  - HOME – Defines the absolute path to your home directory. The system uses this information to determine the directory to change to when you type the cd command with no arguments.
  - LOGNAME – Defines your login name. The default for this variable is automatically set to the login name specified in the passwd database as part of the login process.
  - PATH – Lists, in order, the directories that the system searches to find a program to run when you type a command. If the appropriate directory is not in the search path, you have to type it or else type the complete path name when you enter a command.
  - PS1 – Defines your command prompt. The default prompt for the Bourne, Bourne Again, and Korn shells is the dollar sign ($). The default prompt for the C, TC, and Z shells is the percent sign (%). The default prompt for root in either shell is the pound sign (#).
  - SHELL – Defines the shell that is used by vi and other tools.
  - TERM – Defines the terminal you're currently using. When you run an editor, the system searches for a file with the same name as the definition of this variable. The system first searches the path (if any) referenced by the TERMINFO variable, and then the default directory, /usr/share/lib/terminfo, to determine the characteristics of the terminal. If a definition is not found in either location, the terminal is identified as “dumb.”
  - TZ – Defines the time zone for your system clock.

Setting the PATH Variable

- The PATH environment variable is used by shell to locate commands within the UNIX directory hierarchy. By setting the PATH, you create a fixed set of colon separated directories that the system always searches whenever you type the name of a command.
- For example, if you have no PATH variable set and you want to copy a file, you need to type the full path name for the command, /usr/bin/cp.
- However, if you have set the PATH variable to include the directory /usr/bin, then you can simply type cp and your system will always execute the command.
- This is because your shell searches for the cp command in every directory that is named in the PATH variable, and executes it when it is found. You can significantly streamline your work by using the PATH variable to list the commonly used UnixOS command directories.

- For the Bourne, Bourne Again, and Korn shells, you can specify the PATH variable in your user profile file (in your home directory) by using the following syntax.
  - ***PATH=.:/usr/bin:/home/bin***
- For the C, TC, and Z shells, you can specify the PATH variable in your user profile file (in your home directory) by using the following syntax:
  - ***set path=(/usr/bin home/bin .)***

Command Aliases

- Command aliases are helpful shortcuts for commands you often type. For example, the default setting for the remove command (rm) does not ask for confirmation before it removes files.
- Sometimes this default is inconvenient, as a typing error can remove the wrong file. However, you can use the alias variable to change this setting by editing your user profile file.
- In the C and TC shells, add the following line to your user profile file.
  - **alias rm 'rm -i‘**
- In the Bourne Again, Korn, and Z shells, add the following line to your user profile file.
  - **alias rm='rm -i‘**
- With this line in your user profile file, typing **rm** is now the same as typing **rm -i**, which is the interactive form of the **rm** command. You will then always be asked to confirm the command before any files are deleted.