Variáveis de ambiente

Definição:

São variáveis que guardam informações sobre preferências pessoais usadas por programas para que eles peguem dados sobre seu ambiente sem que você tenha que passar sempre os mesmos dados.
As variáveis de ambiente normalmente são escritas em letras maiúsculas.

Imagine você ter que digitar o caminho completo de um comando para poder executá-lo ou então a todo momento que você fazer paginamento de uma arquivo você passar o tamanho e o qual terminal usado... Tudo isso seria muito chato e trabalhoso!!!

Se não existisse uma variável de ambiente chamada **PATH** você teria que digitar todo o caminho do comando para listar por exemplo:

$ /bin/ls

Quando você digita o comando **ls** , o sistema busca esse comando em algum diretório que esteja na variável **PATH**.

Para ver o conteúdo da variável **PATH**:
$ echo $PATH
/usr/local/bin:/usr/bin:**/bin**:**/**usr/bin/X11:/usr/games

Veja que o **ls** está em **/bin** com o comando **which**:
$ which ls
/bin/ls

Curiosidade: Você algum dia já precisou digitar **./** antes de um script que você fez? Certamente sim... Já se perguntou o motivo disso?

Primeiro, o que significa esse ponto antes da barra?
Significa diretório corrente.

Diretório corrente, mas o que é isso???

Diretório corrente é aquele em que você está dentro dele no momento...

Exemplo:

Entre no diretório temporário:
$ cd /tmp

Verifique agora onde você está:
$ pwd
/tmp

Logo, seu diretório corrente é **/tmp**

Vamos supor que você criou um script chamado **script.sh** em **/tmp** e que seu diretório corrente também é **/tmp**, então para executar o script você precisaria digitar:

$ ./script.sh

Para que você não precise mais usar o **./** basta incluir o diretório corrente que é representado por ponto (**.**) na variável **PATH**. Há duas formas para resolver isso: uma é temporária e outra é permanente.

Na forma temporária, basta incluir o diretório corrente na variável PATH e exportar com o comando **export**, para que essa variável com novo valor seja reconhecida pelo sistema. O comando export torna global o valor da variável.

 # PATH=$PATH:.
 # export PATH

A variável PATH recebeu seu conteúdo novamente só que agora recebeu também o diretório corrente (.). Depois foi só exportá-la.

Para que essa alteração fique permanente para todos os usuários no sistema faça:

 # vi /etc/profile

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "`id -u`" -eq 0 ]; then
  PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
else
  PATH="/usr/local/bin:/usr/bin:/bin:/usr/games:." # o Diretório corrente foi acrescentado

fi

if [ "$PS1" ]; then
  if [ "$BASH" ]; then
    PS1='\u@\h:\w\$ '
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

export PATH

umask 022
```

Salve o arquivo e logue-se novamente.

Outras variáveis de ambiente muito comuns:

$TERM

Define o terminal padrão.

$ echo $TERM
xterm

$HOME

Indica o diretório pessoal do usuário em questão. Exemplo:
$ echo $HOME
/home/leo

Essa variável é muito usada em scripts que necessitam saber o qual diretório pessoal do usuário, ou seja, ao invés de indicar diretamente o diretório pessoal do usuário, a própria variável retorna o valor automaticamente. E esse script pode ser usado por qualquer usuário que tenha permissão de executá-lo.

$USER

Guarda o nome do usuário no momento.

$ echo $USER
leo

$SHELL

Guarda o valor do shell padrão:

$ echo $SHELL
/bin/bash

No Linux, o shell padrão é o **bash**.

$HISTSIZE

Guarda o tamanho máximo do log de comandos executados no shell é usado pelo comando **history**.
Para ver seu valor padrão:

$ echo $HISTSIZE
500

Você pode alterar para 1000 por exemplo:
$ HISTSIZE=1000
$ export HISTSIZE
$ echo $HISTSIZE
1000

$PWD

Guarda o valor do diretório corrente, ou seja, no que você está atualmente.

$ echo $PWD
/home/leo

$ cd /tmp
$ echo $PWD
/tmp

$PS1

Guarda o valor do prompt primário. Você pode personalizá-la.
Veja meu prompt:
leo@saturnov:~$

Códigos para configurar o prompt:
\w Diretório corrente
\d Exibe data
\t Exibe hora
\s Exibe o shell corrente
\u Exibe o nome do usuário
\h Exibe o nome do host (máquina)

Para mudar seu prompt temporariamente faça:
PS1='\u@\h:\w$'

Isso vai gerar um prompt assim: **leo@saturnov:~$**

Ou então:
PS1='\h@\u:\w$ \t '

Isso vai gerar um prompt assim: **saturnov@leo:/tmp 12:29:36$**

$PS2

Guarda o valor do prompt secundário. Você pode personalizá-la. O padrão é um sinal de maior.

$ echo $PS2
>

$EDITOR

Guarda o valor do editor de textos padrão, por exemplo se você quer que o editor de texto padrão seja o VIM para editar quotas em disco ou agendar tarefas com o cron basta alterar o valor dessa variável.

Exemplo:
$ EDITOR=vim
$ export EDITOR

Para que essas configurações não se percam durante um reboot, faça:

 # vim /root/.bash_profile
Adicione no arquivo:
EDITOR=vim
export EDITOR

Salve o arquivo.

 # source /root/.bash_profile (Para o arquivo ser lido novamente)

Agora, o editor de textos padrão do root é o VIM.

Tente editar quotas ou agendar tarefas, o editor que vai ser usado é o VIM agora.

$OSTYPE

Guarda o nome do sistema operacional.

Exemplo:
$ echo $OSTYPE
linux-gnu

$TMOUT

Essa variável define o tempo máximo que o shell ficará inativo. Essa variável é de grande utilidade quando se pensa em segurança, pois se você sai e deixa o terminal de texto aberto, se tiver um valor com 30 setado nela, após 30 segundos de inatividade o shell se fecha.

TMOUT=15
$ export TMOUT

Agora aguarde 15 segundos sem fazer nada no shell...

Para que você possa ver todas as variáveis já definidas, digite:
$ set | more

Para limpar o valor de uma variável:
$ unset VARIAVEL

Exemplo:

Atribuindo um valor para uma variavel:

$ BRASIL=1000
$ echo $BRASIL
1000

O valor da variável é 1000, vamos limpá-la:
$ unset BRASIL
$ echo $BRASIL

Veja que não vai aparecer nada agora.

Um comando interessante também é o **printenv** (o comando **env** também faz a mesma coisa)

Esse comando exibe uma lista de variáveis de ambiente.

Exemplo:

$ printenv
USER=leo
PWD=/home/leo
EDITOR=vim
LANG=pt_BR.UTF-8
HOME=/home/leo
LOGNAME=leo
DISPLAY=:0.0
TERM=xterm
SHELL=/bin/bash