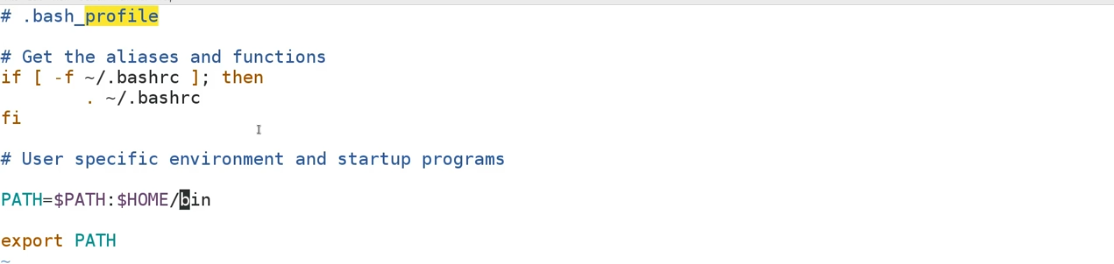
## Bash Files - .bash\_profile

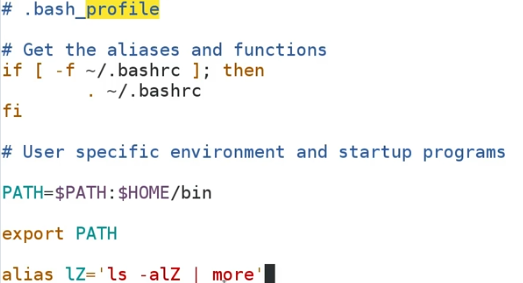
.bash\_profile

Bash configuracion profile

ls -al

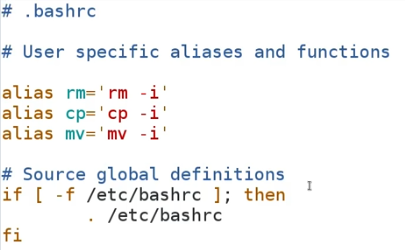
vim .bash\_profile

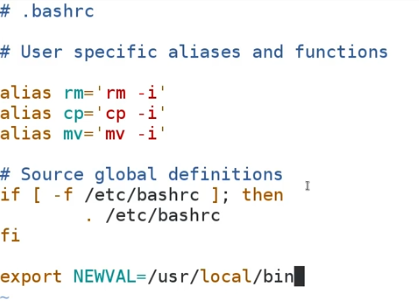




Use profile and use alias

## Bash Files - .bashrc



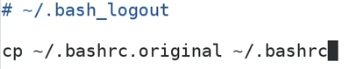


**What Makes a File a Shell Script?**

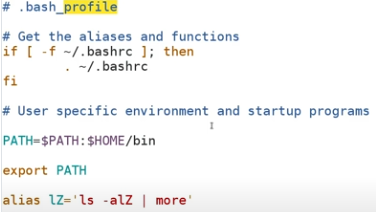
env | grep NEWVAL

bash

env | grep NEWVAL



pwd



vim test.sh



chmod u+x test.sh



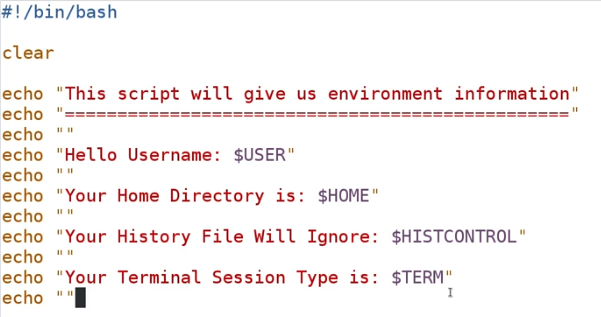
cp test.sh mytest

ls -la /usr/bin/\*.sh

which bash

## Displaying Environment Variables in a Script

env



## Using Variables on the Command Line

FIRSTNAME=”Lopez”

env | grep FIRSTNAME

export FIRSTNAME

env | grep FIRSTNAME

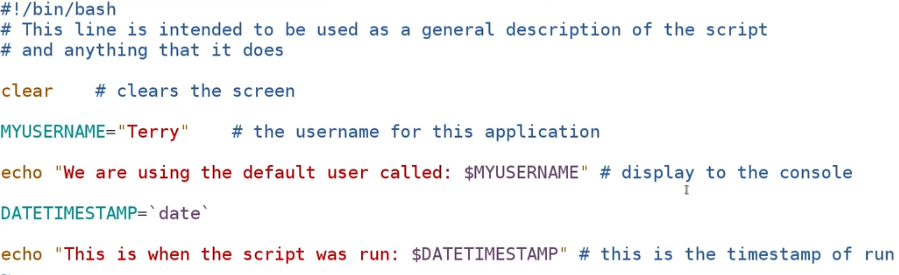
echo “Your first name is: $FIRSTNAME”

export TODAYSDATE=`date`

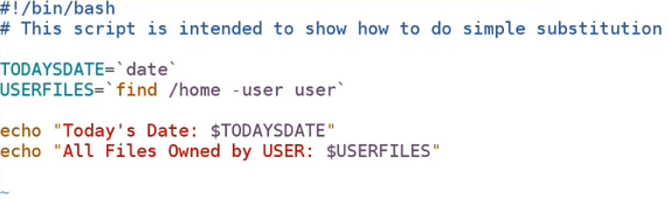
## Setting and Using Variables in Scripts

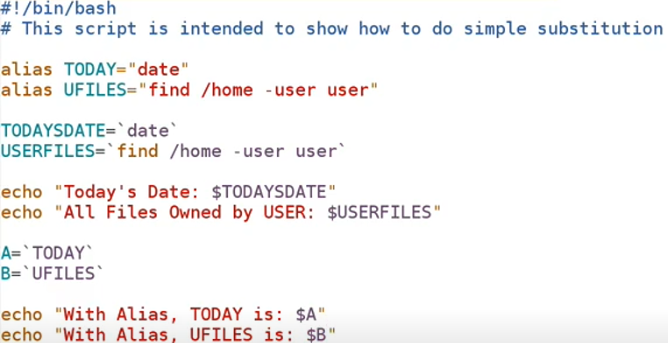
env | grep $PATH  

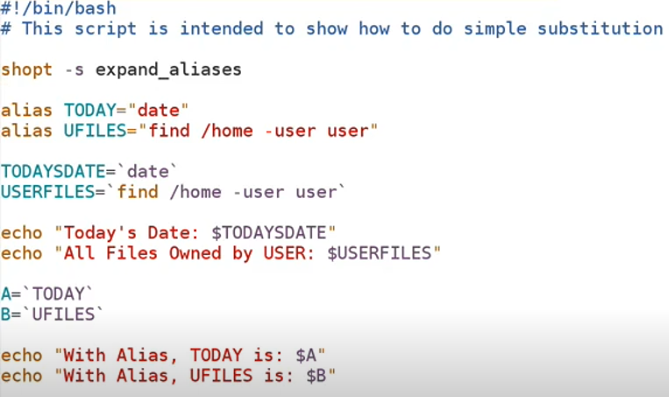

## Comment Types and Structure



## Command Substitution

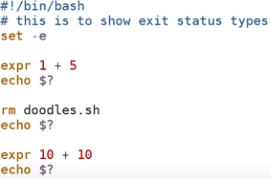






## Exit Status

echo $?



## Arithmetic Operations

expr 2 + 2

expr 2+2

expr 2+2 – 1

expr 2 + 2 – 1

expr 10 \ 2

expr 10 \* 8

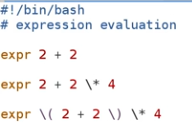
expr 10 \\* 8

echo `expr 5 – 2`

expr 15 % 4

expr ( 2 + 4 ) \\* 4

expr \( 2 + 2 \) \\* 4



## Global and Local Environment Variables

printenv

env

set

## Special Characters - Quotes and Escapes

echo $

echo $COL

echo “\$COL”

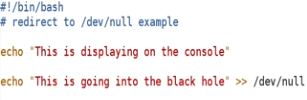
echo '$ONE $TWO $THREE'

echo “\$ONE $TWO THREE”

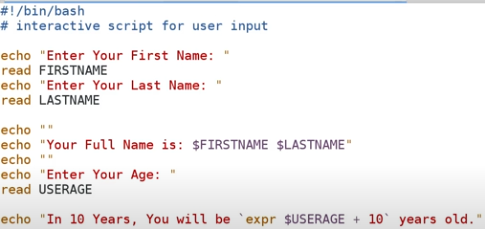
echo “`date`”

echo “Today´s date is: `date`”

## Using /dev/null



## The Read Statement

****

## Shell Expansion

## echo st{ot, ort, oot}

## export NEWPATH=PATH:~

## env | grep NEWPATH

## echo ~+

## echo “${!HO\*}”

## export CARNAME=else && $CARNAME

## echo “${VARNAME:=valuesomething}”

## echo $VARNAME

## echo “$[ 2 \* 2 ]”

## Types of Variables (Implicit vs. Explicit Definition)

## MYVAR=4

## echo $MYVAR

## echo `expr $MYVAR + 5 `

## declare -p MYVAR

## MYVAR1=”Name”

## echo $MYVAR1

## echo “`expr $MYVAR + 5`”

## declare -i NEWVAR=10

## declare -p NEWVAR

## NEWVAR=”Value New”

## echo $MYVAR

## NEWVAR=”Somenthing”

## echo $NEWVAR

## declare -p NEWVAR

## NEWVAR=111

## declare +i NEWVAR

## declare -p NEWVAR

## declare -r READONLY=”This is a string we cannot overwrite”

## declart -p READONLY

## READONLY=”New Value”

## declare +r READONLY

## readonly MYREADONLY=”This String”

## declare -p MYREADONLY

## declare -p MYREADONLY

## MYREADONLY=”Something new”

## declare +r MYREADONY

## Arrays

## MYARRAY=(“FIRST” “SECOND” “THIRD”)

## echo ${MYARRAY[0]}

## MYARRAY[3]=”Fourth”

## echo ${MYARRAY[\*]}

## MYARRAY2=(“FIRST”,“SECOND”,“THIRD”)

## echo $MYARRAY2

## echo ${MYARRAY2[0]}

## echo ${MYARRAY2[1]}

## echo ${MYARRAY2[1]}

## 

## Passing Variables to Scripts at the Command Line

## 

## Commadline.sh

## Commadline.sh Hello

## Commadline.sh Hello World

## Commadline.sh “Hello World”

## Commadline.sh `expr 2 + 2`

## 

## Commadline.sh Hello World

## Commadline.sh Hello World Now

## 

## Commadline.sh UserName Passw0rd123

## The If Statement

## 

## Ifstatement.sh

## 

## textfile.sh

## touch mytext.txt

## 

## textfile.sh wedontknow.sh

## textfile.sh mytext.txt

## 

## 

## 

## iftext.sh mytext.txt

## If/Then/Else

## 

## Ifthenelse.sh

## 

## Ifthenelse.sh

## 

## Ifthenelse.sh

## 

## For Loop

## 

## forexample.sh

## Case Statement

## 

## casestatement.sh

## While Loop

## 

## whileloop.sh

## Execution Operators (&& and ||)

## 

## operators.sh

## Reading Files