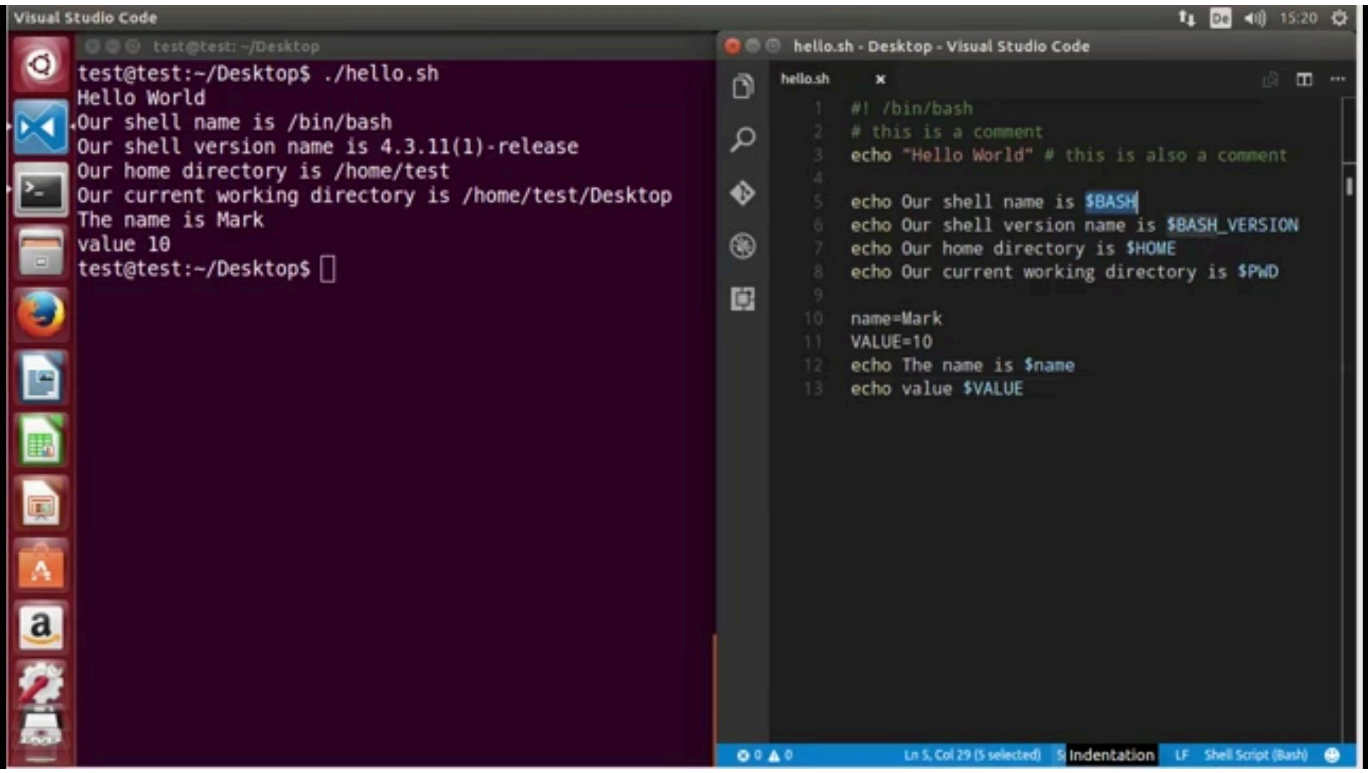
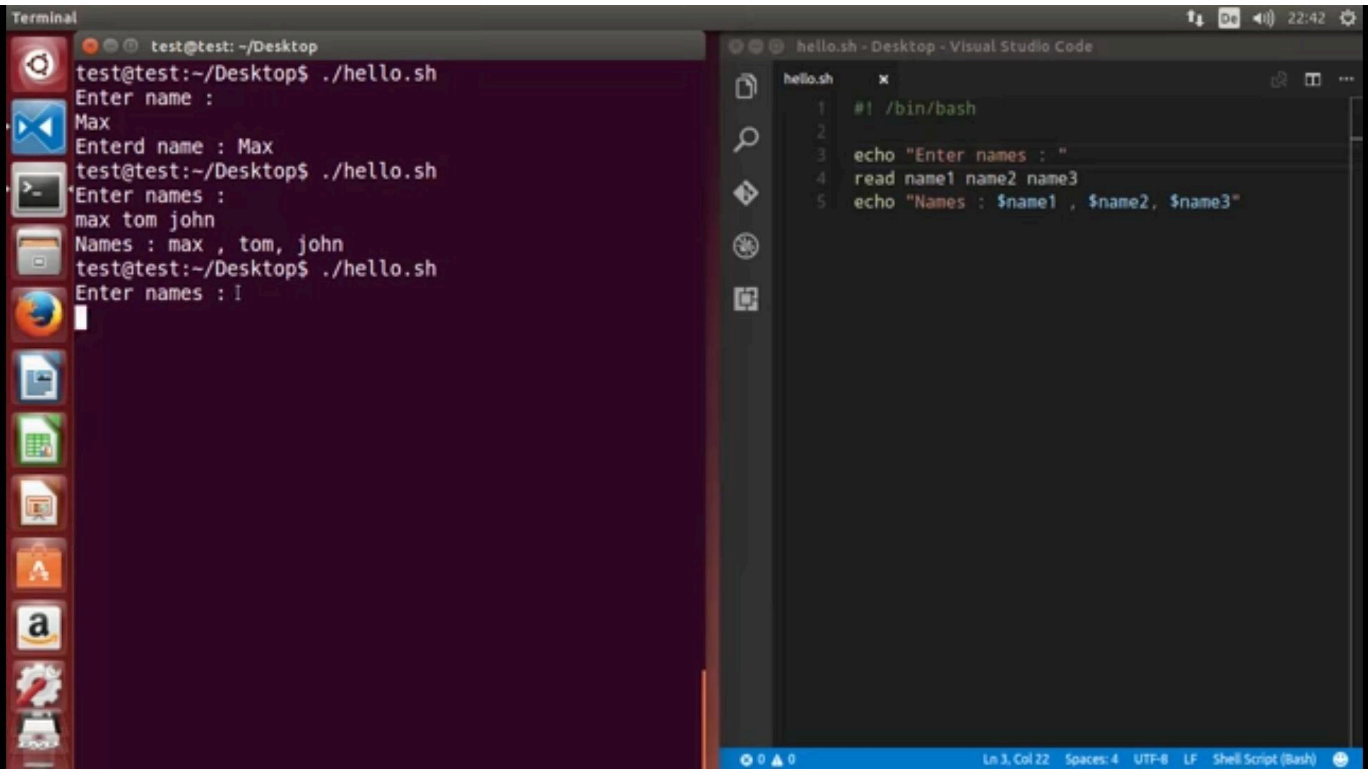
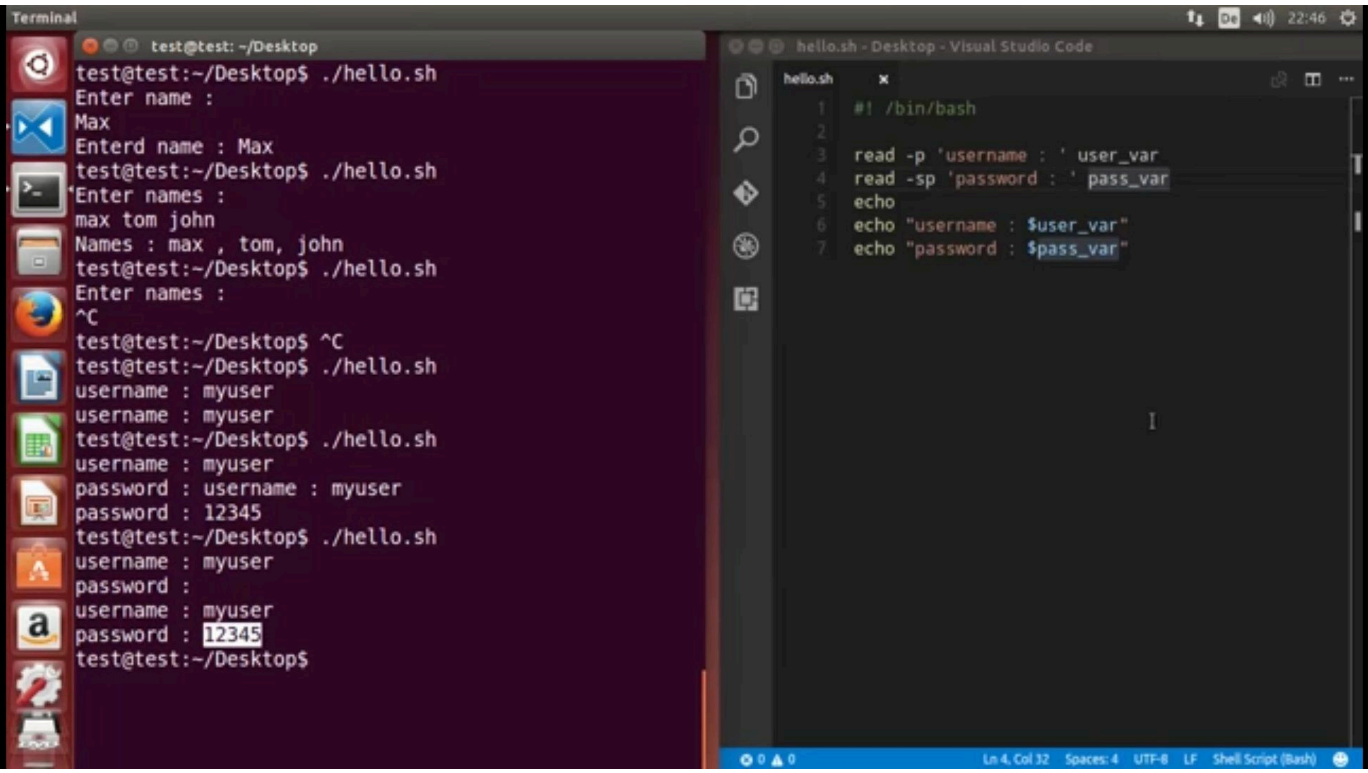
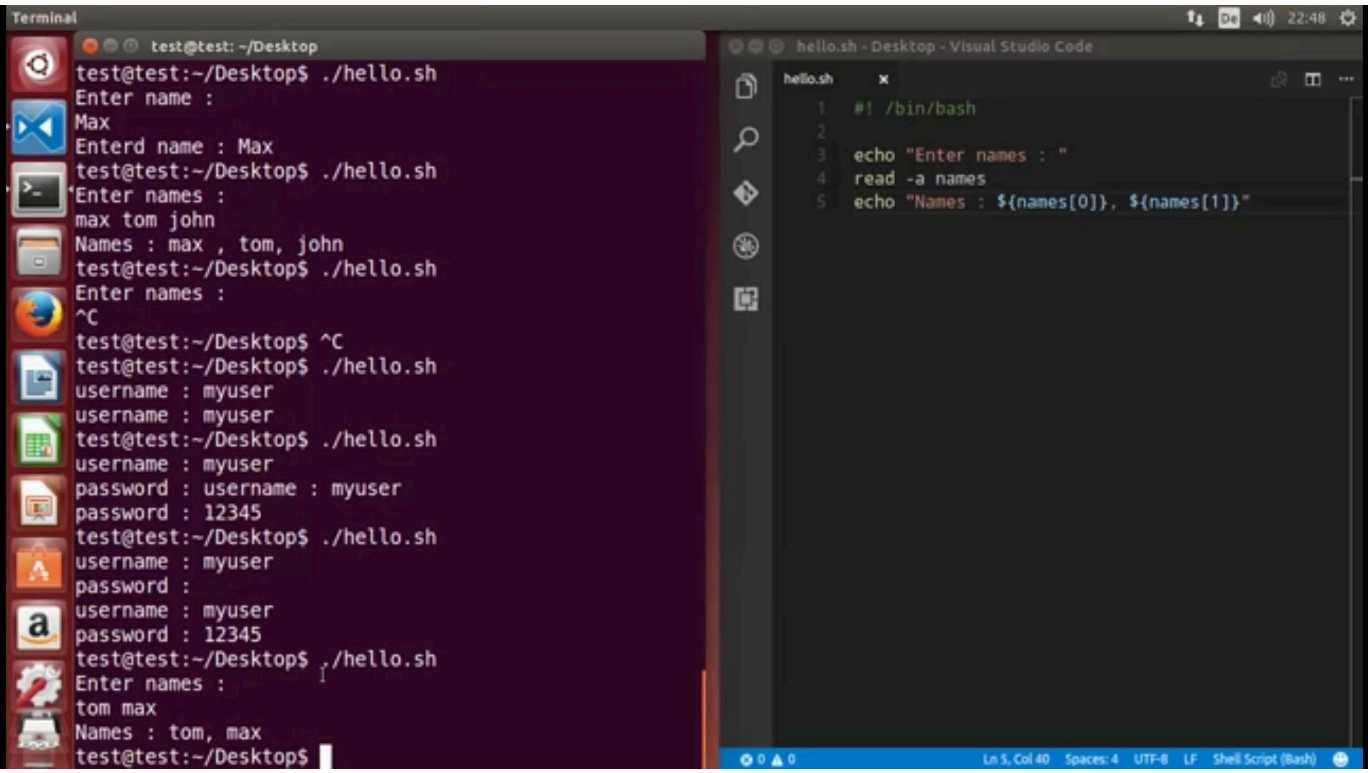
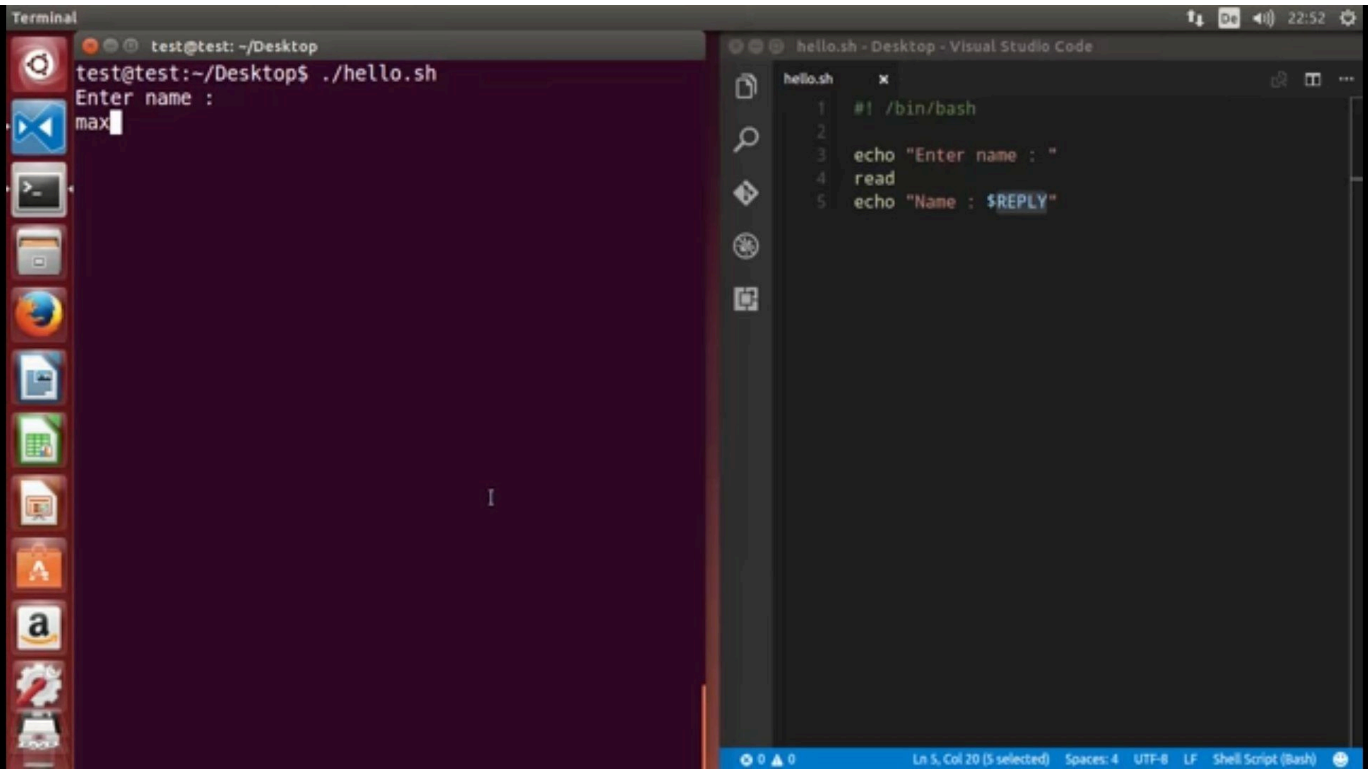
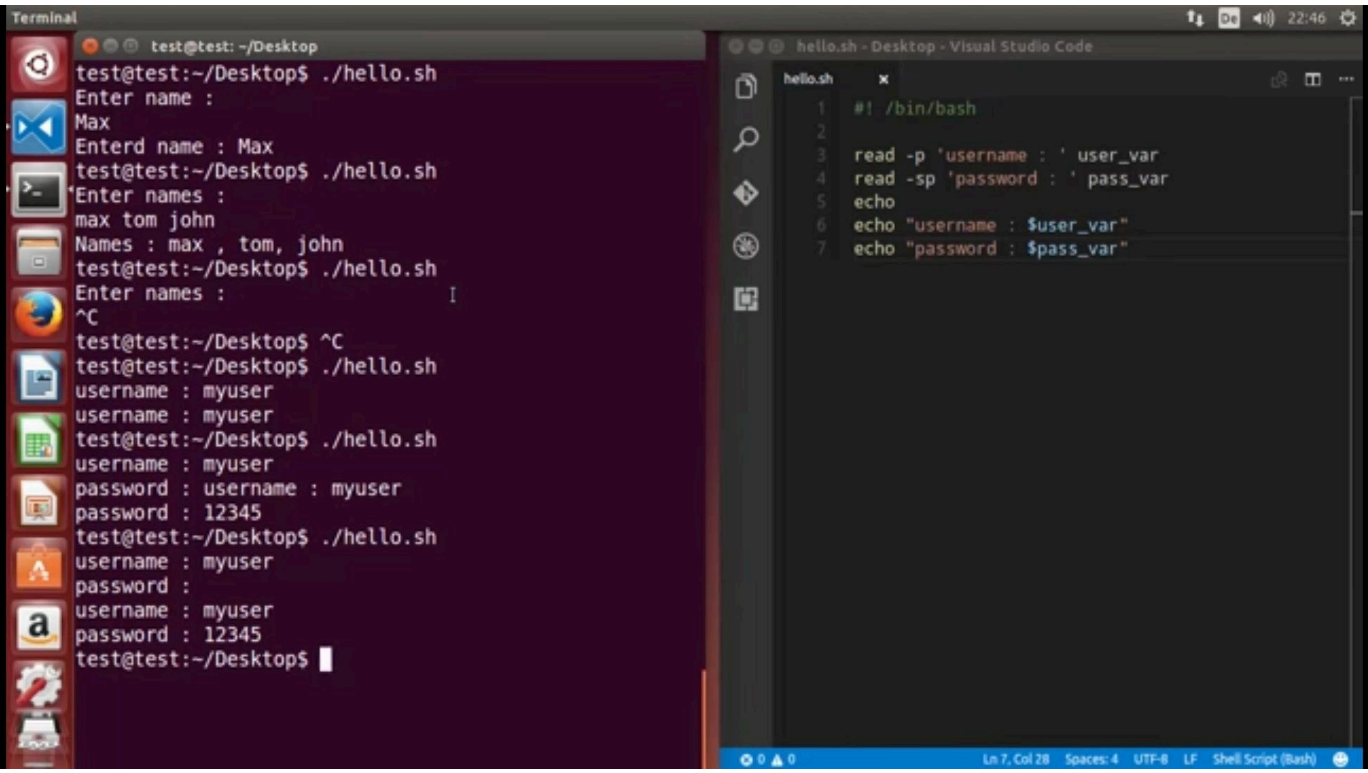

Visual Studio Code
15:20
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Hello World
Our shell name is /bin/bash
Our shell version name is 4.3.11(1)-release
Our home directory is /home/test
Our current working directory is /home/test/Desktop
The name is Mark
value 10
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
this is a comment
echo "Hello World" # this is also a comment
echo Our shell name is $BASH
echo Our shell version name is $BASH_VERSION
echo Our home directory is $HOME
echo Our current working directory is $PWD
name=Mark
VALUE=10
echo The name is $name
echo value $VALUE
Ln 5, Col 29 (5 selected)
Indentation
LF
Shell Script (Bash)

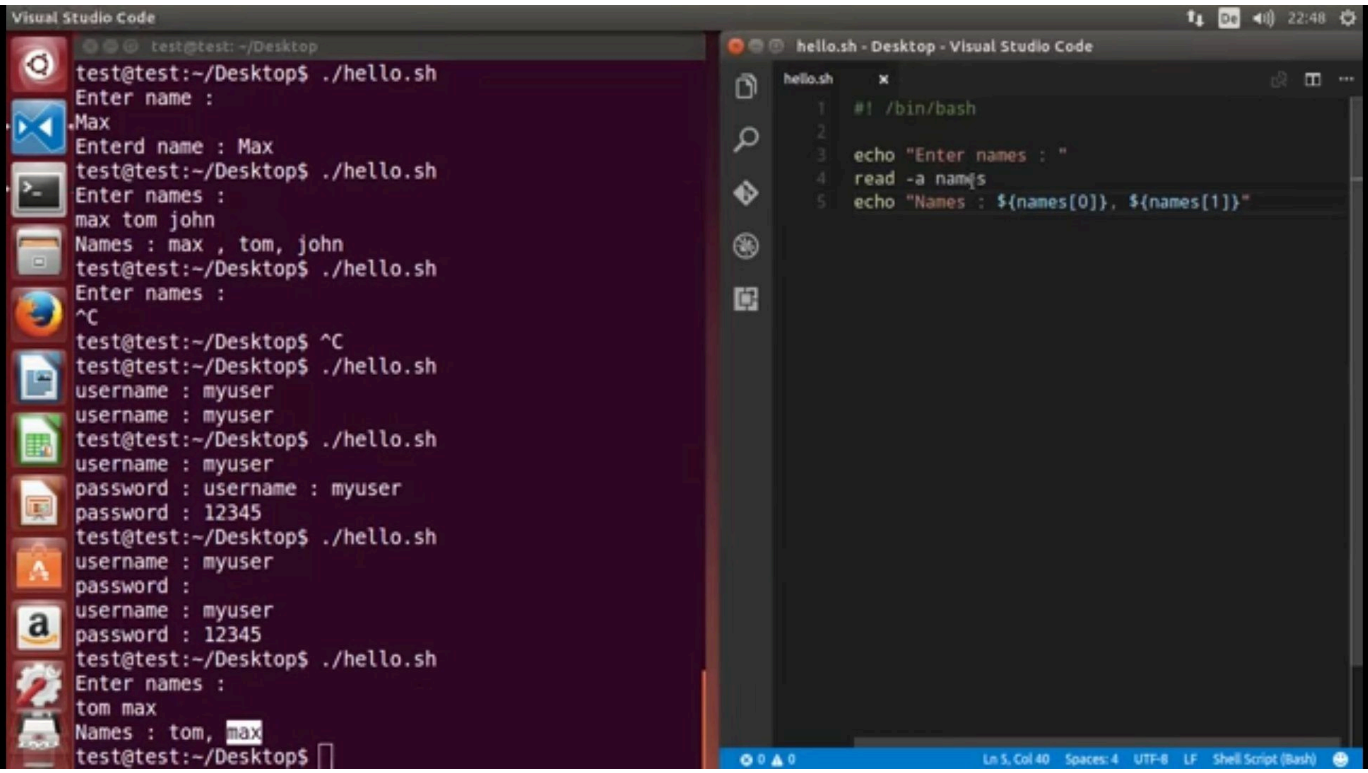

Terminal
22:42
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter name :
Max
Enterd name : Max
test@test:~/Desktop$ ./hello.sh
Enter names :
max tom john
Names : max , tom, john
test@test:~/Desktop$ ./hello.sh
Enter names :
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
echo "Enter names : "
read name1 name2 name3
echo "Names : $name1 , $name2, $name3"
Ln 3, Col 22
Spaces: 4
UTF-8
LF
Shell Script (Bash)

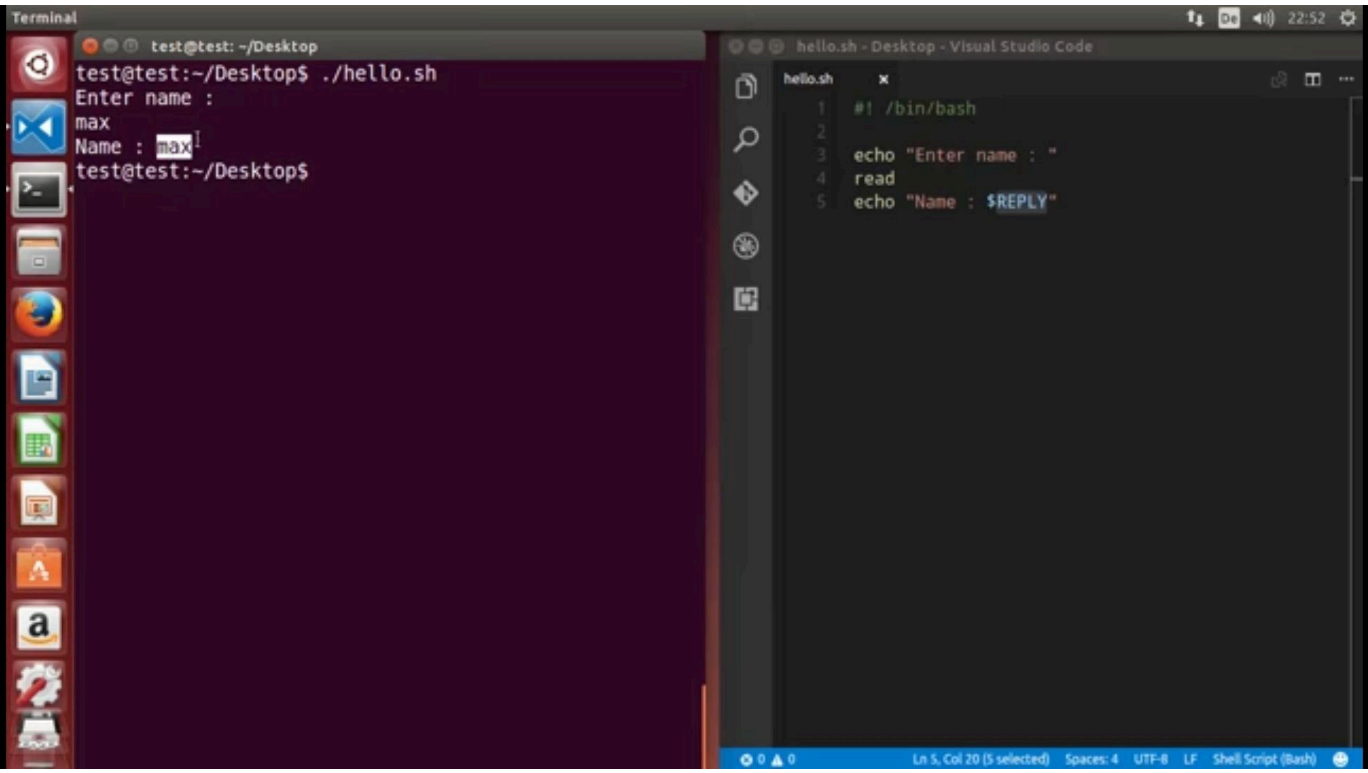

Terminal
22:46
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter name :
Max
Enterd name : Max
test@test:~/Desktop$ ./hello.sh
Enter names :
max tom john
Names : max , tom, john
test@test:~/Desktop$ ./hello.sh
Enter names :
^C
test@test:~/Desktop$ ^C
test@test:~/Desktop$ ./hello.sh
username : myuser
username : myuser
test@test:~/Desktop$ ./hello.sh
username : myuser
password : username : myuser
password : 12345
test@test:~/Desktop$ ./hello.sh
username : myuser
password :
username : myuser
password : 12345
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
read -p 'username : ' user_var
read -sp 'password : ' pass_var
echo
echo "username : $user_var"
echo "password : $pass_var"
Ln 4, Col 32
Spaces: 4
UTF-8
LF
Shell Script (Bash)

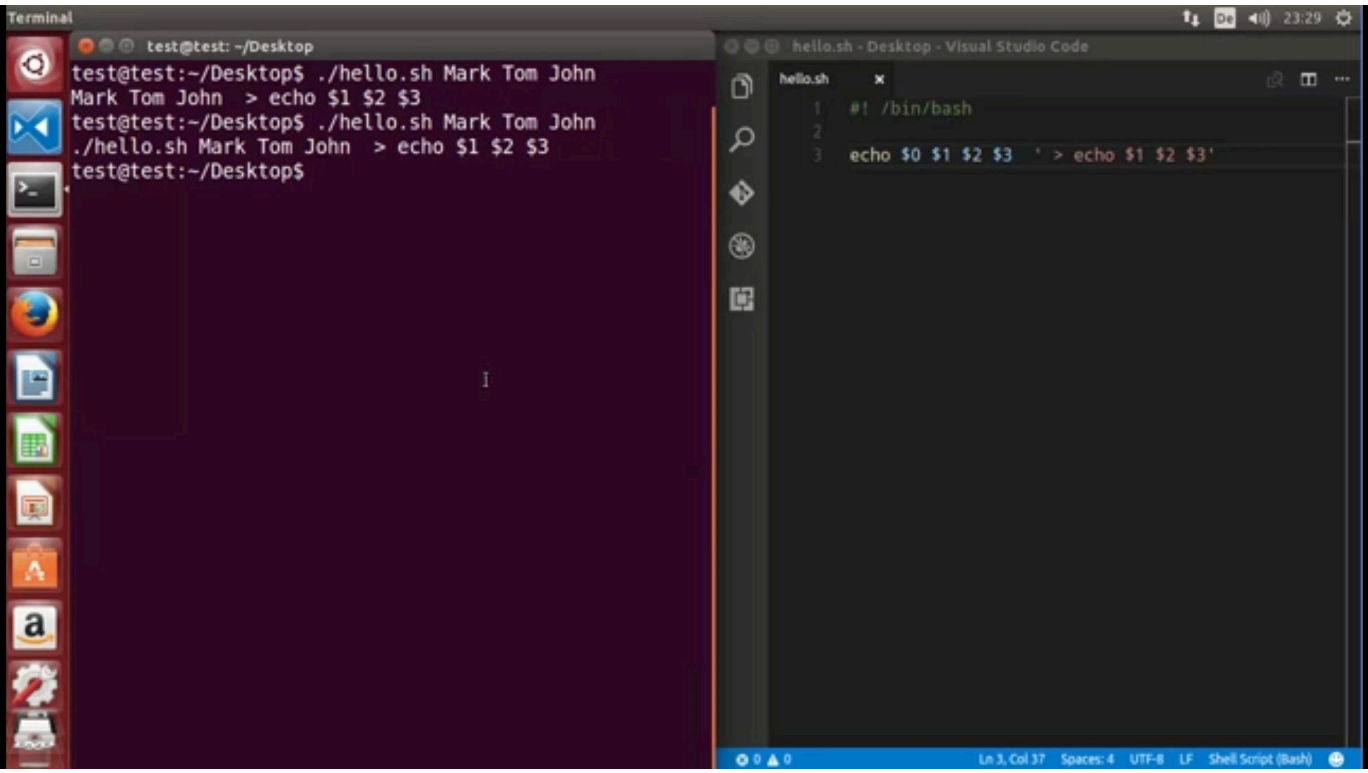

Terminal
22:48
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter name :
Max
Enterd name : Max
test@test:~/Desktop$ ./hello.sh
Enter names :
max tom john
Names : max , tom, john
test@test:~/Desktop$ ./hello.sh
Enter names :
^C
test@test:~/Desktop$ ^C
test@test:~/Desktop$ ./hello.sh
username : myuser
username : myuser
test@test:~/Desktop$ ./hello.sh
username : myuser
password : username : myuser
password : 12345
test@test:~/Desktop$ ./hello.sh
username : myuser
password :
username : myuser
password : 12345
test@test:~/Desktop$ ./hello.sh
Enter names :
tom max
Names : tom, max
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
echo "Enter names : "
read -a names
echo "Names : ${names[0]}, ${names[1]}"
Ln 5, Col 40
Spaces: 4
UTF-8
LF
Shell Script (Bash)

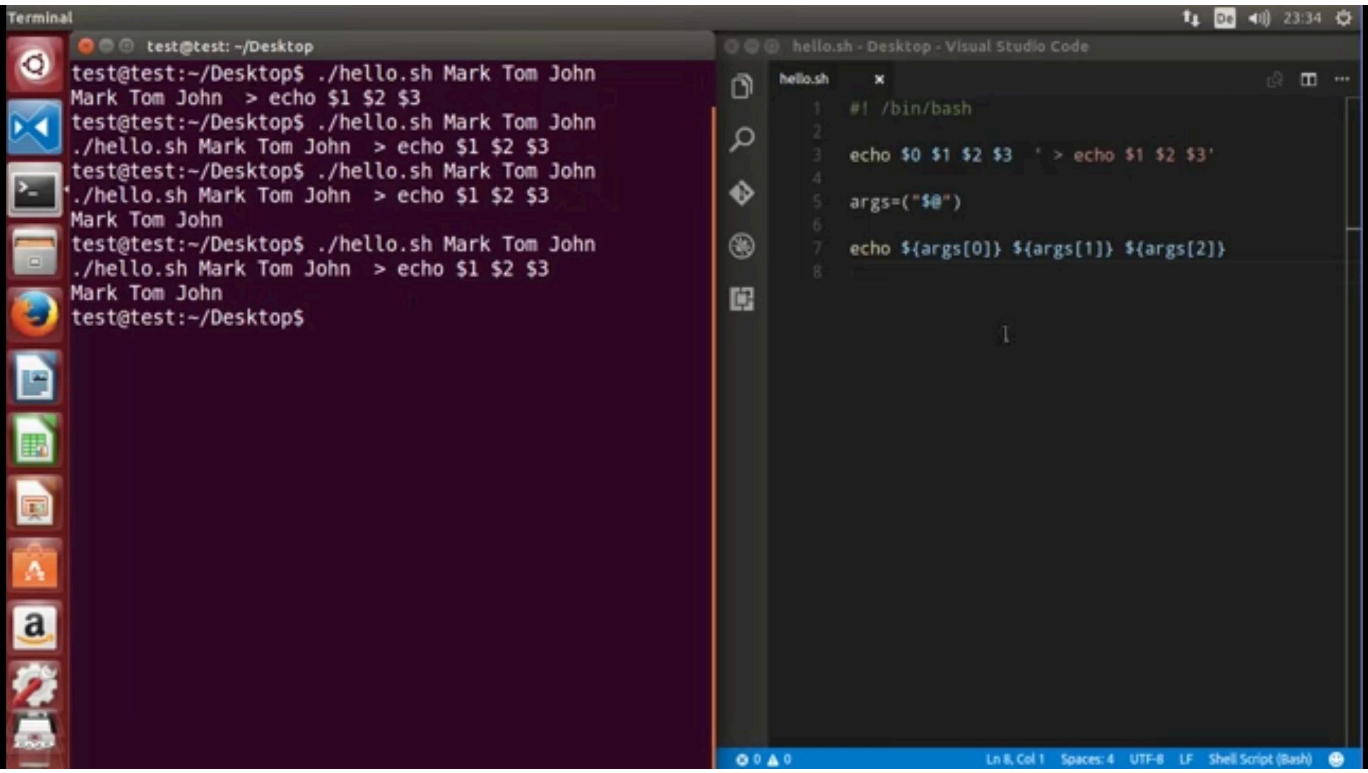

Terminal
22:52
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter name :
max
hello.sh - Desktop - Visual Studio Code
hello.sh
1 #! /bin/bash
2
3 echo "Enter name : "
4 read
5 echo "Name : $REPLY"
0 0
Ln 5, Col 20 (5 selected)
Spaces: 4
UTF-8
LF
Shell Script (Bash)

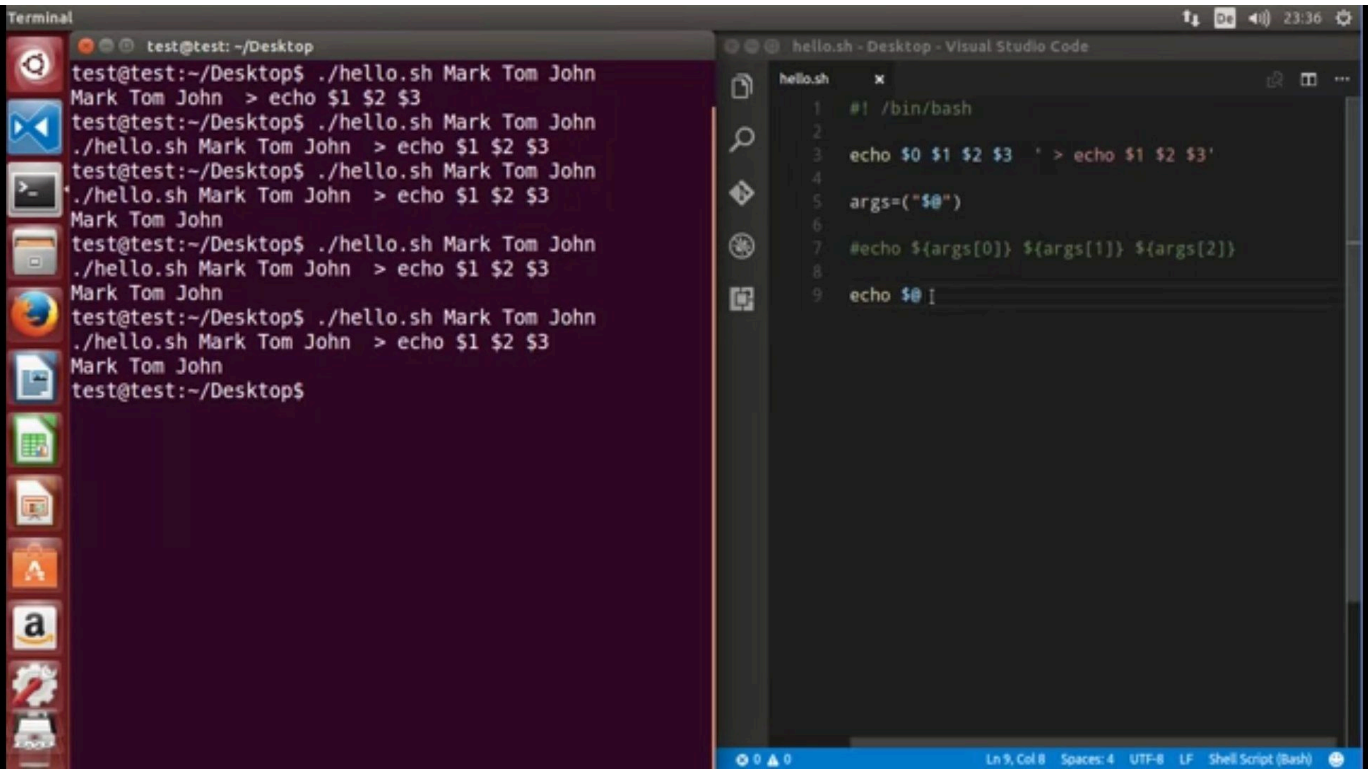

Terminal
22:46
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter name :
Max
Enterd name : Max
test@test:~/Desktop$ ./hello.sh
Enter names :
max tom john
Names : max , tom, john
test@test:~/Desktop$ ./hello.sh
Enter names :
^C
test@test:~/Desktop$ ^C
test@test:~/Desktop$ ./hello.sh
username : myuser
username : myuser
test@test:~/Desktop$ ./hello.sh
username : myuser
password : username : myuser
password : 12345
test@test:~/Desktop$ ./hello.sh
username : myuser
password :
username : myuser
password : 12345
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
read -p 'username : ' user_var
read -sp 'password : ' pass_var
echo
echo "username : $user_var"
echo "password : $pass_var"
Ln 7, Col 28
Spaces: 4
UTF-8
LF
Shell Script (Bash)

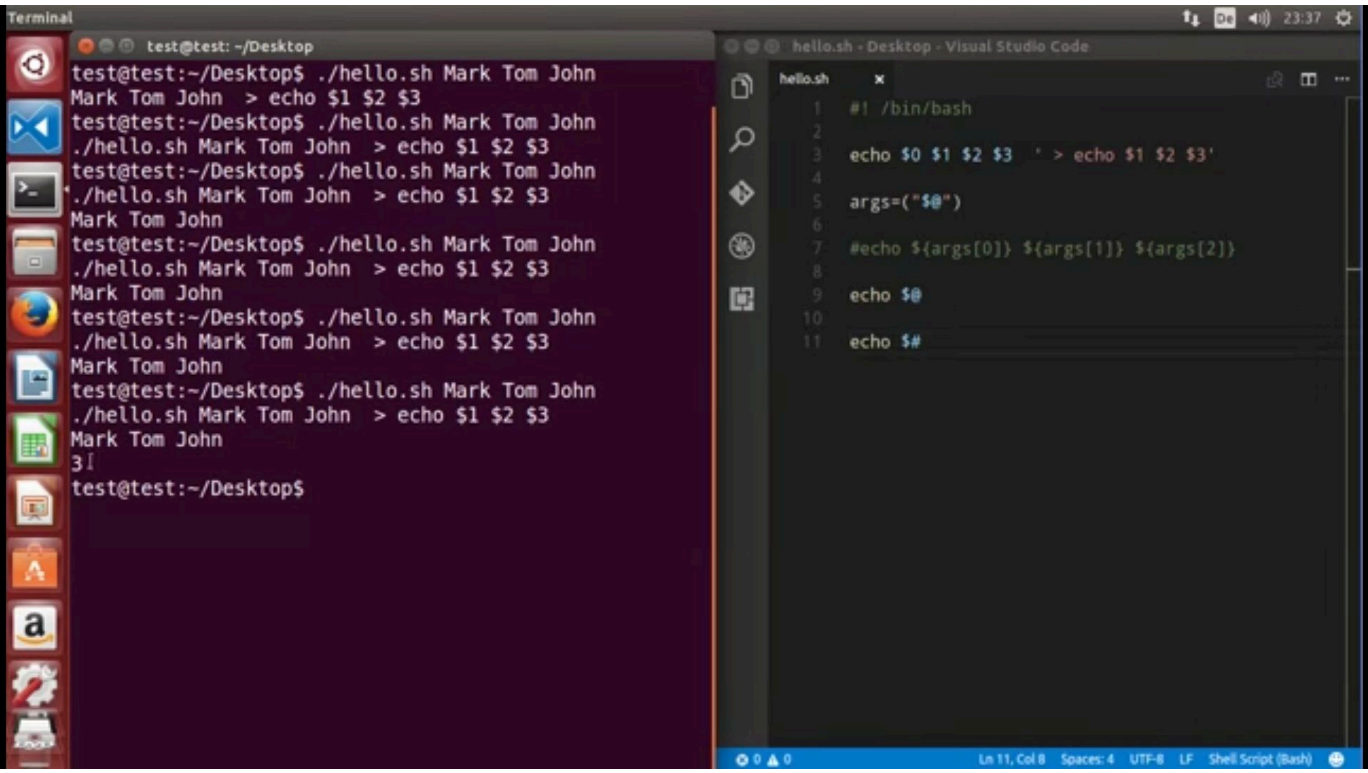

Visual Studio Code
22:48
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter name :
Max
Enterd name : Max
test@test:~/Desktop$ ./hello.sh
Enter names :
max tom john
Names : max , tom, john
test@test:~/Desktop$ ./hello.sh
Enter names :
^C
test@test:~/Desktop$ ^C
test@test:~/Desktop$ ./hello.sh
username : myuser
username : myuser
test@test:~/Desktop$ ./hello.sh
username : myuser
password : username : myuser
password : 12345
test@test:~/Desktop$ ./hello.sh
username : myuser
password :
username : myuser
password : 12345
test@test:~/Desktop$ ./hello.sh
Enter names :
tom max
Names : tom, max
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
echo "Enter names : "
read -a names
echo "Names : ${names[0]}, ${names[1]}"
Ln 5, Col 40
Spaces: 4
UTF-8
LF
Shell Script (Bash)

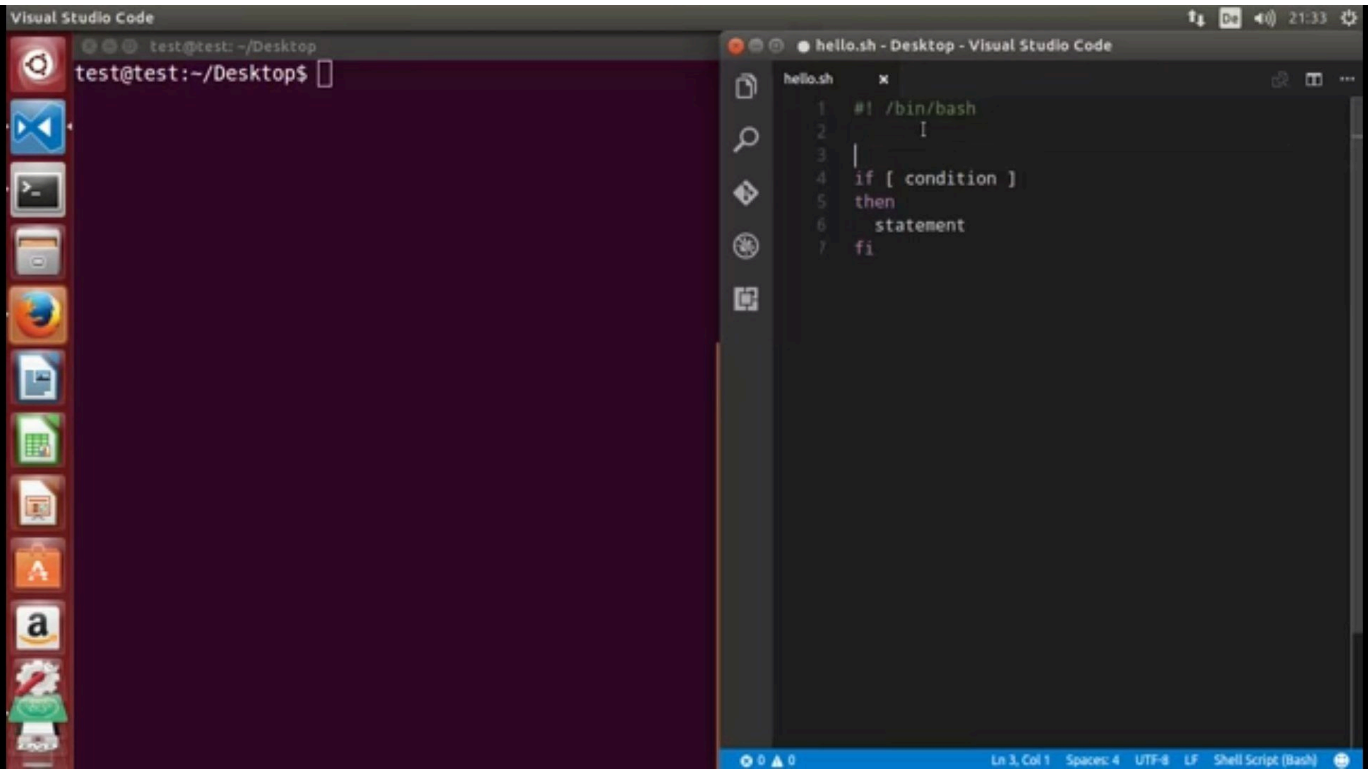

Terminal
22:52
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter name :
max
Name : max
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
1 #! /bin/bash
2
3 echo "Enter name : "
4 read
5 echo "Name : $REPLY"
Ln 5, Col 20 (5 selected)
Spaces: 4
UTF-8
LF
Shell Script (Bash)

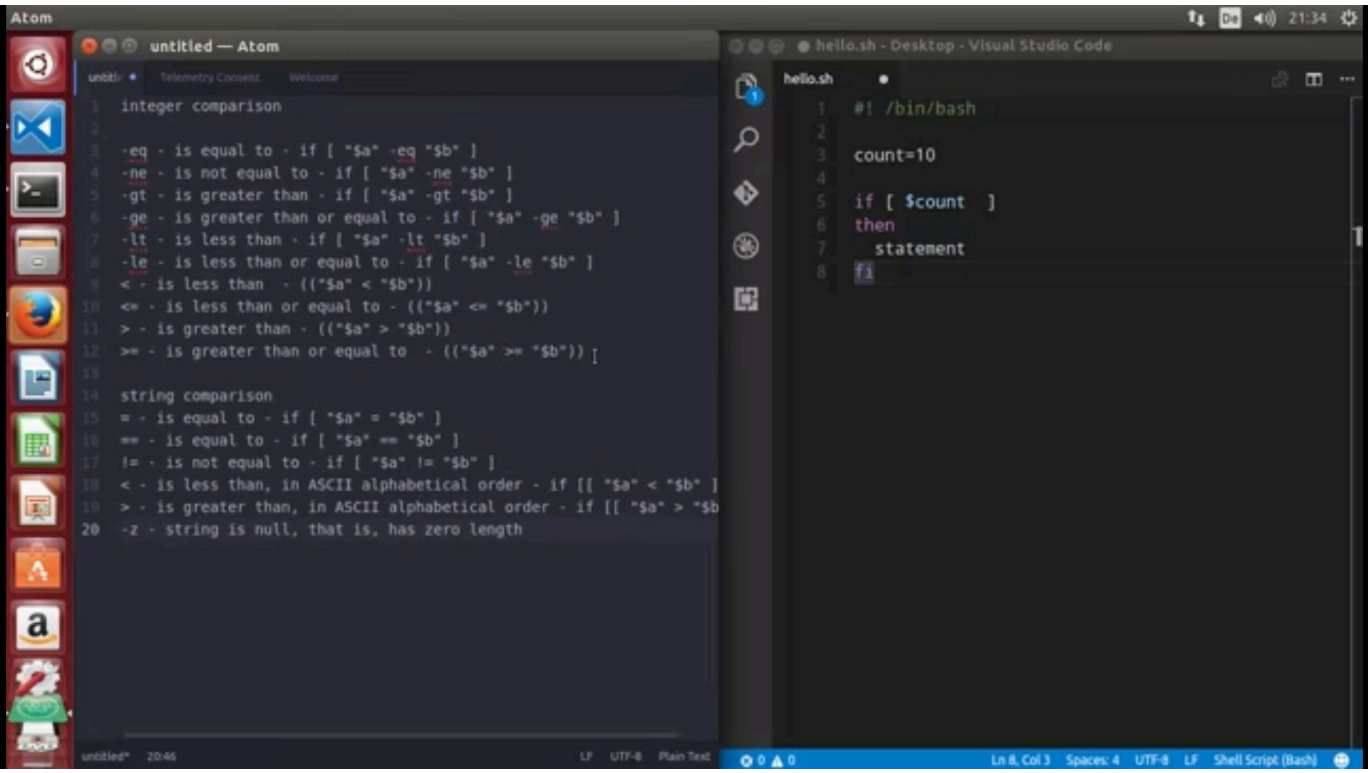

Terminal
23:29
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh Mark Tom John
Mark Tom John > echo $1 $2 $3
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
1 #! /bin/bash
2
3 echo $0 $1 $2 $3 ' > echo $1 $2 $3'
Ln 3, Col 37
Spaces: 4
UTF-8
LF
Shell Script (Bash)

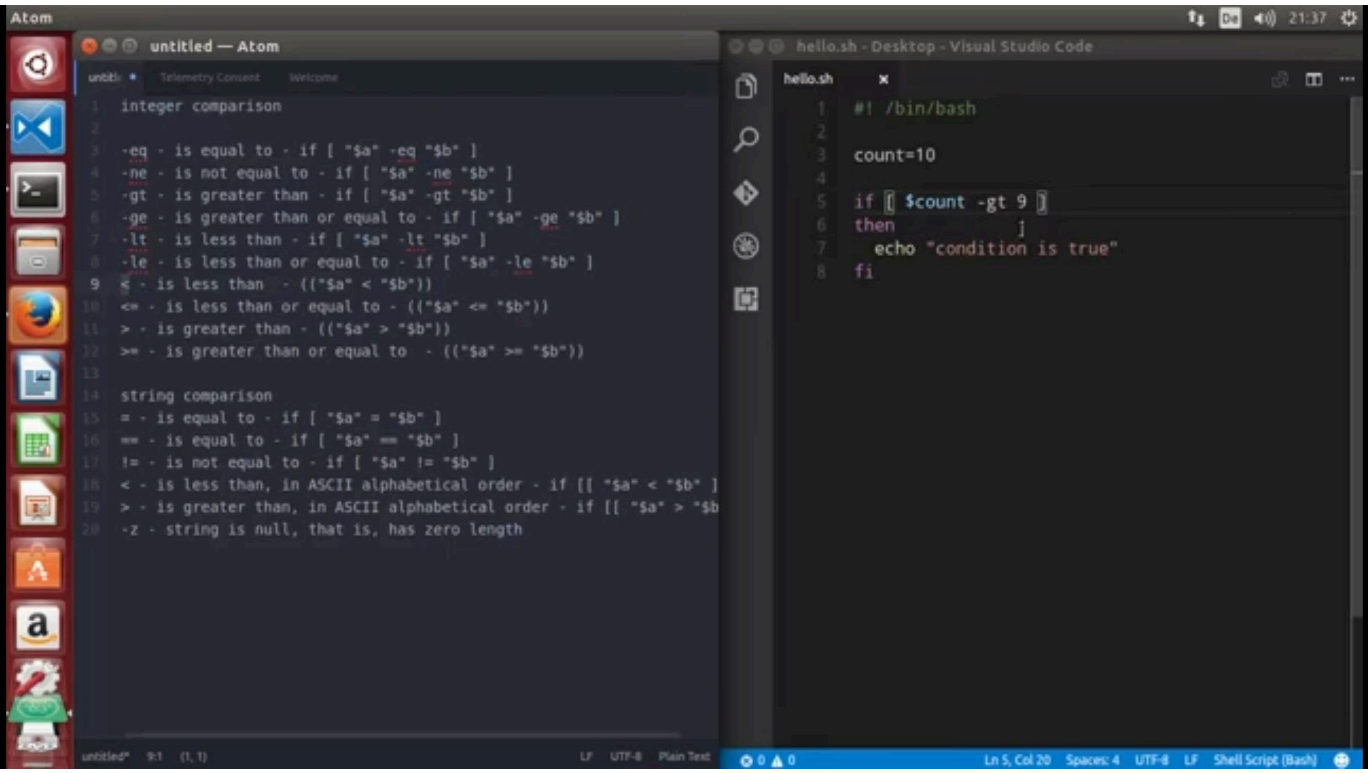

Terminal
23:34
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh Mark Tom John
Mark Tom John > echo $1 $2 $3
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
Mark Tom John
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
Mark Tom John
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
echo $0 $1 $2 $3 ' > echo $1 $2 $3'
args=("$@")
echo ${args[0]} ${args[1]} ${args[2]}
Ln 8, Col 1
Spaces: 4
UTF-8
LF
Shell Script (Bash)

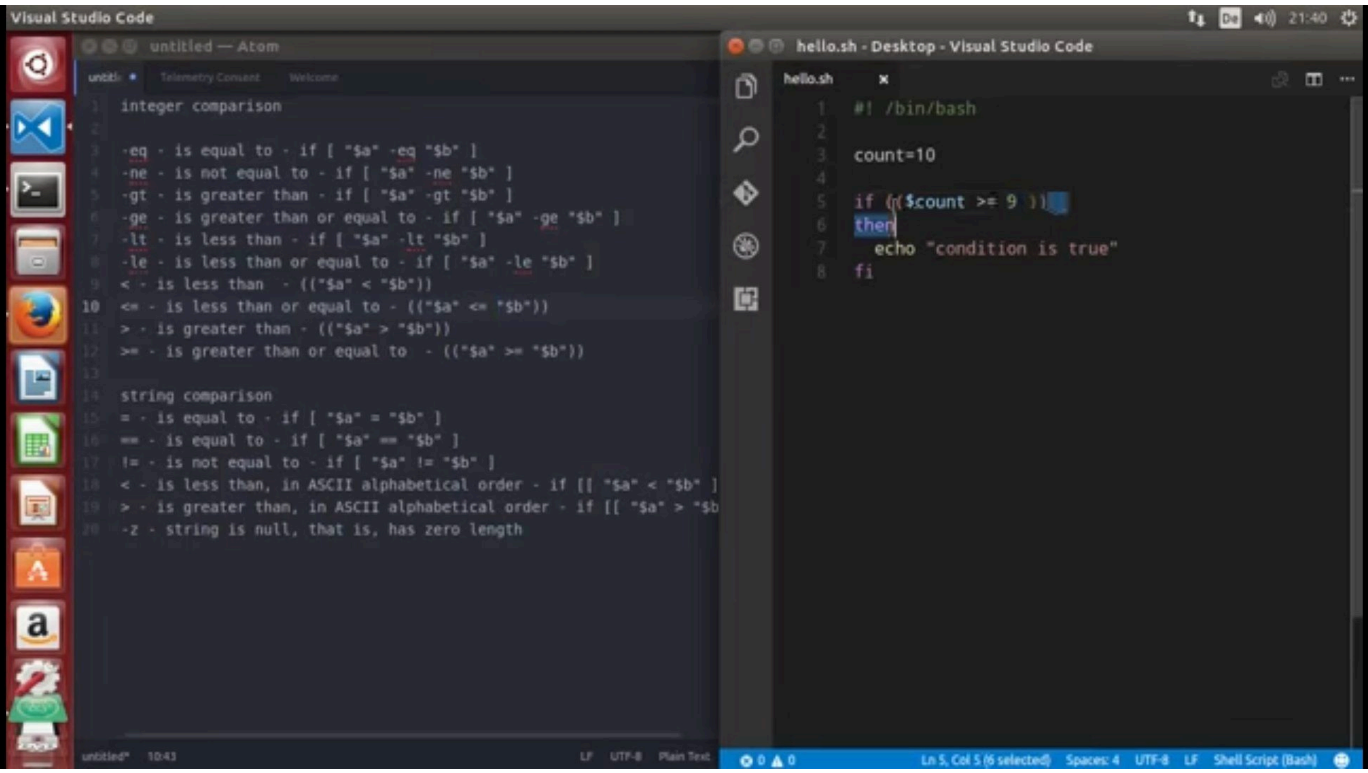

Terminal
23:36
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh Mark Tom John
Mark Tom John > echo $1 $2 $3
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
Mark Tom John
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
Mark Tom John
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
Mark Tom John
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
1 #! /bin/bash
2
3 echo $0 $1 $2 $3 ' > echo $1 $2 $3'
4
5 args=("$@")
6
7 #echo ${args[0]} ${args[1]} ${args[2]}
8
9 echo $@
Ln 9, Col 8 Spaces: 4 UTF-8 LF Shell Script (Bash)

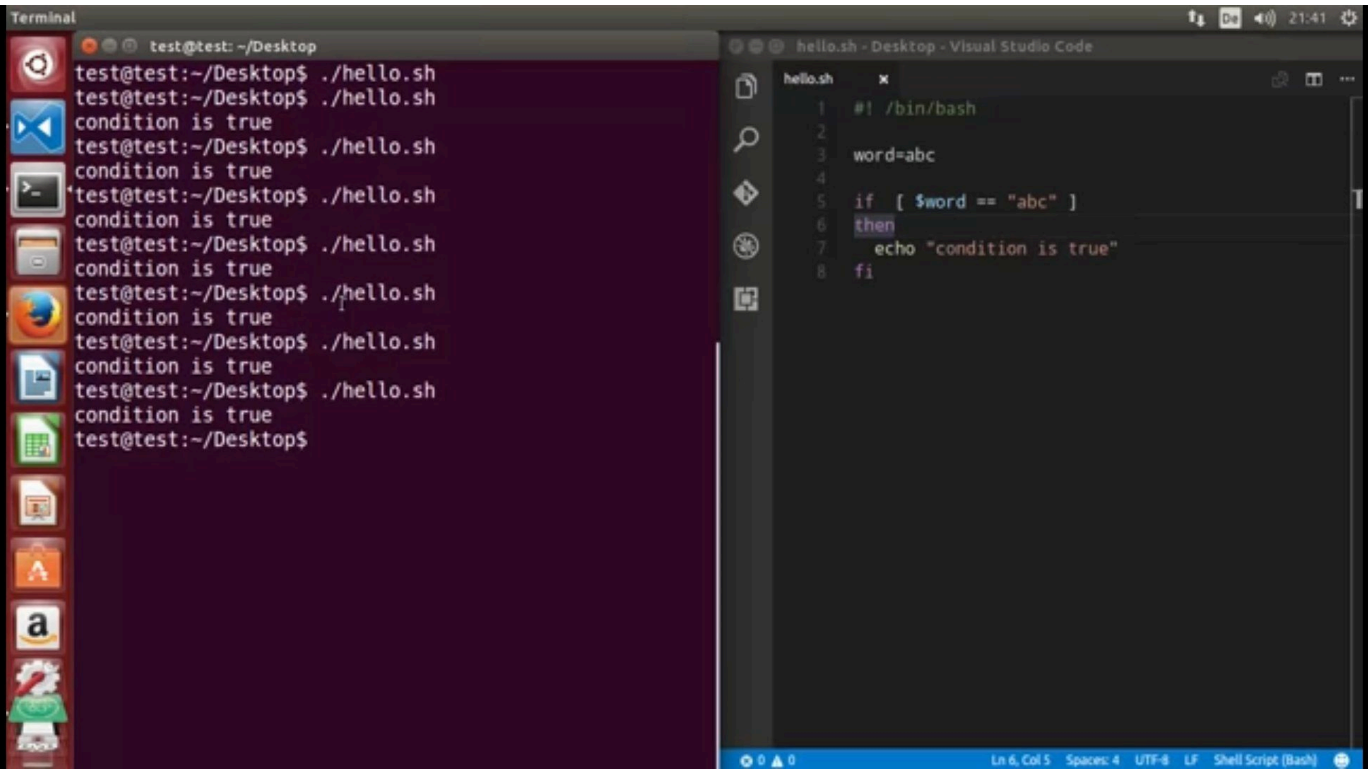

Terminal
23:37
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh Mark Tom John
Mark Tom John > echo $1 $2 $3
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
Mark Tom John
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
Mark Tom John
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
Mark Tom John
test@test:~/Desktop$ ./hello.sh Mark Tom John
./hello.sh Mark Tom John > echo $1 $2 $3
Mark Tom John
3
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
1 #! /bin/bash
2
3 echo $0 $1 $2 $3 ' > echo $1 $2 $3'
4
5 args=("$@")
6
7 #echo ${args[0]} ${args[1]} ${args[2]}
8
9 echo $@
10
11 echo $#
Ln 11, Col 8 Spaces: 4 UTF-8 LF Shell Script (Bash)

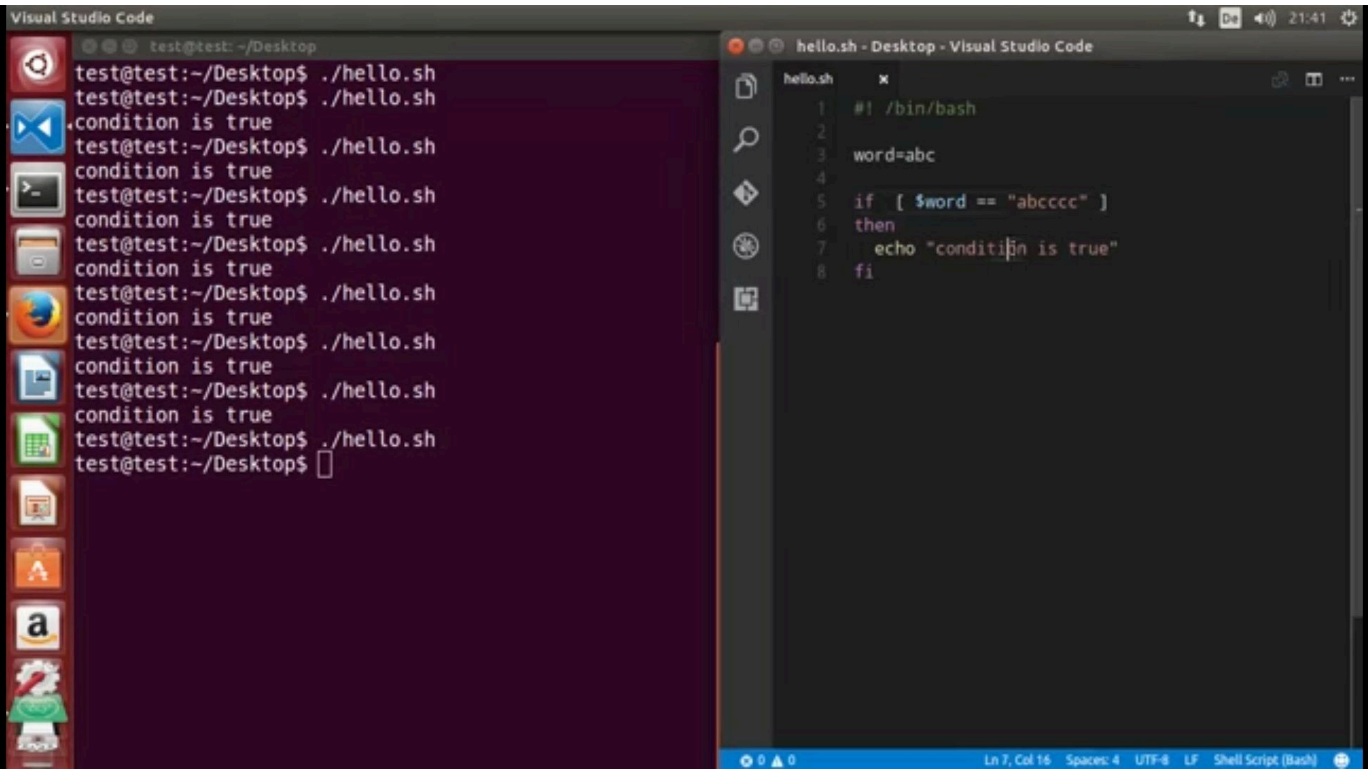

Visual Studio Code
21:33
test@test: ~/Desktop
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
if [condition]
then
statement
fi
Ln 3, Col 1
Spaces: 4
UTF-8
LF
Shell Script (Bash)

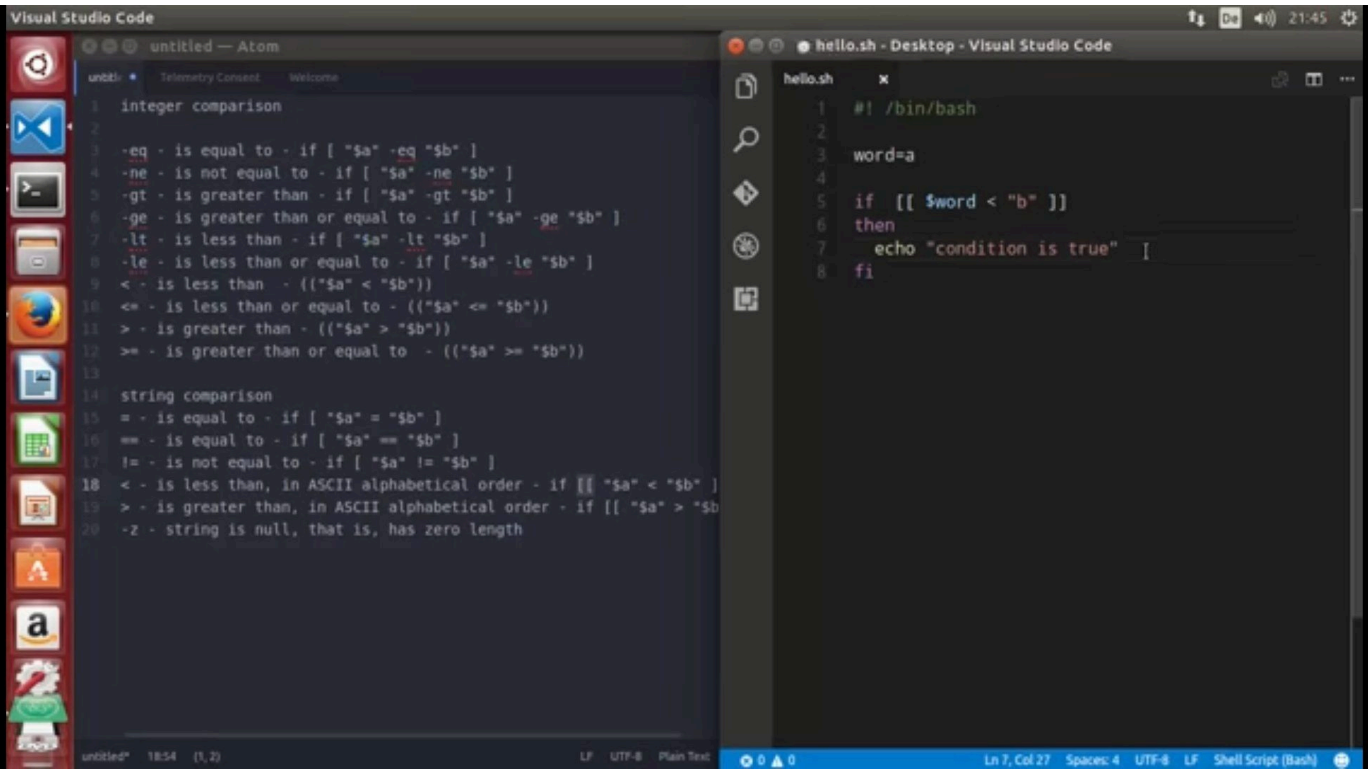

Atom
21:34
untitled — Atom
integer comparison
-eq - is equal to - if ["$a" -eq "$b"]
-ne - is not equal to - if ["$a" -ne "$b"]
-gt - is greater than - if ["$a" -gt "$b"]
-ge - is greater than or equal to - if ["$a" -ge "$b"]
-lt - is less than - if ["$a" -lt "$b"]
-le - is less than or equal to - if ["$a" -le "$b"]
< - is less than - (("$a" < "$b"))
<= - is less than or equal to - (("$a" <= "$b"))
> - is greater than - (("$a" > "$b"))
>= - is greater than or equal to - (("$a" >= "$b"))
string comparison
= - is equal to - if ["$a" = "$b"]
== - is equal to - if ["$a" == "$b"]
!= - is not equal to - if ["$a" != "$b"]
< - is less than, in ASCII alphabetical order - if [["$a" < "$b"]
> - is greater than, in ASCII alphabetical order - if [["$a" > "$b
-z - string is null, that is, has zero length
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
count=10
if [$count]
then
statement
fi
Ln 8, Col 3
Spaces: 4
UTF-8
LF
Shell Script (Bash)

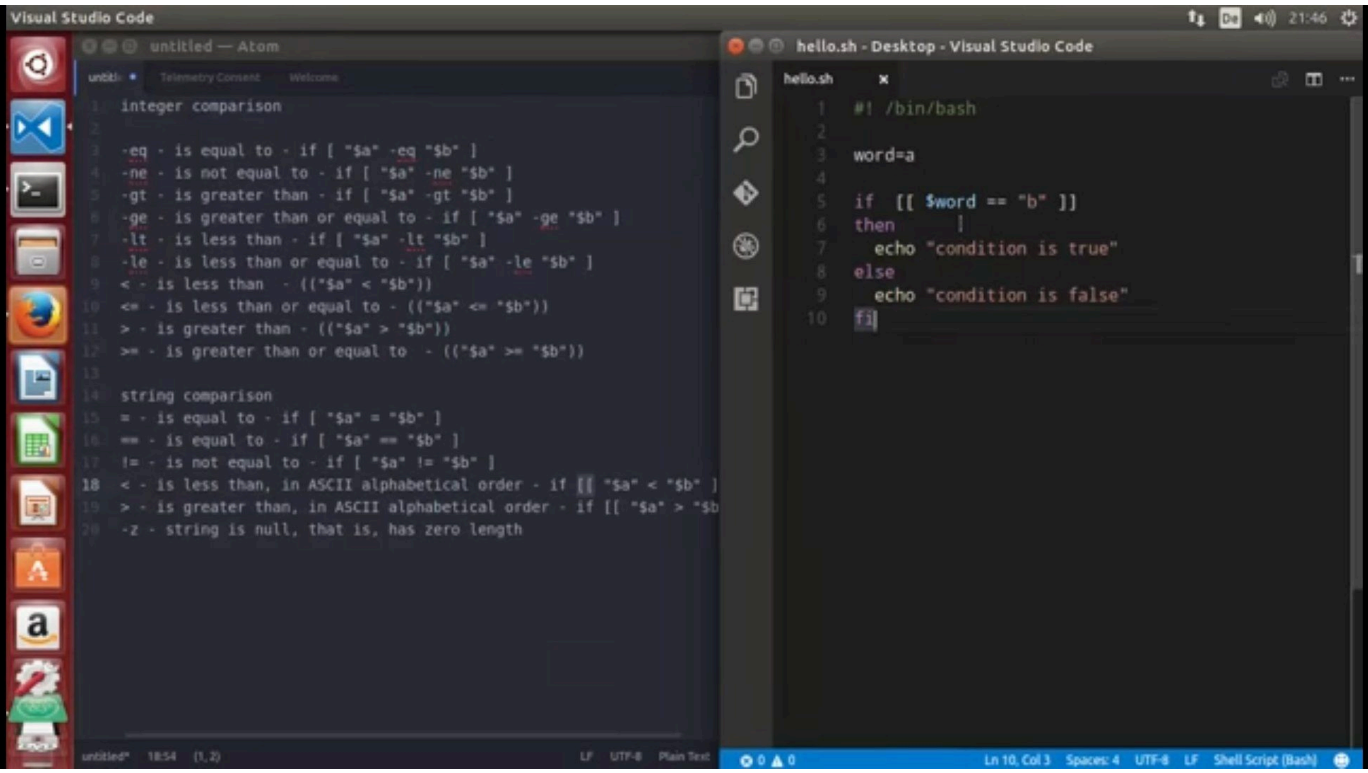

Atom
untitled — Atom
integer comparison
-eq - is equal to - if ["$a" -eq "$b"]
-ne - is not equal to - if ["$a" -ne "$b"]
-gt - is greater than - if ["$a" -gt "$b"]
-ge - is greater than or equal to - if ["$a" -ge "$b"]
-lt - is less than - if ["$a" -lt "$b"]
-le - is less than or equal to - if ["$a" -le "$b"]
< - is less than - (("$a" < "$b"))
<= - is less than or equal to - (("$a" <= "$b"))
> - is greater than - (("$a" > "$b"))
>= - is greater than or equal to - (("$a" >= "$b"))
string comparison
= - is equal to - if ["$a" = "$b"]
== - is equal to - if ["$a" == "$b"]
!= - is not equal to - if ["$a" != "$b"]
< - is less than, in ASCII alphabetical order - if [["$a" < "$b"]
> - is greater than, in ASCII alphabetical order - if [["$a" > "$b
-z - string is null, that is, has zero length
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
count=10
if [$count -gt 9]
then
echo "condition is true"
fi
Ln 5, Col 20
Spaces: 4
UTF-8
LF
Shell Script (Bash)

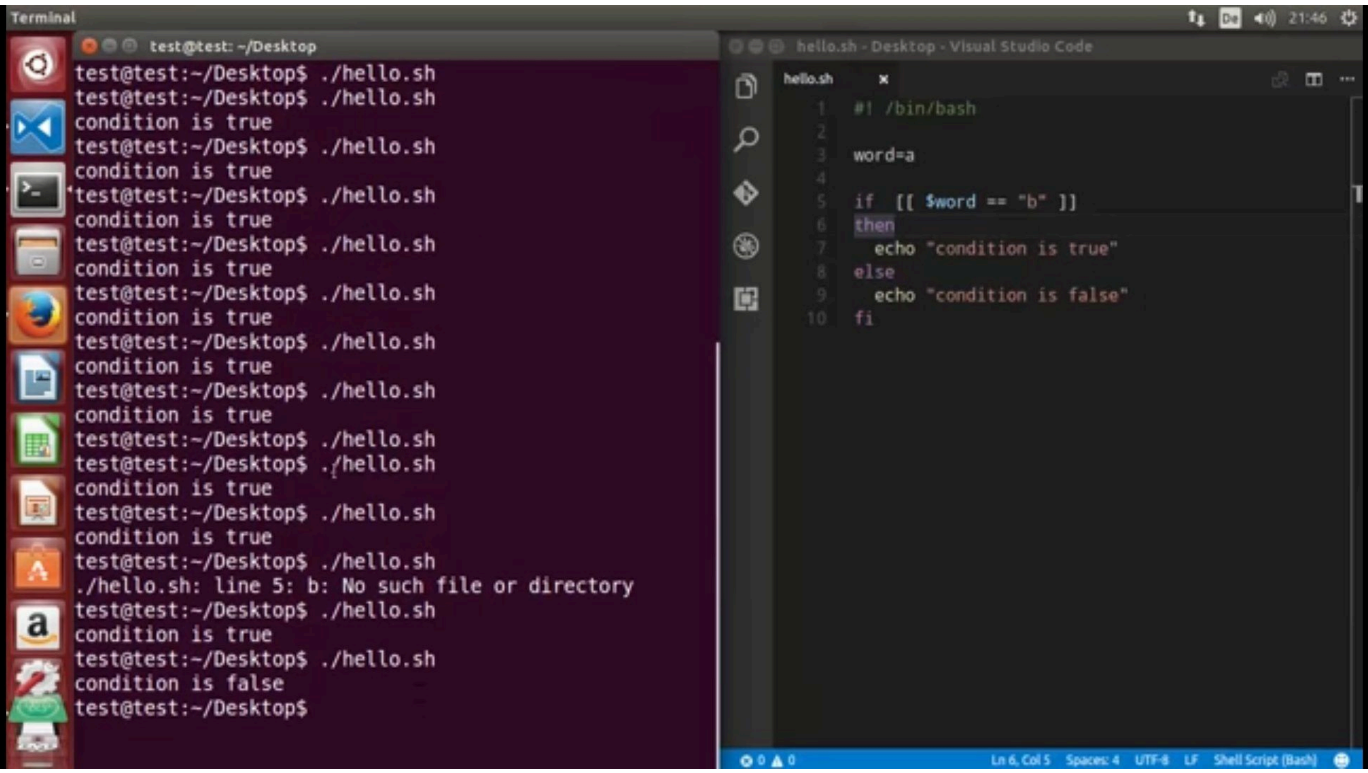

Visual Studio Code
21:40
untitled — Atom
integer comparison
-eq - is equal to - if ["$a" -eq "$b"]
-ne - is not equal to - if ["$a" -ne "$b"]
-gt - is greater than - if ["$a" -gt "$b"]
-ge - is greater than or equal to - if ["$a" -ge "$b"]
-lt - is less than - if ["$a" -lt "$b"]
-le - is less than or equal to - if ["$a" -le "$b"]
< - is less than - (("$a" < "$b"))
<= - is less than or equal to - (("$a" <= "$b"))
> - is greater than - (("$a" > "$b"))
>= - is greater than or equal to - (("$a" >= "$b"))
string comparison
= - is equal to - if ["$a" = "$b"]
== - is equal to - if ["$a" == "$b"]
!= - is not equal to - if ["$a" != "$b"]
< - is less than, in ASCII alphabetical order - if [["$a" < "$b"]
> - is greater than, in ASCII alphabetical order - if [["$a" > "$b
-z - string is null, that is, has zero length
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
count=10
if (($count >= 9))
then
echo "condition is true"
fi
Ln 5, Col 5 (6 selected)
Spaces: 4
UTF-8
LF
Shell Script (Bash)

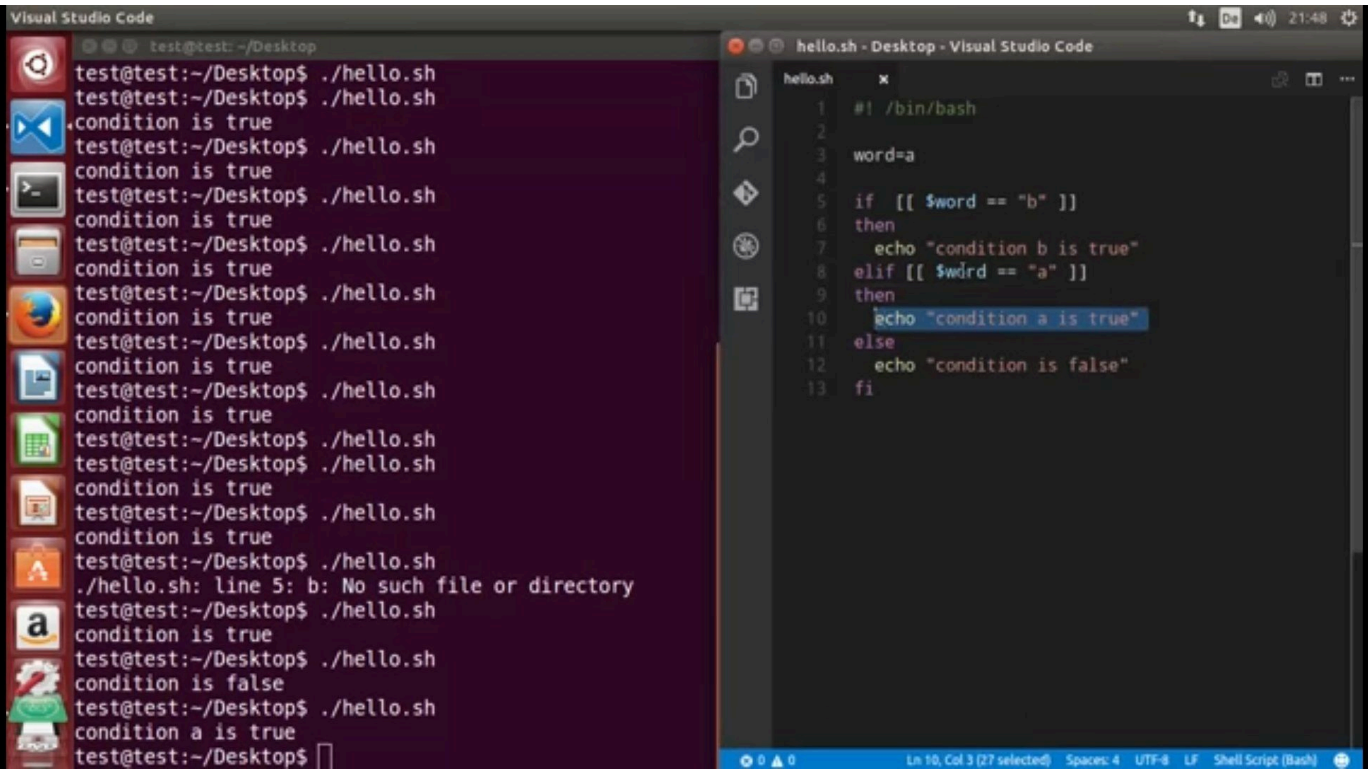

Terminal
21:41
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
word=abc
if [$word == "abc"]
then
echo "condition is true"
fi
Ln 6, Col 5
Spaces: 4
UTF-8
LF
Shell Script (Bash)

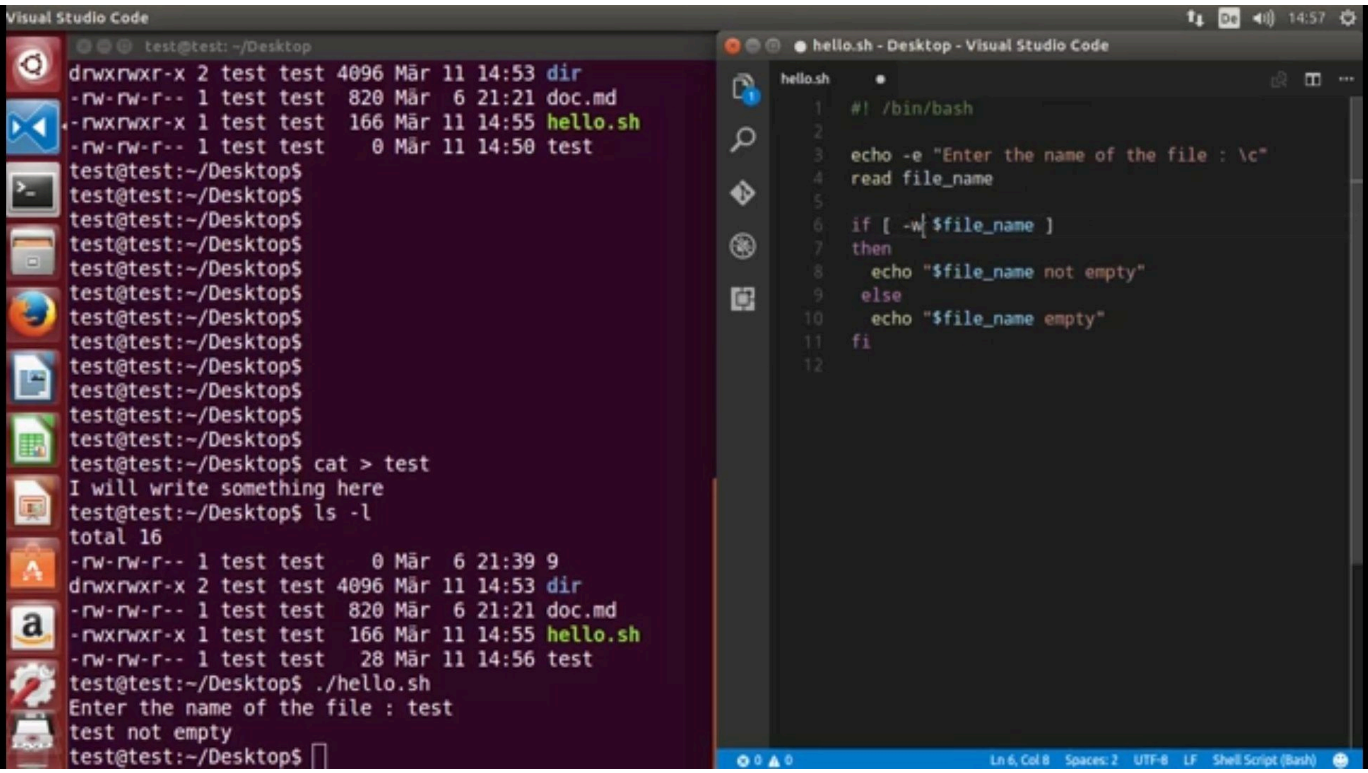

Visual Studio Code
21:41
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
word=abc
if [$word == "abcccc"]
then
echo "condition is true"
fi
0 0
Ln 7, Col 16
Spaces: 4
UTF-8
LF
Shell Script (Bash)

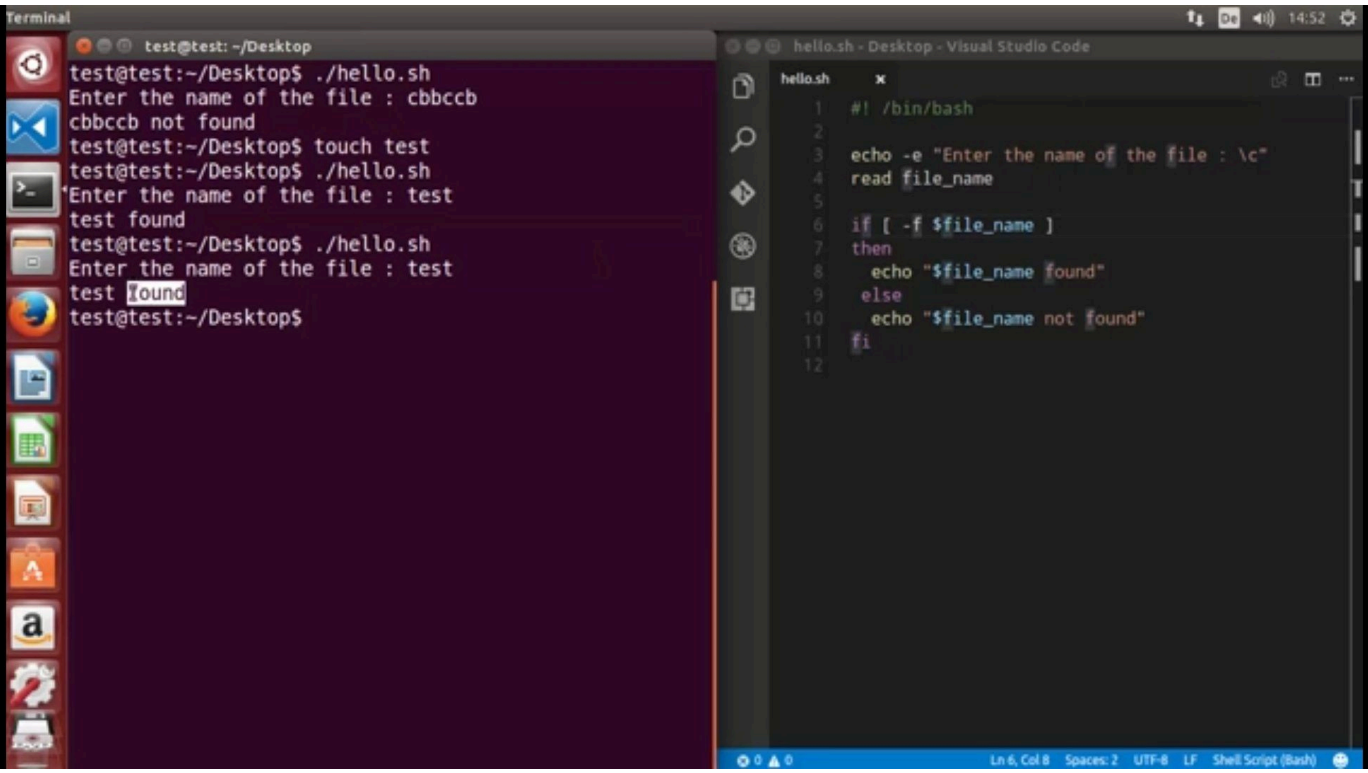

Visual Studio Code
21:45
untitled — Atom
integer comparison
-eq - is equal to - if ["$a" -eq "$b"]
-ne - is not equal to - if ["$a" -ne "$b"]
-gt - is greater than - if ["$a" -gt "$b"]
-ge - is greater than or equal to - if ["$a" -ge "$b"]
-lt - is less than - if ["$a" -lt "$b"]
-le - is less than or equal to - if ["$a" -le "$b"]
< - is less than - (("$a" < "$b"))
<= - is less than or equal to - (("$a" <= "$b"))
> - is greater than - (("$a" > "$b"))
>= - is greater than or equal to - (("$a" >= "$b"))
string comparison
= - is equal to - if ["$a" = "$b"]
== - is equal to - if ["$a" == "$b"]
!= - is not equal to - if ["$a" != "$b"]
< - is less than, in ASCII alphabetical order - if [["$a" < "$b"]]
> - is greater than, in ASCII alphabetical order - if [["$a" > "$b
-z - string is null, that is, has zero length
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
word=a
if [[$word < "b"]]
then
echo "condition is true"
fi
Ln 7, Col 27
Spaces: 4
UTF-8
LF
Shell Script (Bash)

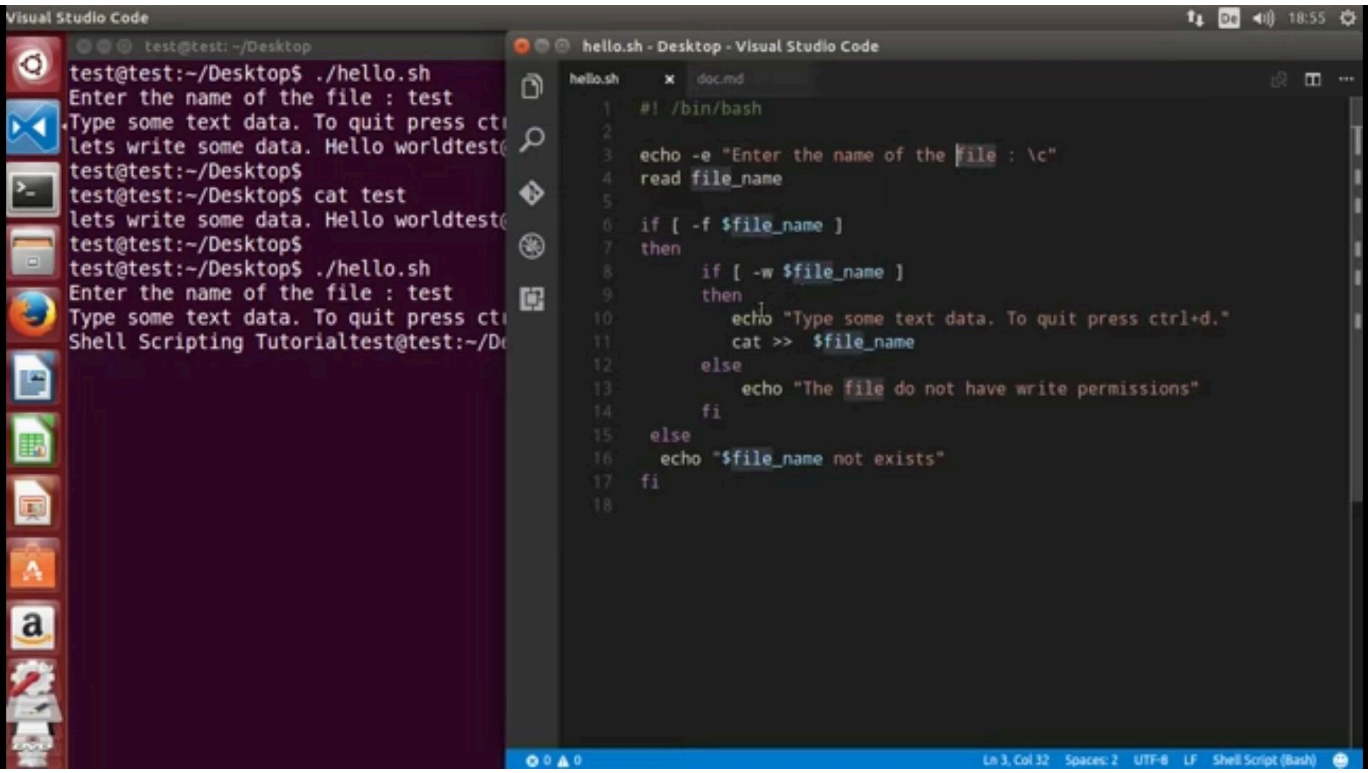

Visual Studio Code
21:46
untitled — Atom
integer comparison
-eq - is equal to - if ["$a" -eq "$b"]
-ne - is not equal to - if ["$a" -ne "$b"]
-gt - is greater than - if ["$a" -gt "$b"]
-ge - is greater than or equal to - if ["$a" -ge "$b"]
-lt - is less than - if ["$a" -lt "$b"]
-le - is less than or equal to - if ["$a" -le "$b"]
< - is less than - (("$a" < "$b"))
<= - is less than or equal to - (("$a" <= "$b"))
> - is greater than - (("$a" > "$b"))
>= - is greater than or equal to - (("$a" >= "$b"))
string comparison
= - is equal to - if ["$a" = "$b"]
== - is equal to - if ["$a" == "$b"]
!= - is not equal to - if ["$a" != "$b"]
< - is less than, in ASCII alphabetical order - if [["$a" < "$b"]]
> - is greater than, in ASCII alphabetical order - if [["$a" > "$b
-z - string is null, that is, has zero length
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
word=a
if [[$word == "b"]]
then
echo "condition is true"
else
echo "condition is false"
fi
Ln 10, Col 3
Spaces: 4
UTF-8
LF
Shell Script (Bash)

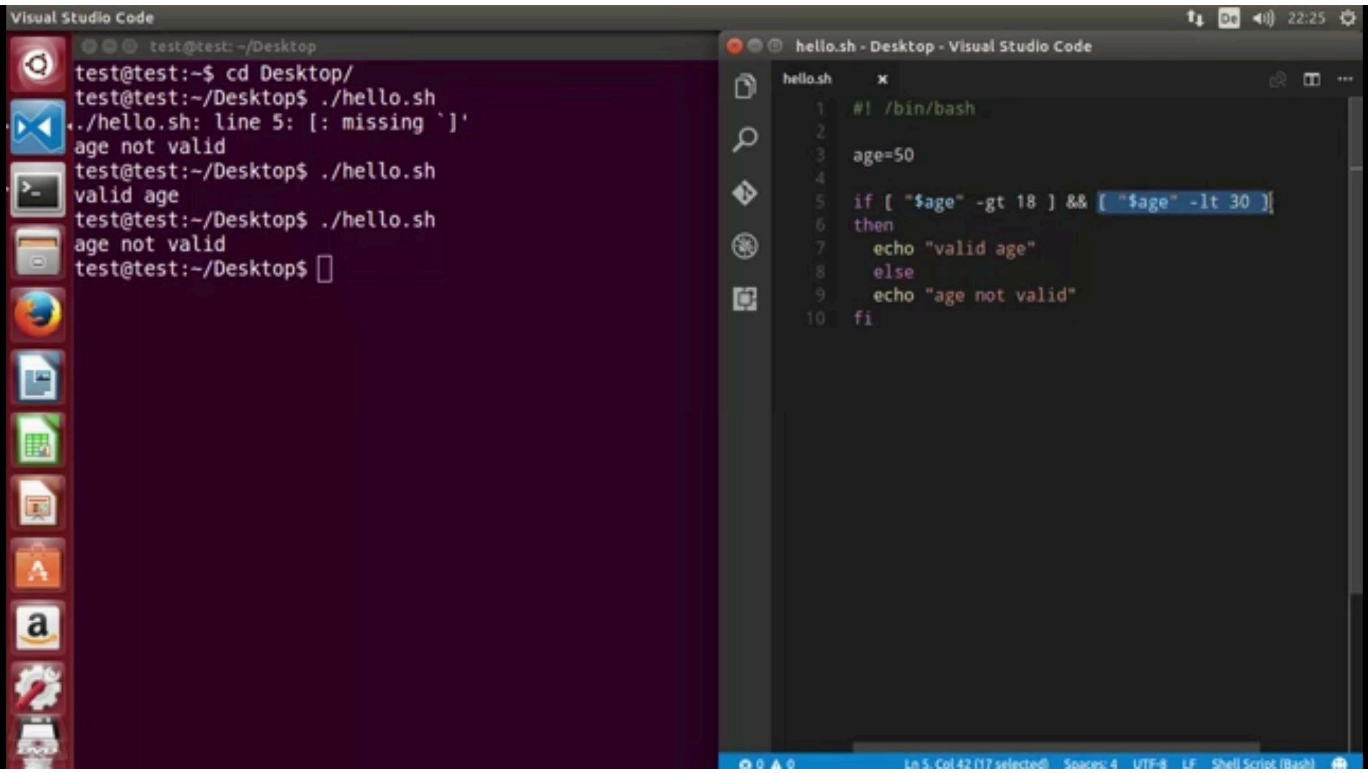

Terminal
21:46
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 5: b: No such file or directory
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is false
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
word=a
if [[$word == "b"]]
then
echo "condition is true"
else
echo "condition is false"
fi
Ln 6, Col 5
Spaces: 4
UTF-8
LF
Shell Script (Bash)

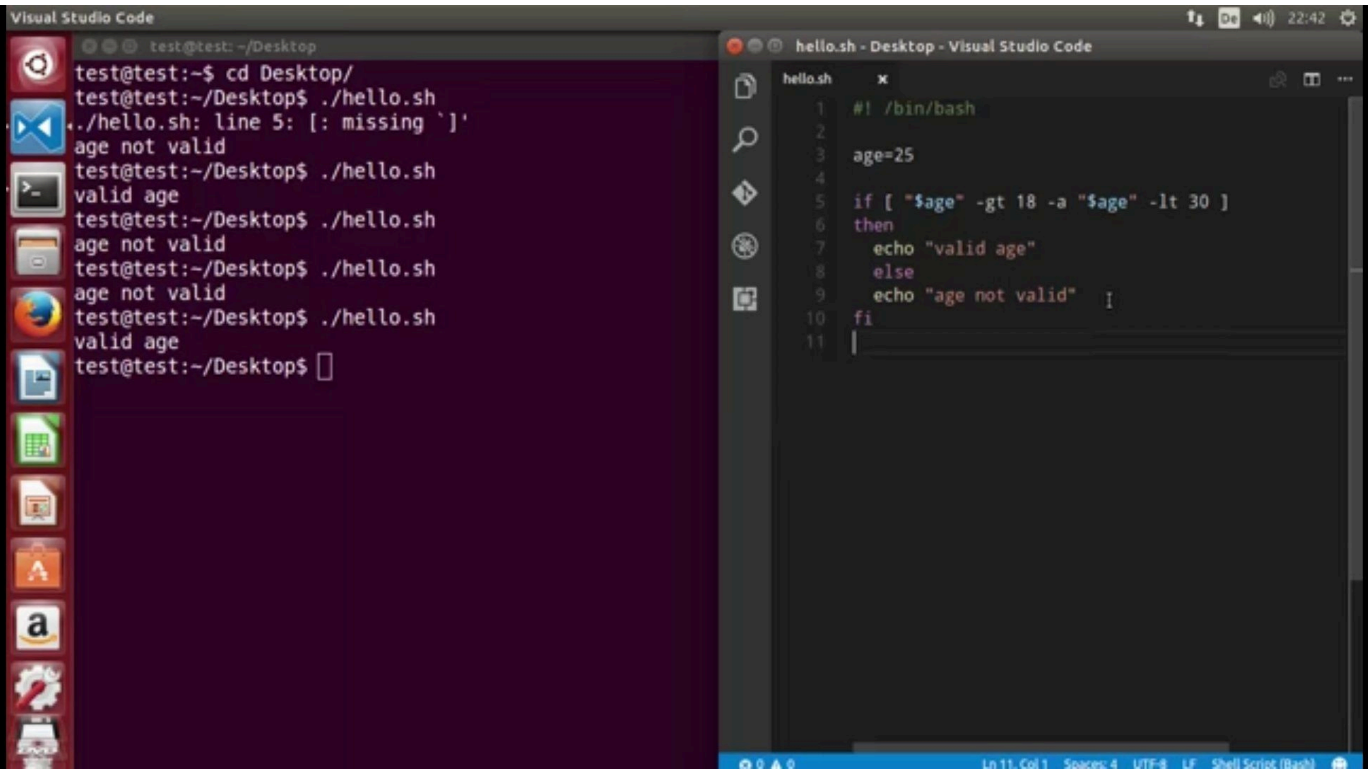

Visual Studio Code
21:48
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 5: b: No such file or directory
test@test:~/Desktop$ ./hello.sh
condition is true
test@test:~/Desktop$ ./hello.sh
condition is false
test@test:~/Desktop$ ./hello.sh
condition a is true
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
word=a
if [[$word == "b"]]
then
echo "condition b is true"
elif [[$word == "a"]]
then
echo "condition a is true"
else
echo "condition is false"
fi
Ln 10, Col 3 (27 selected) Spaces: 4 UTF-8 LF Shell Script (Bash)

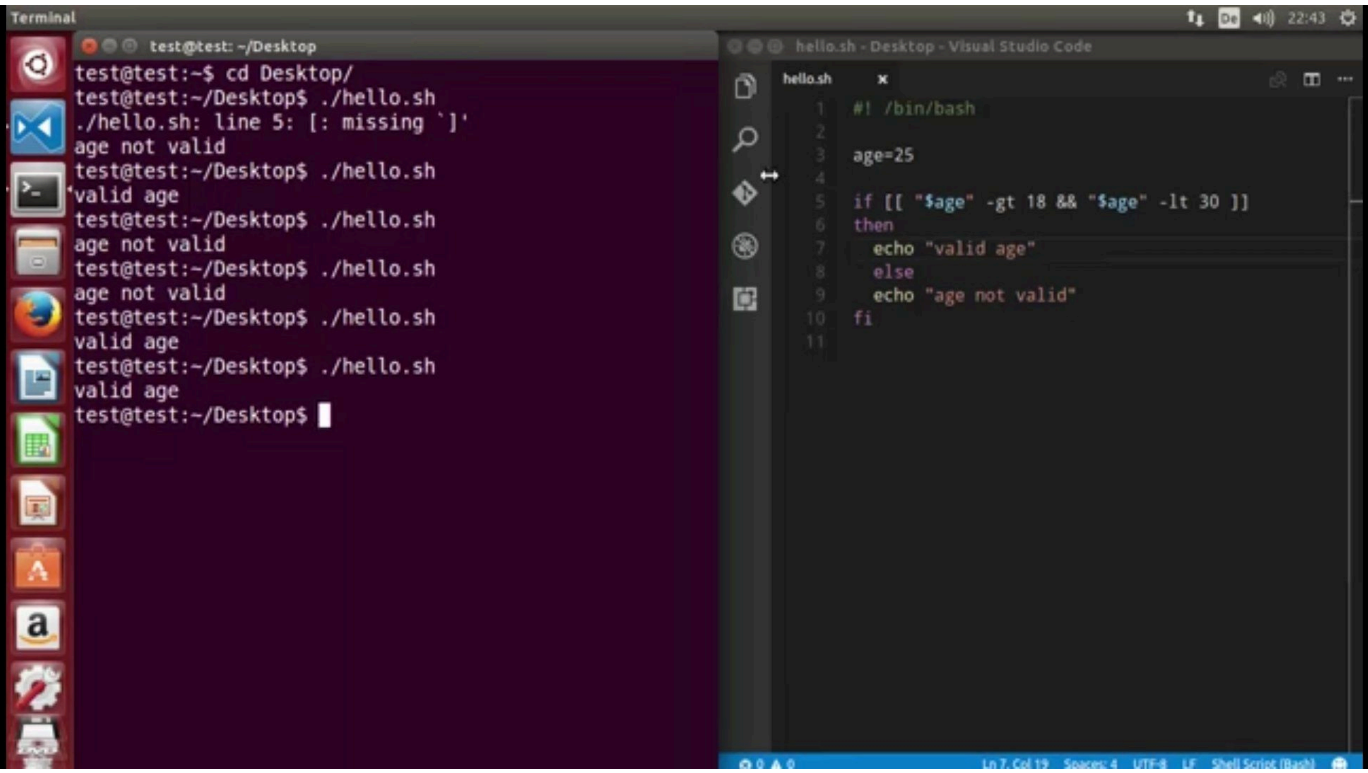

Visual Studio Code
14:57
test@test: ~/Desktop
drwxrwxr-x 2 test test 4096 Mär 11 14:53 dir
-rw-rw-r-- 1 test test 820 Mär 6 21:21 doc.md
-rwxrwxr-x 1 test test 166 Mär 11 14:55 hello.sh
-rw-rw-r-- 1 test test 0 Mär 11 14:50 test
test@test:~/Desktop$
test@test:~/Desktop$ cat > test
I will write something here
test@test:~/Desktop$ ls -l
total 16
-rw-rw-r-- 1 test test 0 Mär 6 21:39 9
drwxrwxr-x 2 test test 4096 Mär 11 14:53 dir
-rw-rw-r-- 1 test test 820 Mär 6 21:21 doc.md
-rwxrwxr-x 1 test test 166 Mär 11 14:55 hello.sh
-rw-rw-r-- 1 test test 28 Mär 11 14:56 test
test@test:~/Desktop$ ./hello.sh
Enter the name of the file : test
test not empty
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
echo -e "Enter the name of the file : \c"
read file_name
if [-w $file_name]
then
echo "$file_name not empty"
else
echo "$file_name empty"
fi
Ln 6, Col 8 Spaces: 2 UTF-8 LF Shell Script (Bash)

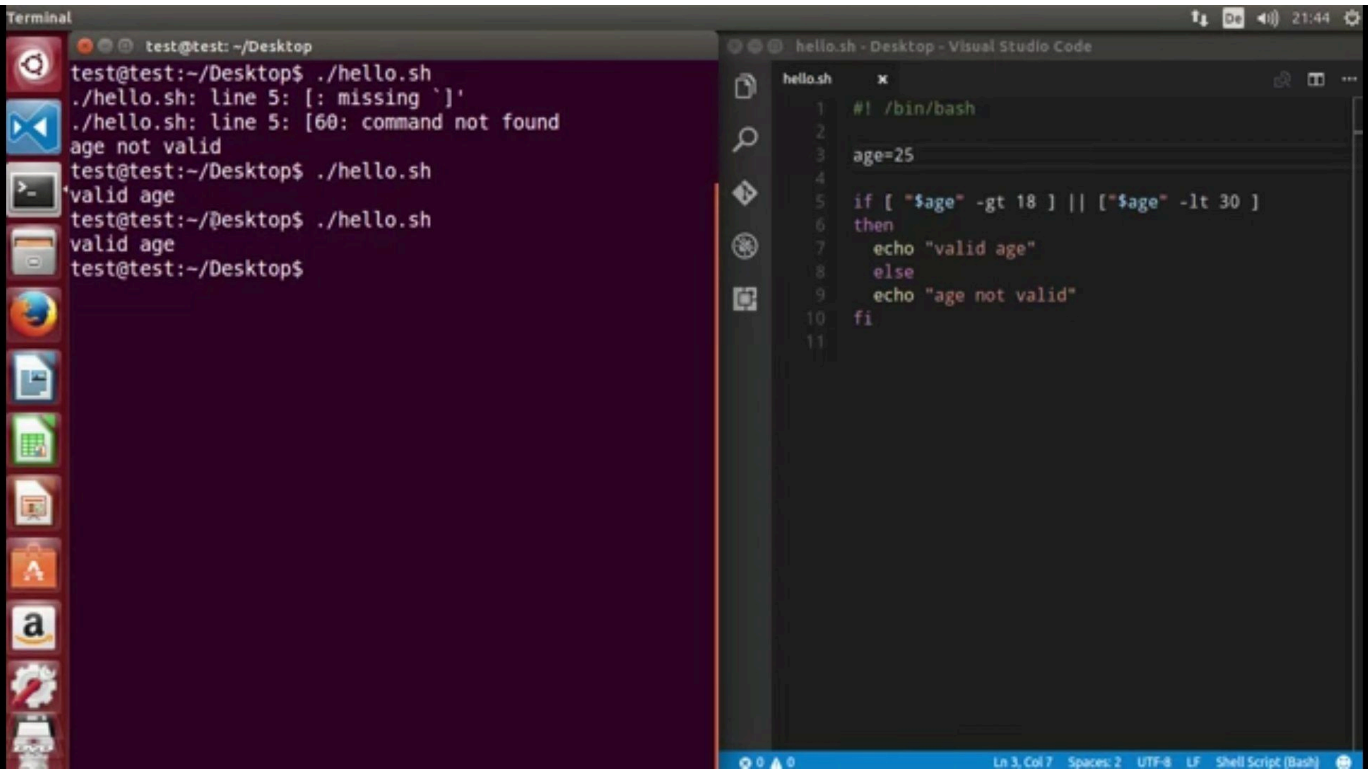

Terminal
14:52
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter the name of the file : cbbccb
cbbccb not found
test@test:~/Desktop$ touch test
test@test:~/Desktop$ ./hello.sh
Enter the name of the file : test
test found
test@test:~/Desktop$ ./hello.sh
Enter the name of the file : test
test found
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
echo -e "Enter the name of the file : \c"
read file_name
if [-f $file_name]
then
echo "$file_name found"
else
echo "$file_name not found"
fi
Ln 6, Col 8
Spaces: 2
UTF-8
LF
Shell Script (Bash)

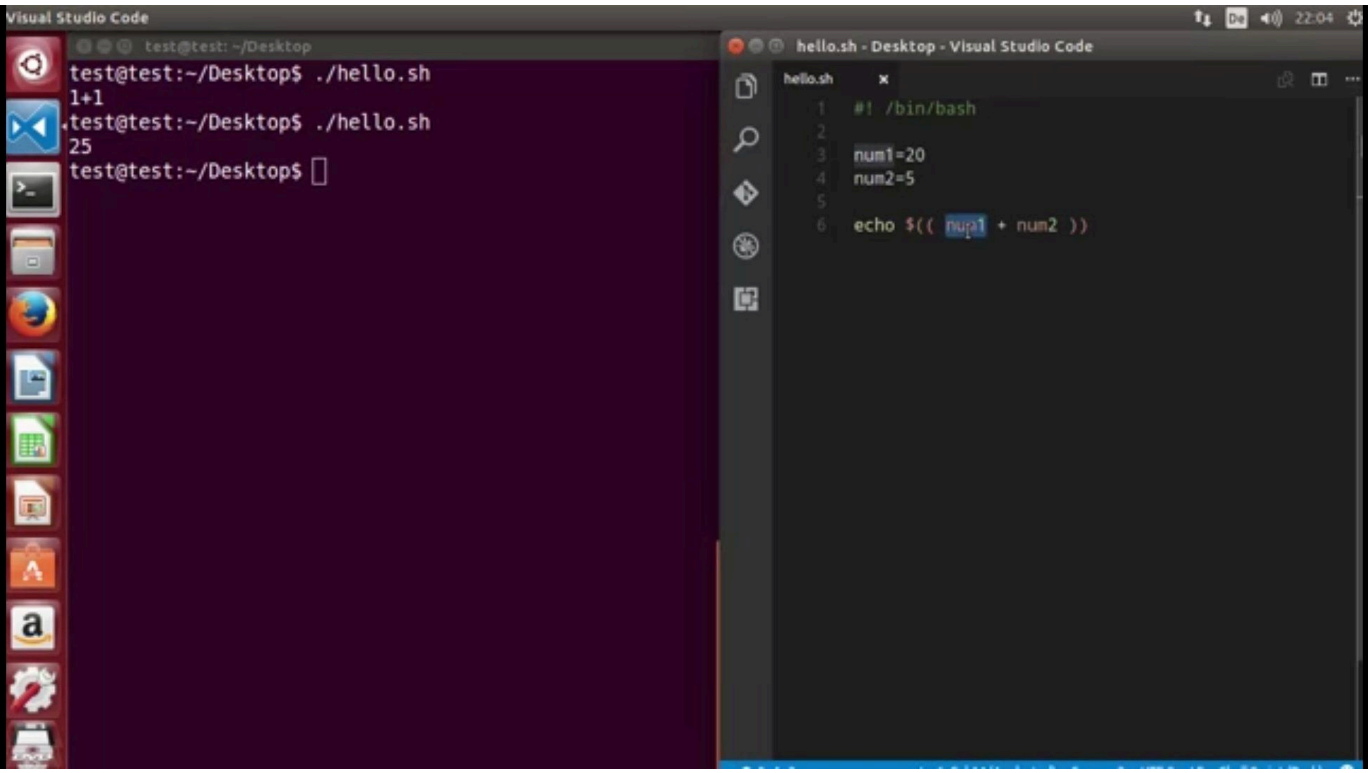

Visual Studio Code
18:55
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter the name of the file : test
Type some text data. To quit press ctr
lets write some data. Hello worldtest
test@test:~/Desktop$
test@test:~/Desktop$ cat test
lets write some data. Hello worldtest
test@test:~/Desktop$
test@test:~/Desktop$ ./hello.sh
Enter the name of the file : test
Type some text data. To quit press ctr
Shell Scripting Tutorialtest@test:~/D
hello.sh - Desktop - Visual Studio Code
hello.sh
doc.md
#! /bin/bash
echo -e "Enter the name of the file : \c"
read file_name
if [-f $file_name]
then
if [-w $file_name]
then
echo "Type some text data. To quit press ctrl+d."
cat >> $file_name
else
echo "The file do not have write permissions"
fi
else
echo "$file_name not exists"
fi
Ln 3, Col 32
Spaces: 2
UTF-8
LF
Shell Script (Bash)

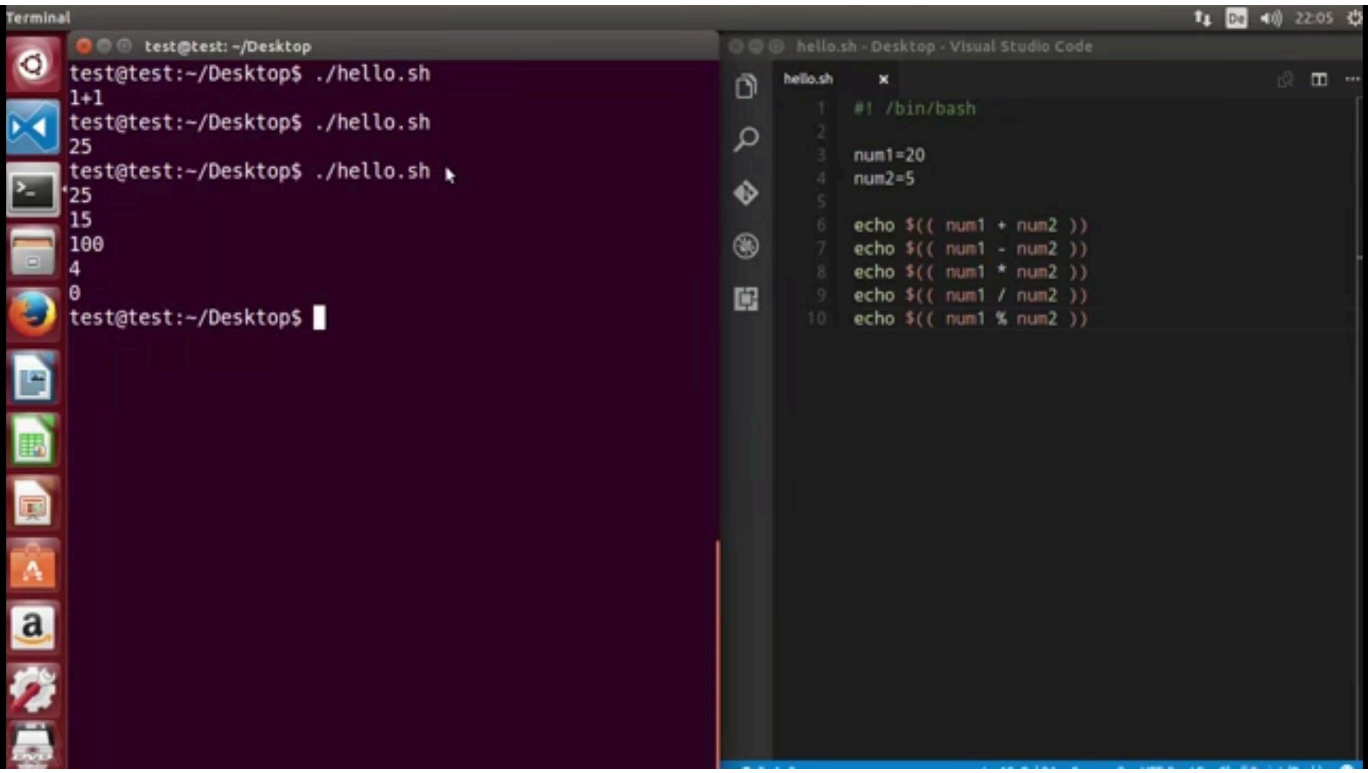

Visual Studio Code
22:25
test@test: ~/Desktop
test@test:~$ cd Desktop/
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 5: [: missing `]'
age not valid
test@test:~/Desktop$ ./hello.sh
valid age
test@test:~/Desktop$ ./hello.sh
age not valid
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
age=50
if ["$age" -gt 18] && ["$age" -lt 30]
then
echo "valid age"
else
echo "age not valid"
fi

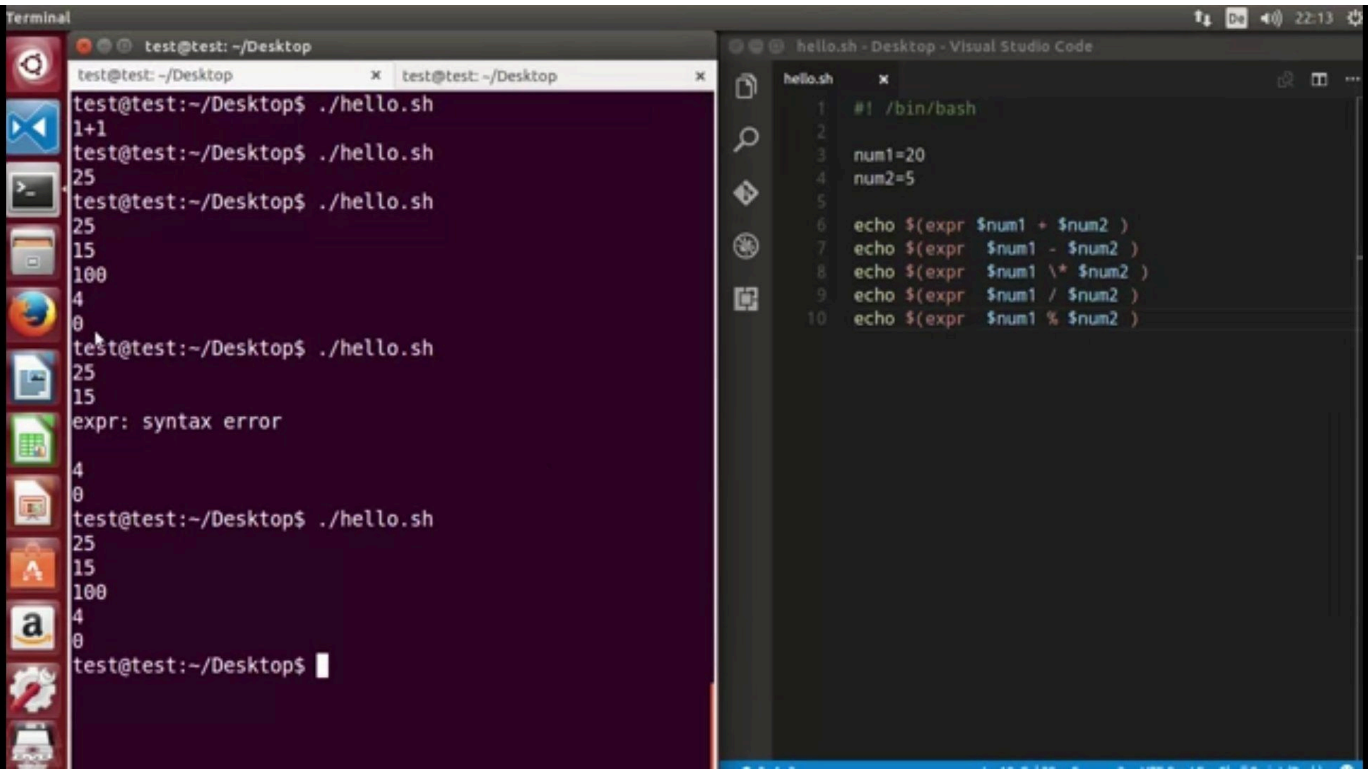

Visual Studio Code
22:42
test@test: ~/Desktop
test@test:~$ cd Desktop/
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 5: [: missing `]'
age not valid
test@test:~/Desktop$ ./hello.sh
valid age
test@test:~/Desktop$ ./hello.sh
age not valid
test@test:~/Desktop$ ./hello.sh
age not valid
test@test:~/Desktop$ ./hello.sh
valid age
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
age=25
if ["$age" -gt 18 -a "$age" -lt 30]
then
echo "valid age"
else
echo "age not valid"
fi
Ln 11, Col 1
Spaces: 4
UTF-8
LF
Shell Script (Bash)

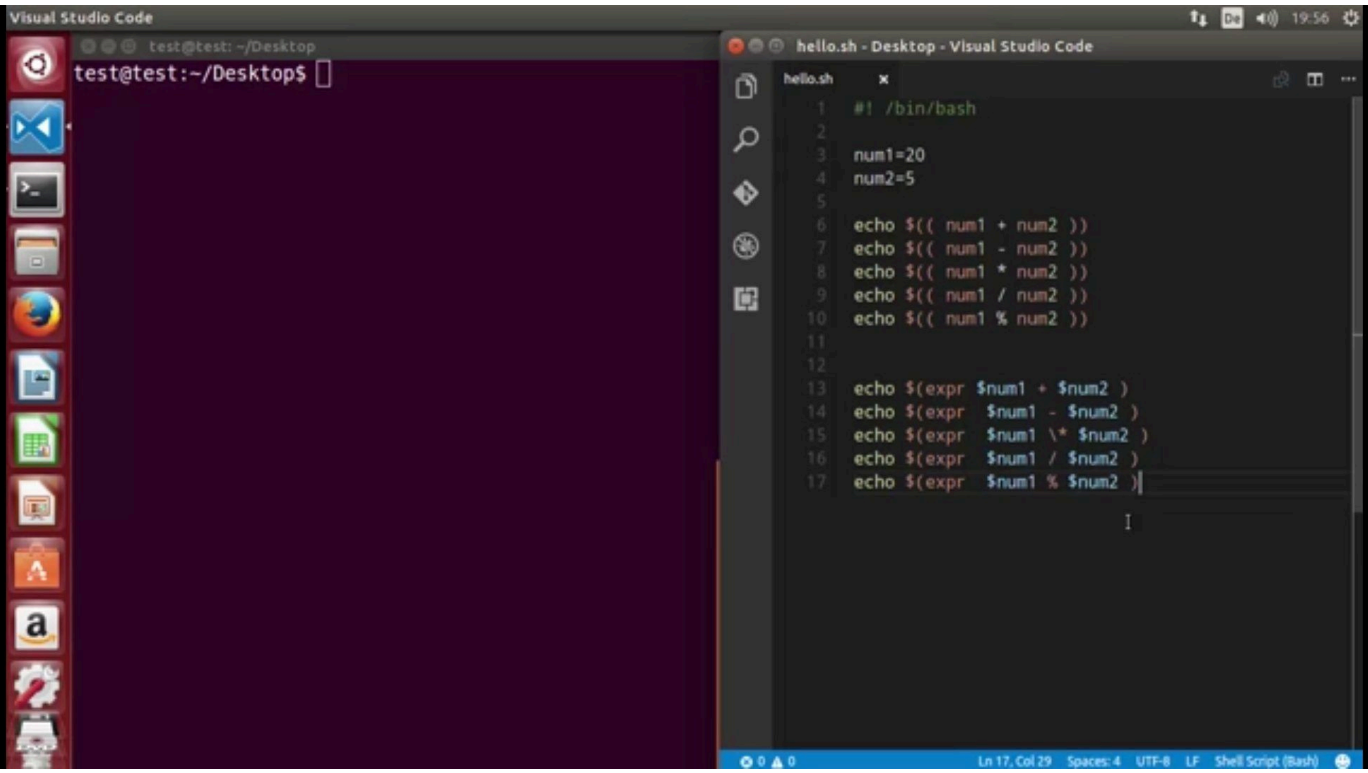

Terminal
22:43
test@test: ~/Desktop
test@test:~$ cd Desktop/
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 5: [: missing `]'
age not valid
test@test:~/Desktop$ ./hello.sh
valid age
test@test:~/Desktop$ ./hello.sh
age not valid
test@test:~/Desktop$ ./hello.sh
age not valid
test@test:~/Desktop$ ./hello.sh
valid age
test@test:~/Desktop$ ./hello.sh
valid age
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
age=25
if [["$age" -gt 18 && "$age" -lt 30]]
then
echo "valid age"
else
echo "age not valid"
fi

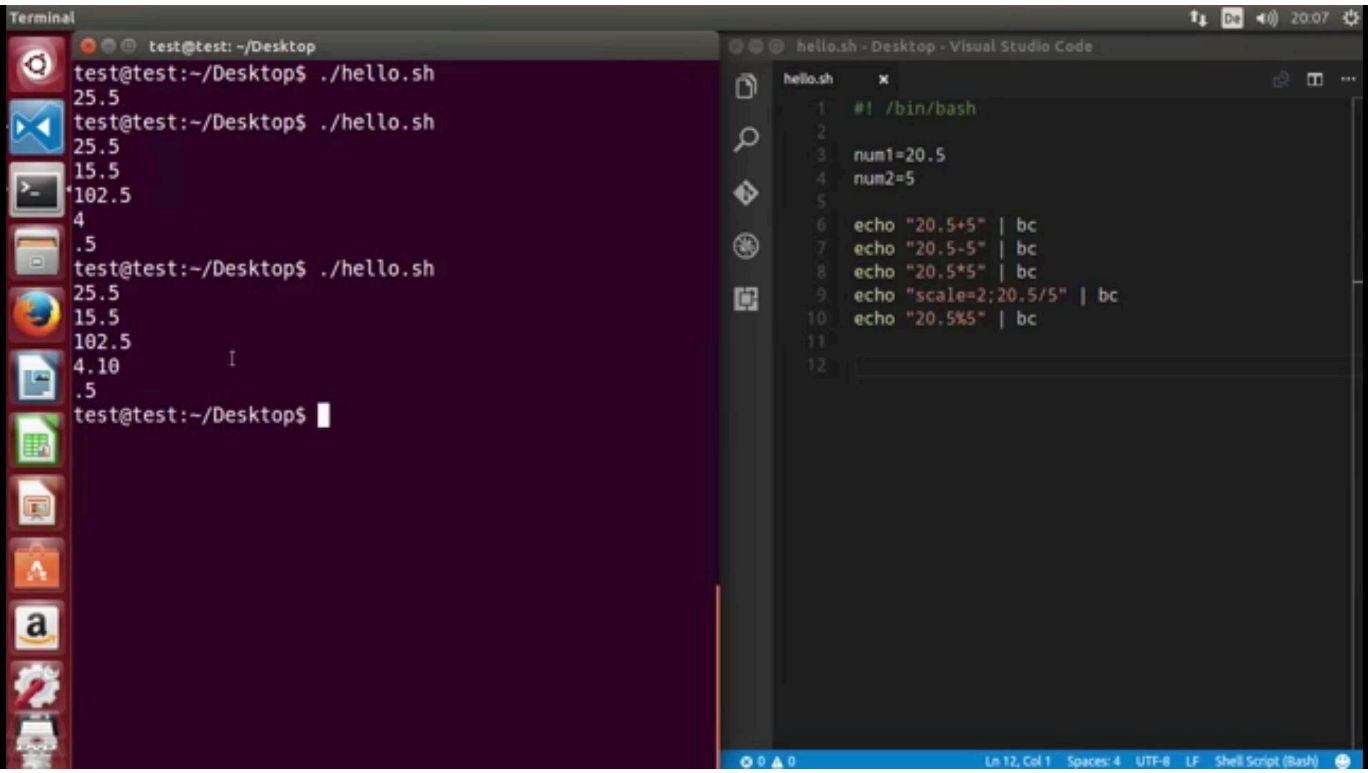

Terminal
21:44
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 5: [: missing `]'
./hello.sh: line 5: [60: command not found
age not valid
test@test:~/Desktop$ ./hello.sh
valid age
test@test:~/Desktop$ ./hello.sh
valid age
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
age=25
if ["$age" -gt 18] || ["$age" -lt 30]
then
echo "valid age"
else
echo "age not valid"
fi
Ln 3, Col 7
Spaces: 2
UTF-8
LF
Shell Script (Bash)

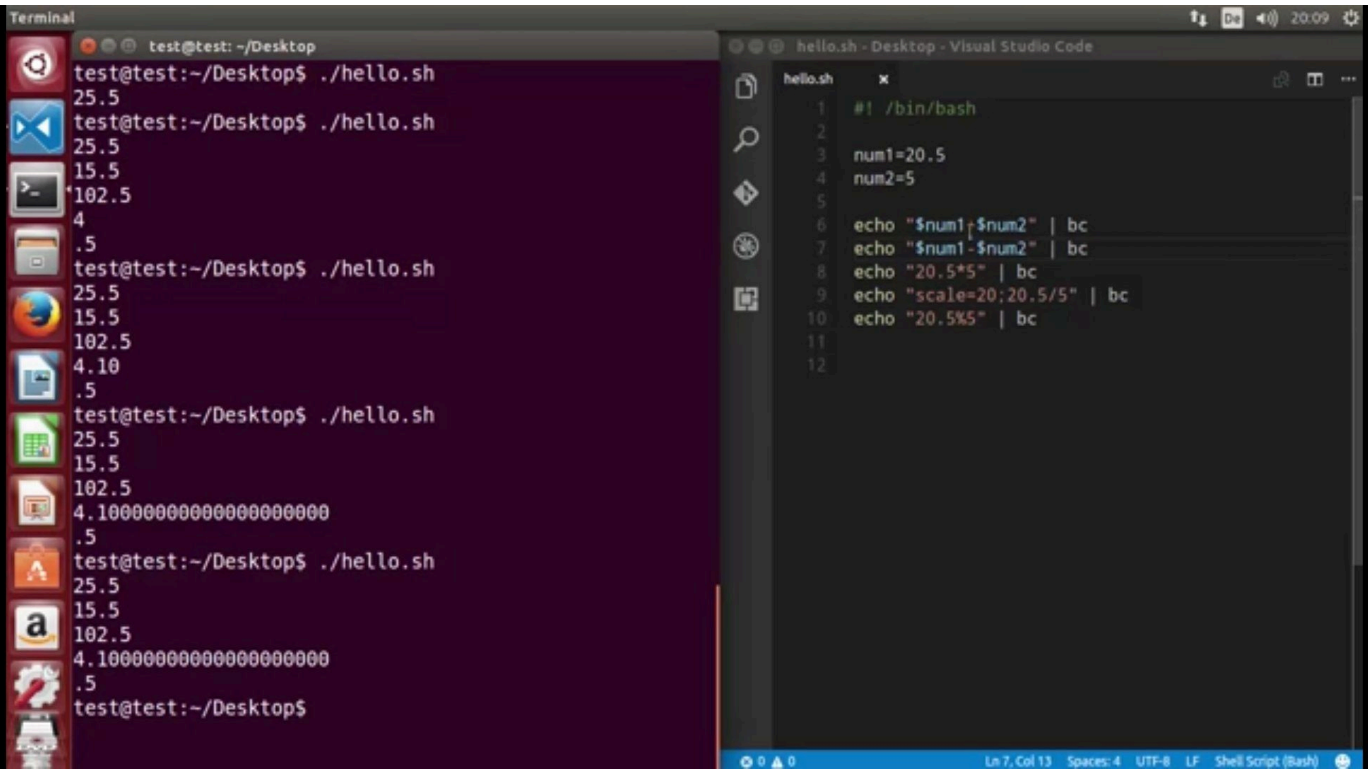

Visual Studio Code
22:04
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1+1
test@test:~/Desktop$ ./hello.sh
25
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
num1=20
num2=5
echo $((num1 + num2))

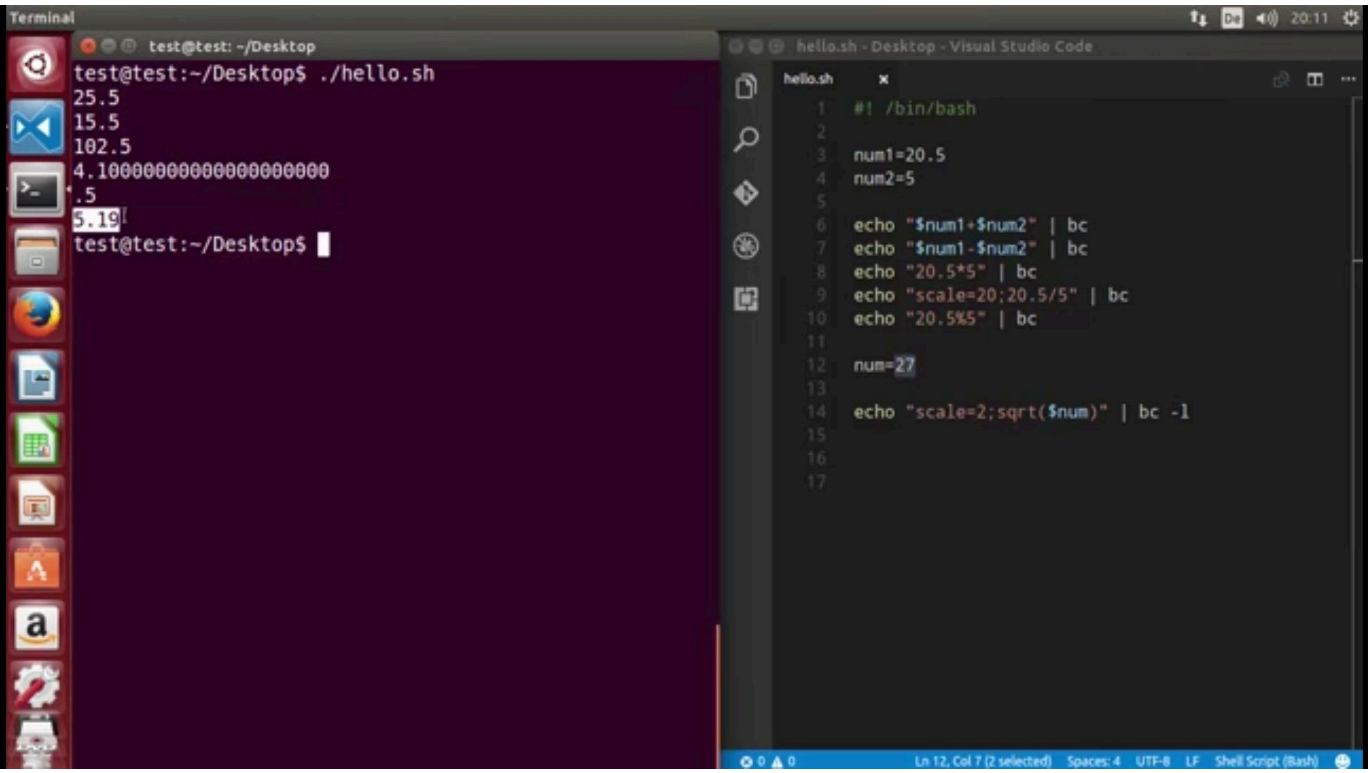

Terminal
22:05
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1+1
test@test:~/Desktop$ ./hello.sh
25
test@test:~/Desktop$ ./hello.sh
25
15
100
4
0
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
num1=20
num2=5
echo $((num1 + num2))
echo $((num1 - num2))
echo $((num1 * num2))
echo $((num1 / num2))
echo $((num1 % num2))

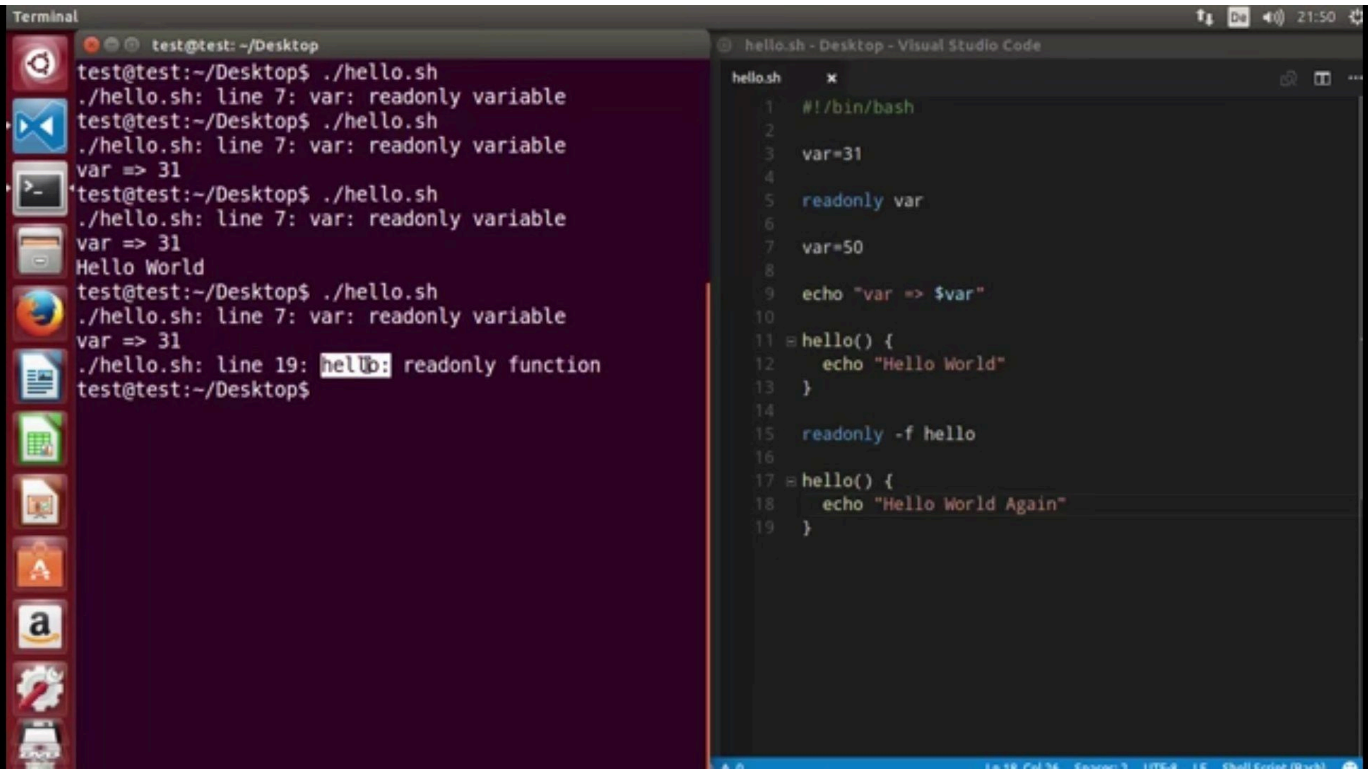

Terminal
22:13
test@test: ~/Desktop
test@test: ~/Desktop
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1+1
test@test:~/Desktop$ ./hello.sh
25
test@test:~/Desktop$ ./hello.sh
25
15
100
4
0
test@test:~/Desktop$ ./hello.sh
25
15
expr: syntax error
4
0
test@test:~/Desktop$ ./hello.sh
25
15
100
4
0
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
num1=20
num2=5
echo $(expr $num1 + $num2)
echo $(expr $num1 - $num2)
echo $(expr $num1 * $num2)
echo $(expr $num1 / $num2)
echo $(expr $num1 % $num2)

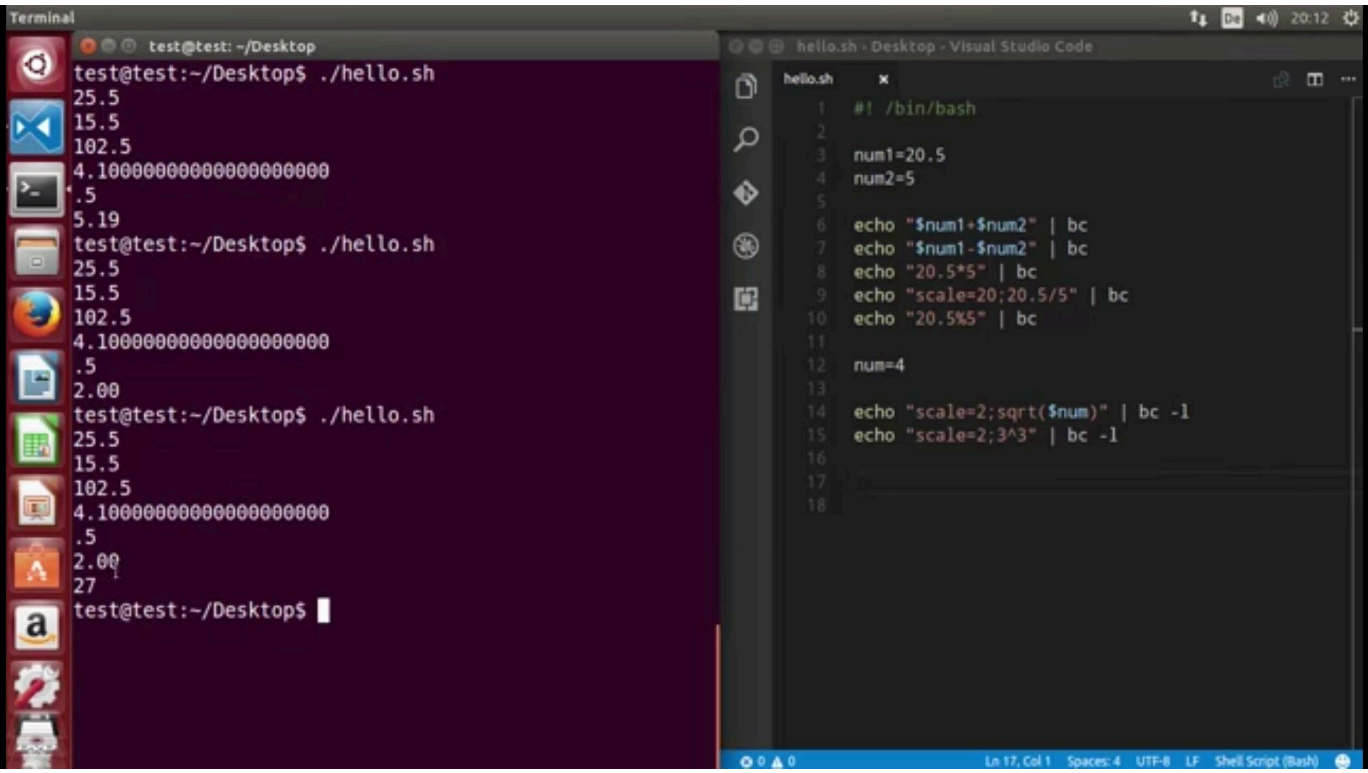

Visual Studio Code
19:56
test@test: ~/Desktop
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
num1=20
num2=5
echo $((num1 + num2))
echo $((num1 - num2))
echo $((num1 * num2))
echo $((num1 / num2))
echo $((num1 % num2))
echo $(expr $num1 + $num2)
echo $(expr $num1 - $num2)
echo $(expr $num1 * $num2)
echo $(expr $num1 / $num2)
echo $(expr $num1 % $num2)
Ln 17, Col 29
Spaces: 4
UTF-8
LF
Shell Script (Bash)

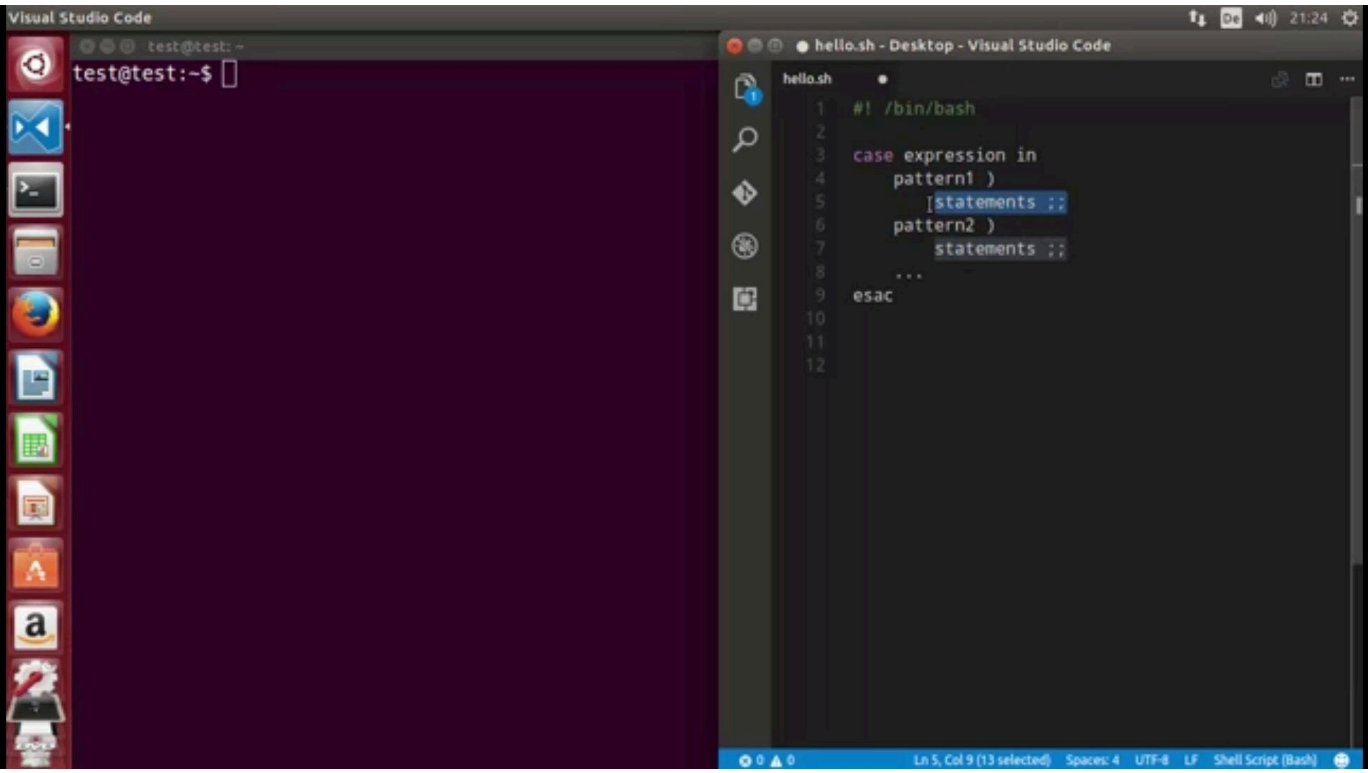

Terminal
20:07
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
25.5
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4
.5
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4.10
.5
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
num1=20.5
num2=5
echo "20.5+5" | bc
echo "20.5-5" | bc
echo "20.5*5" | bc
echo "scale=2;20.5/5" | bc
echo "20.5%5" | bc
Ln 12, Col 1
Spaces: 4
UTF-8
LF
Shell Script (Bash)

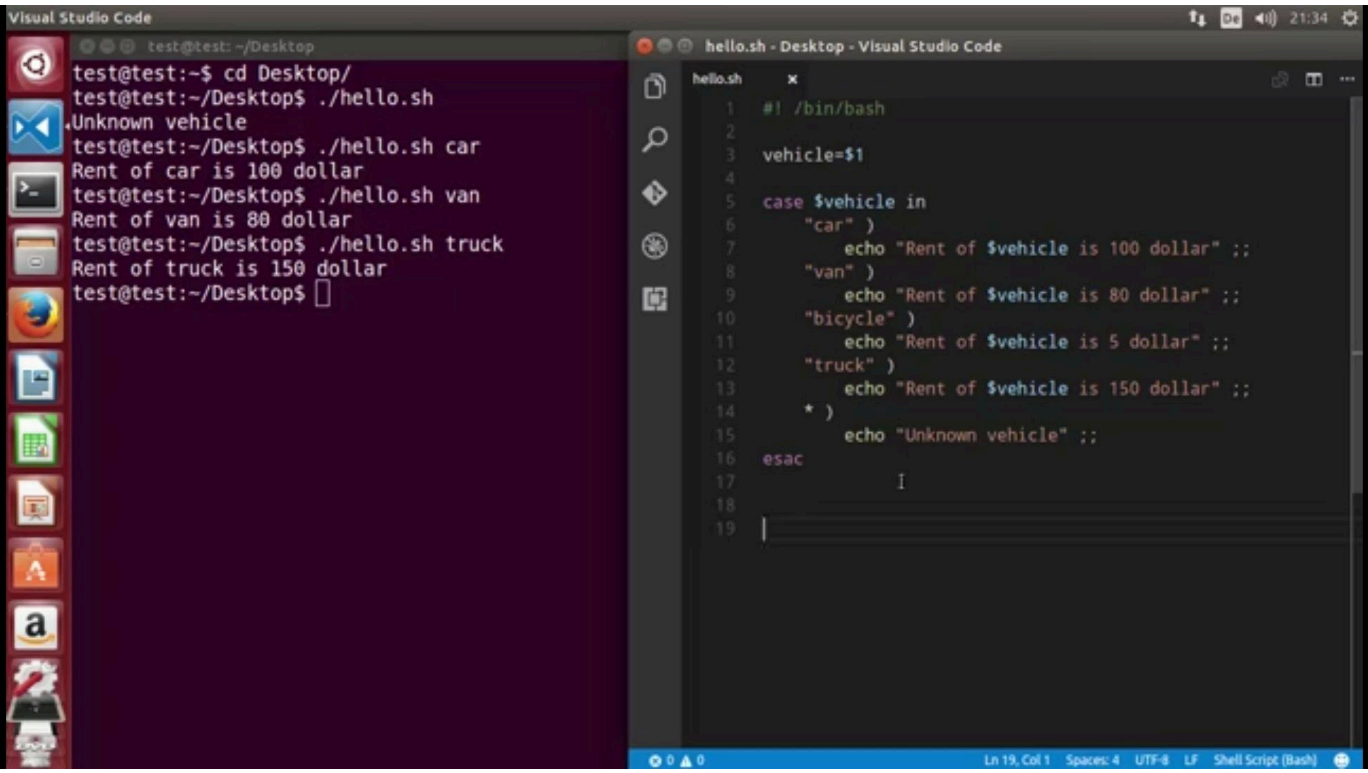

Terminal
20:09
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
25.5
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4
.5
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4.10
.5
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4.10000000000000000000
.5
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4.10000000000000000000
.5
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
num1=20.5
num2=5
echo "$num1+$num2" | bc
echo "$num1-$num2" | bc
echo "20.5*5" | bc
echo "scale=20;20.5/5" | bc
echo "20.5%5" | bc
Ln 7, Col 13
Spaces: 4
UTF-8
LF
Shell Script (Bash)

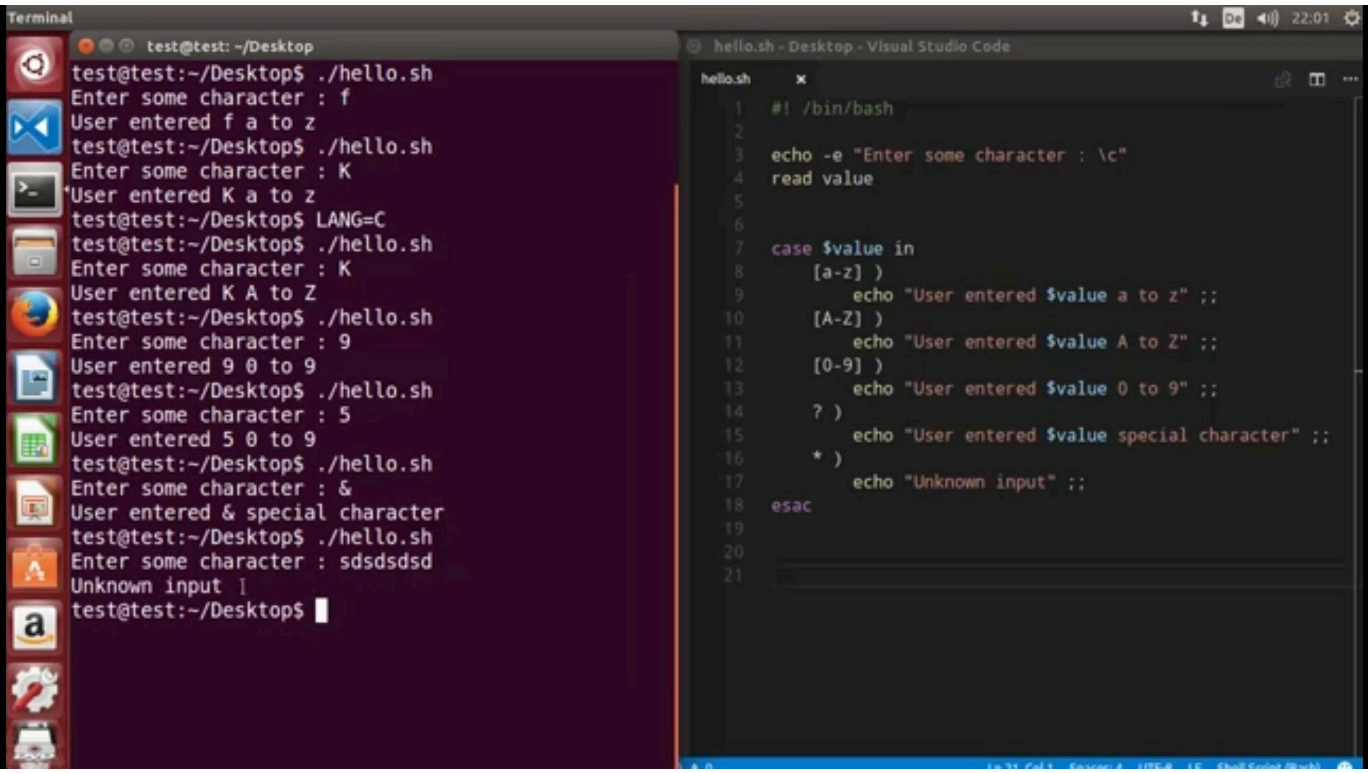

Terminal
20:11
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4.10000000000000000000
.5
5.19
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
num1=20.5
num2=5
echo "$num1+$num2" | bc
echo "$num1-$num2" | bc
echo "20.5*5" | bc
echo "scale=20;20.5/5" | bc
echo "20.5%5" | bc
num=27
echo "scale=2;sqrt($num)" | bc -l
Ln 12, Col 7 (2 selected) Spaces: 4 UTF-8 LF Shell Script (Bash)

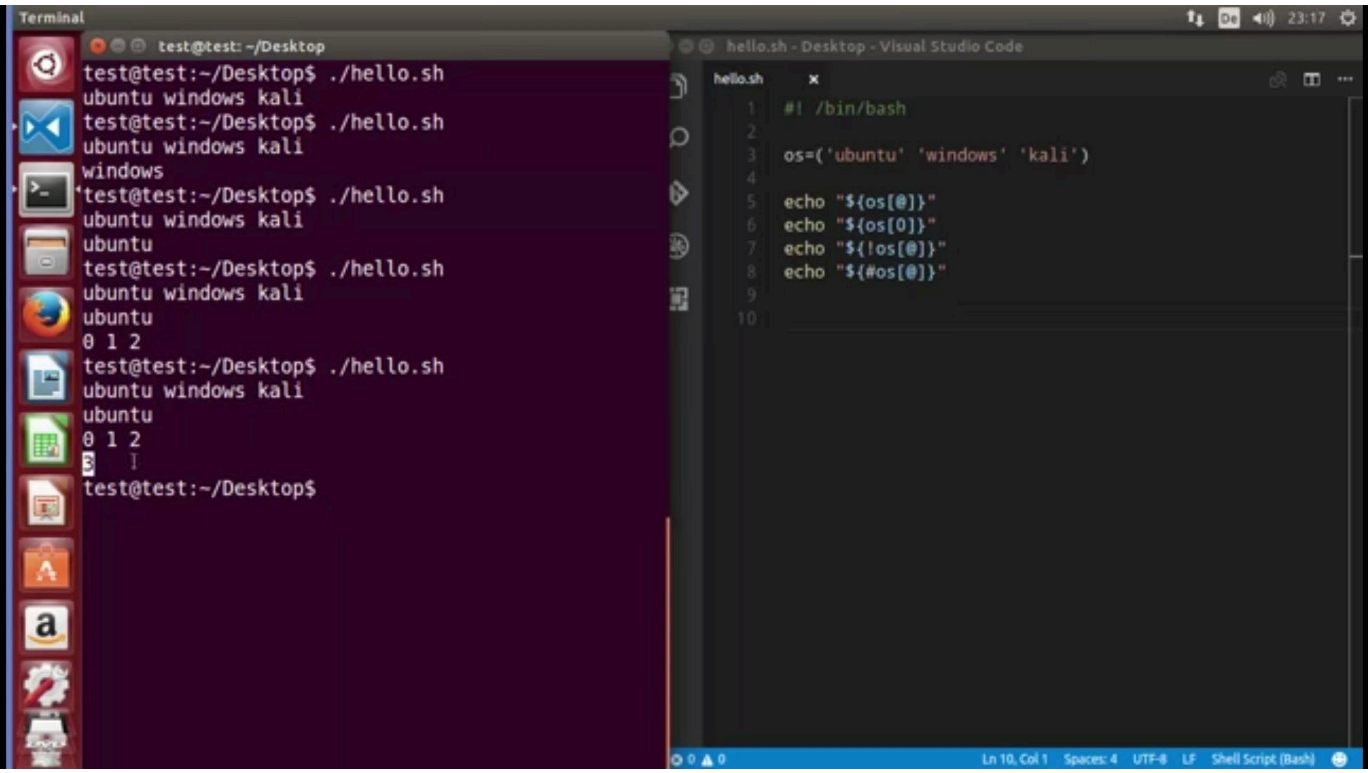

Terminal
21:50
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 7: var: readonly variable
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 7: var: readonly variable
var => 31
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 7: var: readonly variable
var => 31
Hello World
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 7: var: readonly variable
var => 31
./hello.sh: line 19: hello: readonly function
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
var=31
readonly var
var=50
echo "var => $var"
hello() {
echo "Hello World"
}
readonly -f hello
hello() {
echo "Hello World Again"
}

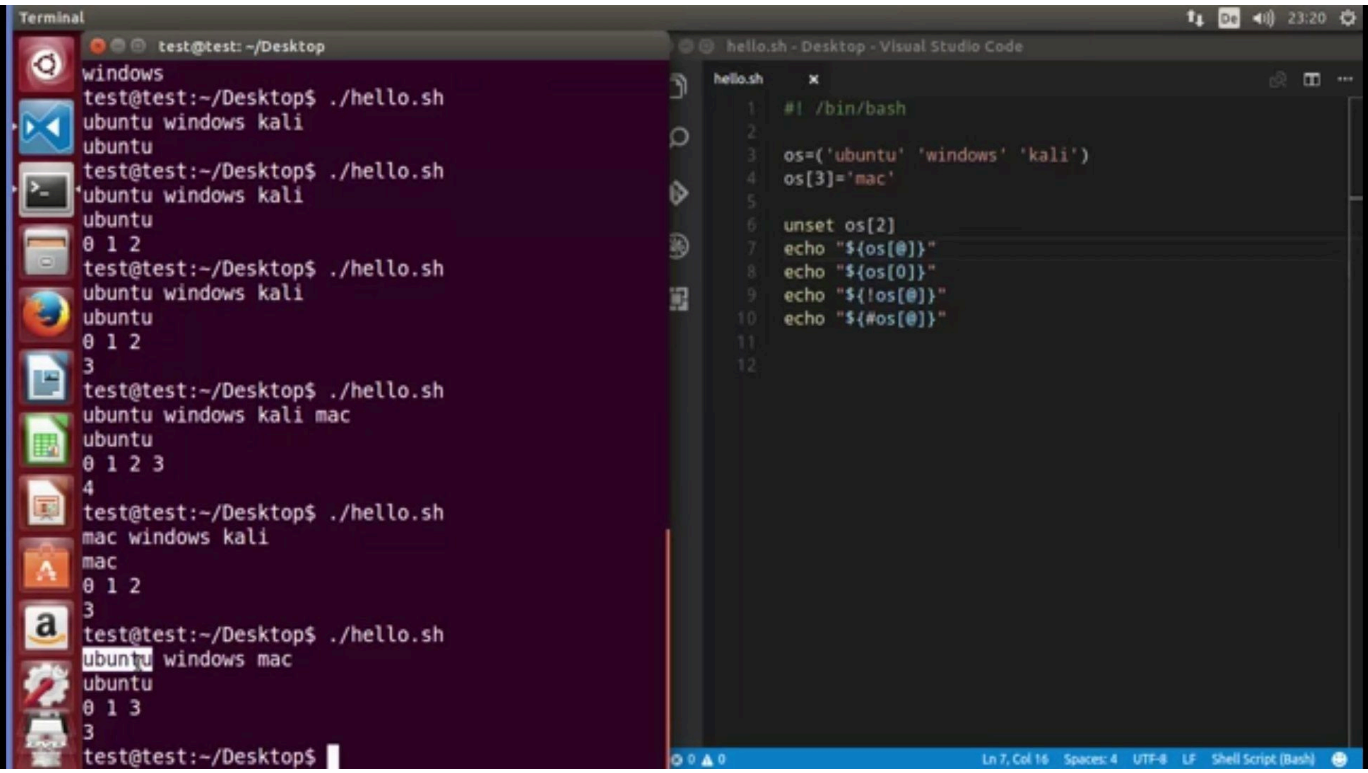

Terminal
20:12
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4.10000000000000000000
.5
5.19
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4.10000000000000000000
.5
2.00
test@test:~/Desktop$ ./hello.sh
25.5
15.5
102.5
4.10000000000000000000
.5
2.00
27
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
num1=20.5
num2=5
echo "$num1+$num2" | bc
echo "$num1-$num2" | bc
echo "20.5*5" | bc
echo "scale=20;20.5/5" | bc
echo "20.5%5" | bc
num=4
echo "scale=2;sqrt($num)" | bc -l
echo "scale=2;3^3" | bc -l
Ln 17, Col 1
Spaces: 4
UTF-8
LF
Shell Script (Bash)

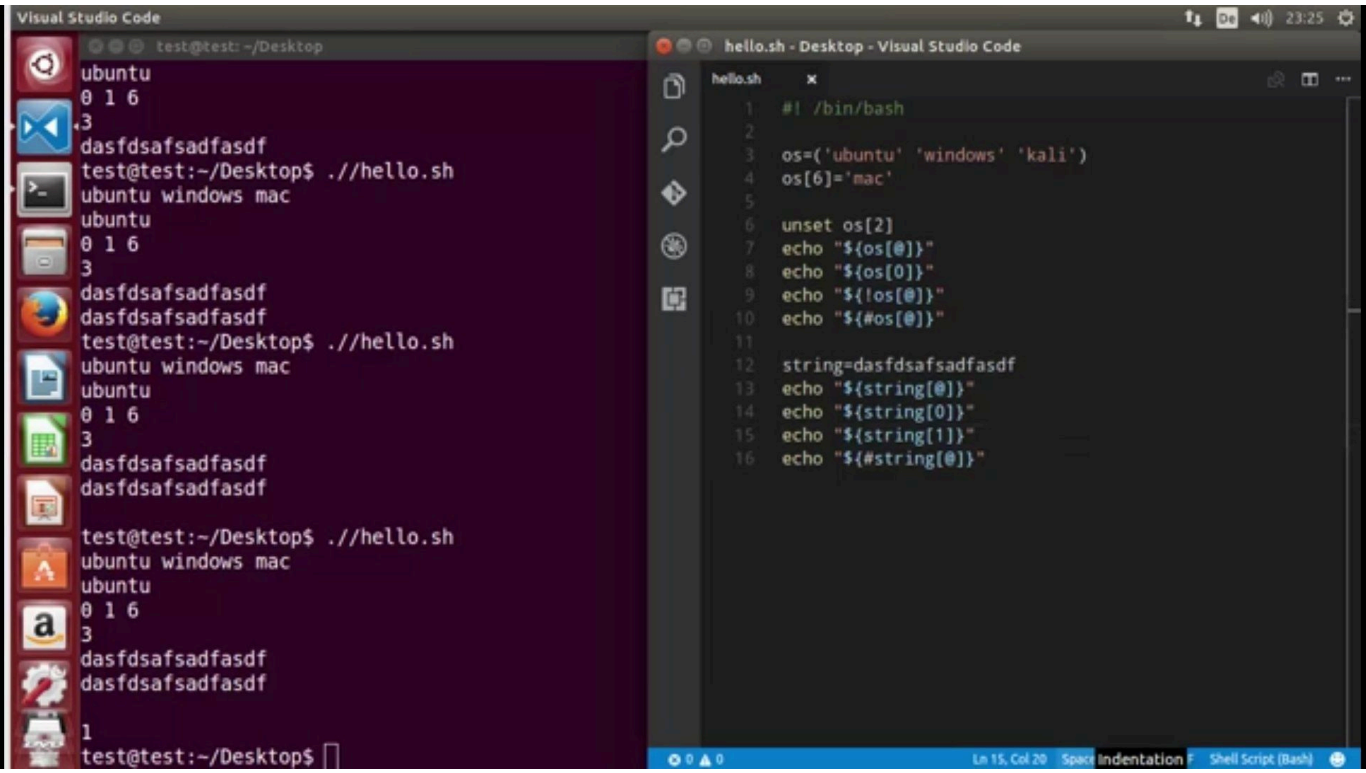

Visual Studio Code
21:24
test@test: ~
test@test:~$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
case expression in
pattern1)
statements ;;
pattern2)
statements ;;
...
esac
Ln 5, Col 9 (13 selected)
Spaces: 4
UTF-8
LF
Shell Script (Bash)

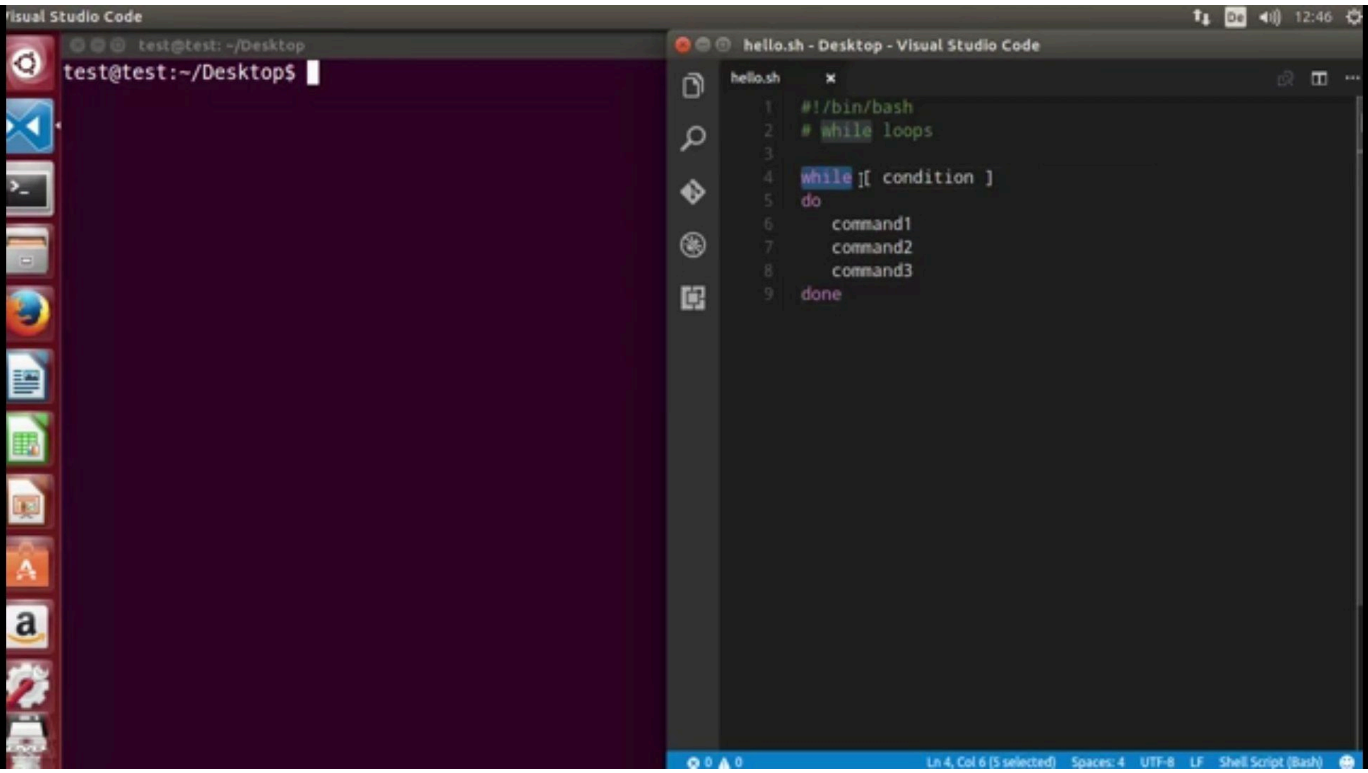

Visual Studio Code
21:34
test@test: ~/Desktop
test@test:~$ cd Desktop/
test@test:~/Desktop$ ./hello.sh
Unknown vehicle
test@test:~/Desktop$ ./hello.sh car
Rent of car is 100 dollar
test@test:~/Desktop$ ./hello.sh van
Rent of van is 80 dollar
test@test:~/Desktop$ ./hello.sh truck
Rent of truck is 150 dollar
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
vehicle=$1
case $vehicle in
"car")
echo "Rent of $vehicle is 100 dollar" ;;
"van")
echo "Rent of $vehicle is 80 dollar" ;;
"bicycle")
echo "Rent of $vehicle is 5 dollar" ;;
"truck")
echo "Rent of $vehicle is 150 dollar" ;;
*)
echo "Unknown vehicle" ;;
esac
Ln 19, Col 1
Spaces: 4
UTF-8
LF
Shell Script (Bash)

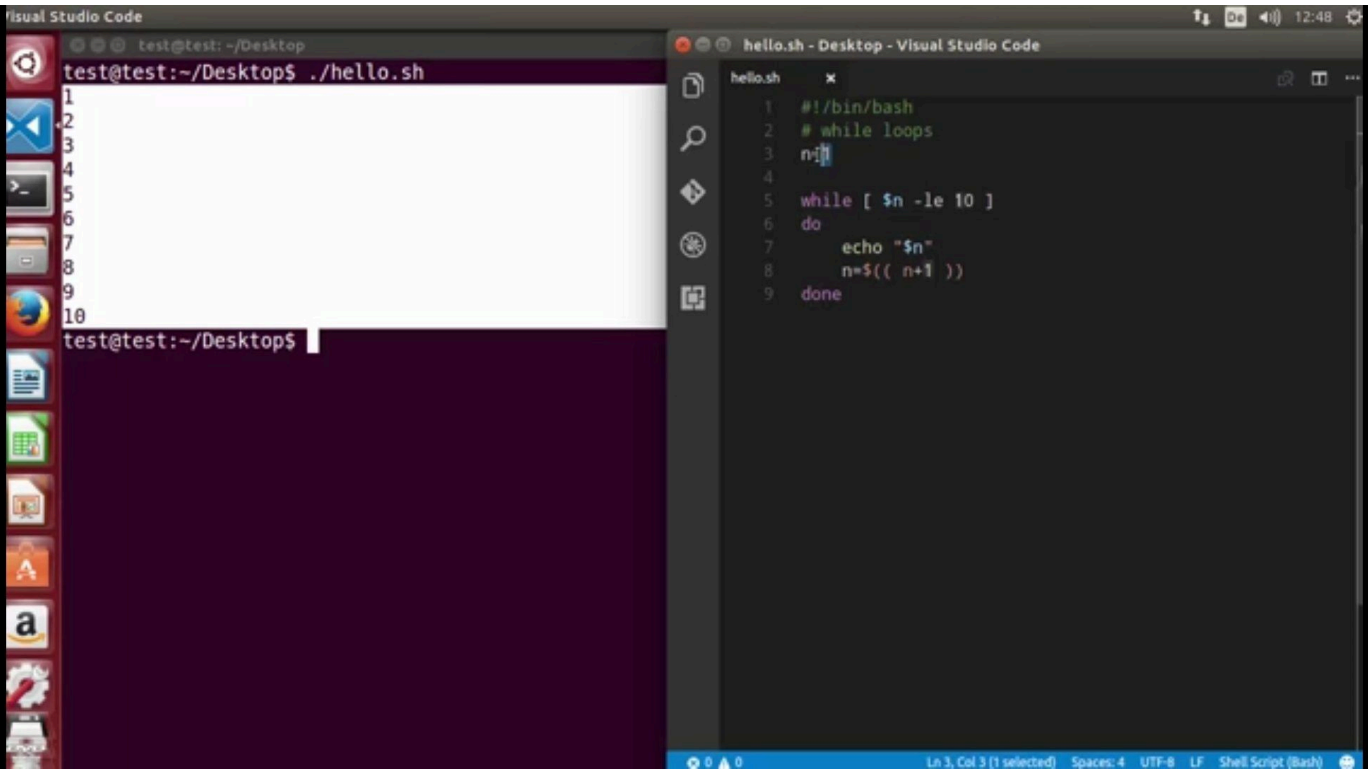

Terminal
22:01
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
Enter some character : f
User entered f a to z
test@test:~/Desktop$ ./hello.sh
Enter some character : K
User entered K a to z
test@test:~/Desktop$ LANG=C
test@test:~/Desktop$ ./hello.sh
Enter some character : K
User entered K A to Z
test@test:~/Desktop$ ./hello.sh
Enter some character : 9
User entered 9 0 to 9
test@test:~/Desktop$ ./hello.sh
Enter some character : 5
User entered 5 0 to 9
test@test:~/Desktop$ ./hello.sh
Enter some character : &
User entered & special character
test@test:~/Desktop$ ./hello.sh
Enter some character : sdsdsdsd
Unknown input
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
echo -e "Enter some character : \c"
read value
case $value in
[a-z])
echo "User entered $value a to z" ;;
[A-Z])
echo "User entered $value A to Z" ;;
[0-9])
echo "User entered $value 0 to 9" ;;
?)
echo "User entered $value special character" ;;
*)
echo "Unknown input" ;;
esac

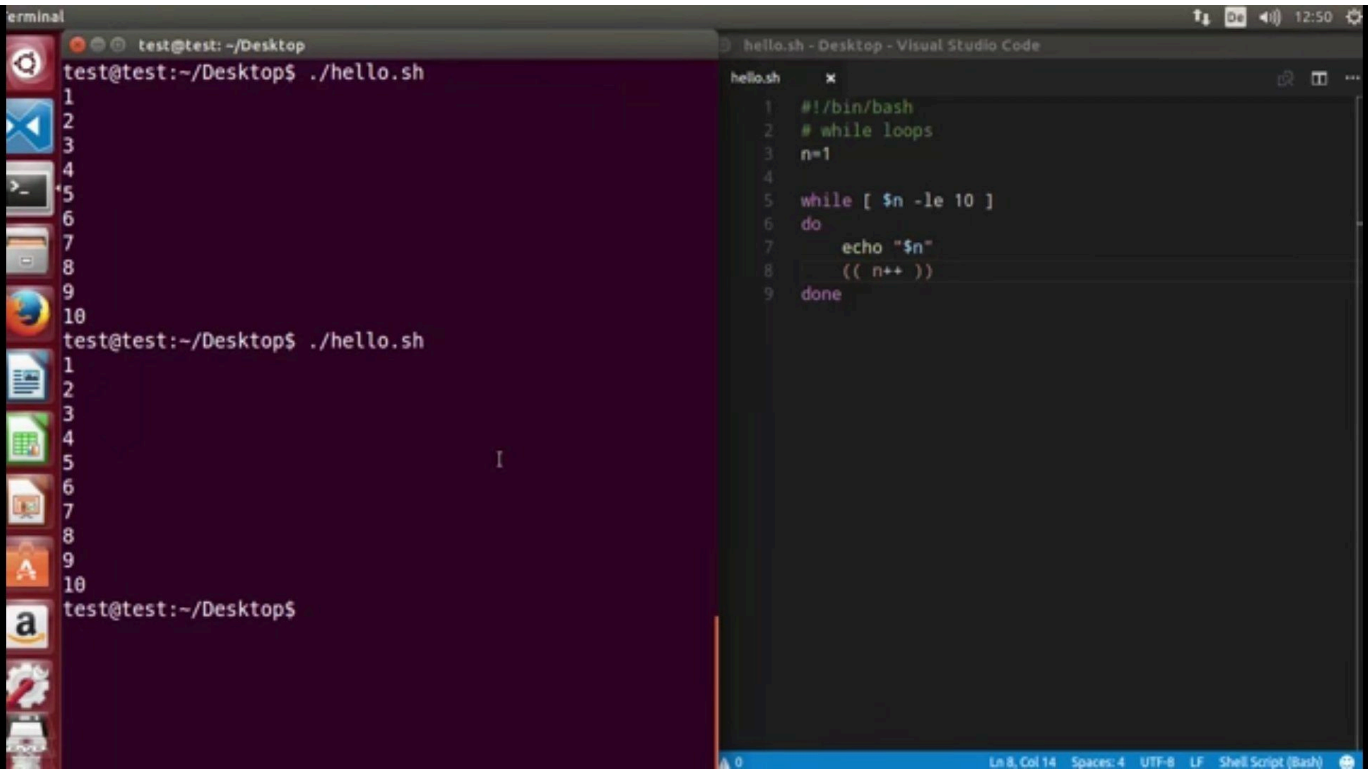

Terminal
23:17
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
ubuntu windows kali
test@test:~/Desktop$ ./hello.sh
ubuntu windows kali
windows
test@test:~/Desktop$ ./hello.sh
ubuntu windows kali
ubuntu
test@test:~/Desktop$ ./hello.sh
ubuntu windows kali
ubuntu
0 1 2
test@test:~/Desktop$ ./hello.sh
ubuntu windows kali
ubuntu
0 1 2
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
os=('ubuntu' 'windows' 'kali')
echo "${os[@]}"
echo "${os[0]}"
echo "${!os[@]}"
echo "${#os[@]}"
Ln 10, Col 1
Spaces: 4
UTF-8
LF
Shell Script (Bash)

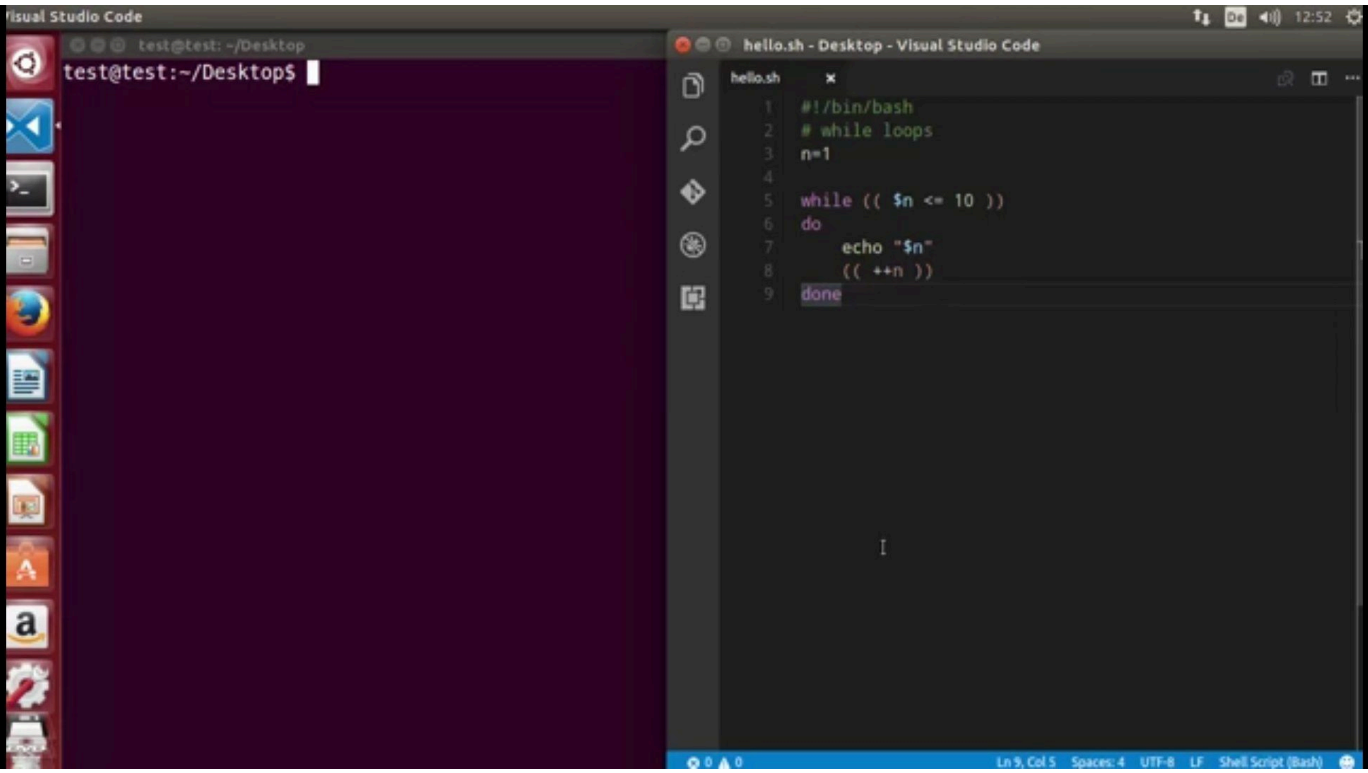

Terminal
23:20
test@test: ~/Desktop
windows
test@test:~/Desktop$ ./hello.sh
ubuntu windows kali
ubuntu
test@test:~/Desktop$ ./hello.sh
ubuntu windows kali
ubuntu
0 1 2
test@test:~/Desktop$ ./hello.sh
ubuntu windows kali
ubuntu
0 1 2
3
test@test:~/Desktop$ ./hello.sh
ubuntu windows kali mac
ubuntu
0 1 2 3
4
test@test:~/Desktop$ ./hello.sh
mac windows kali
mac
0 1 2
3
test@test:~/Desktop$ ./hello.sh
ubuntu windows mac
ubuntu
0 1 3
3
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
os=('ubuntu' 'windows' 'kali')
os[3]='mac'
unset os[2]
echo "${os[@]}"
echo "${os[0]}"
echo "${!os[@]}"
echo "${#os[@]}"
Ln 7, Col 16
Spaces: 4
UTF-8
LF
Shell Script (Bash)

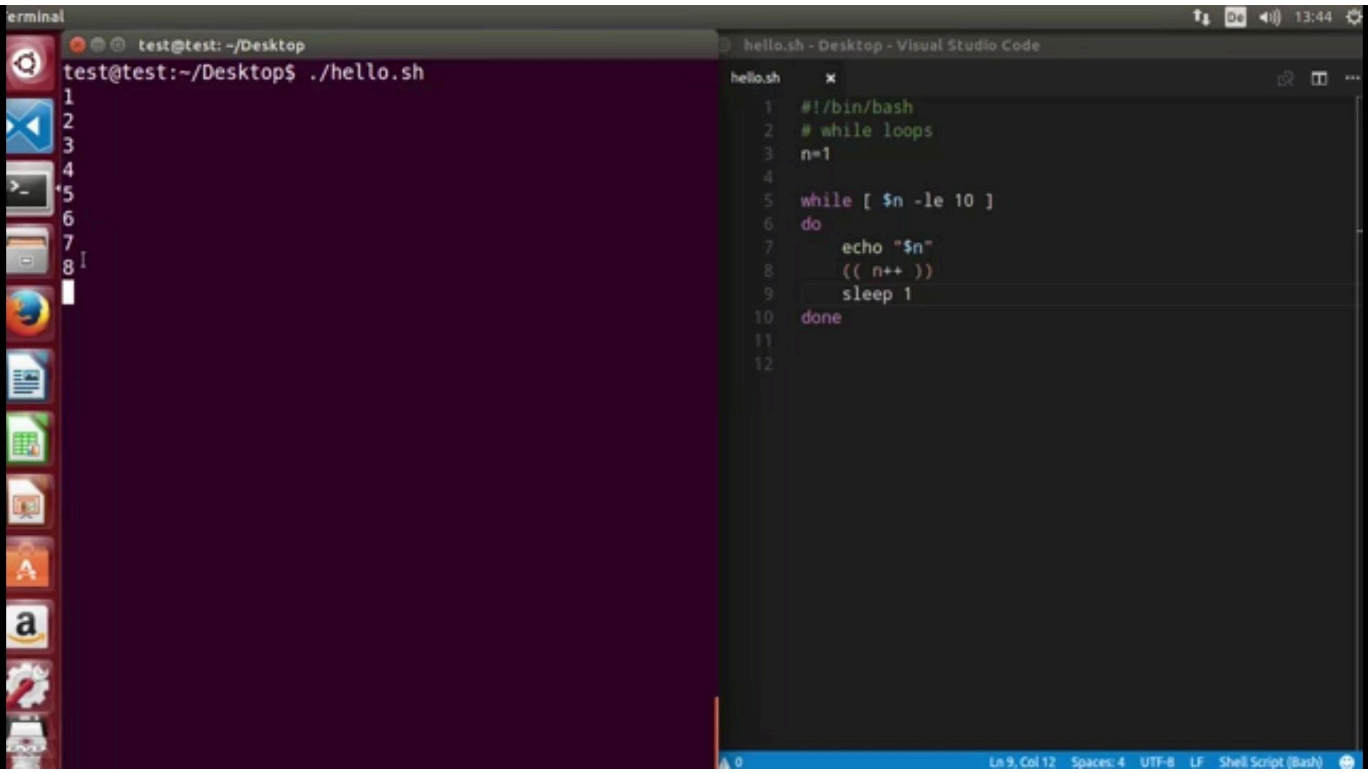

Visual Studio Code
23:25
test@test: ~/Desktop
ubuntu
0 1 6
3
dasfdsafsadfasdf
test@test:~/Desktop$ .//hello.sh
ubuntu windows mac
ubuntu
0 1 6
3
dasfdsafsadfasdf
dasfdsafsadfasdf
test@test:~/Desktop$ .//hello.sh
ubuntu windows mac
ubuntu
0 1 6
3
dasfdsafsadfasdf
dasfdsafsadfasdf
test@test:~/Desktop$ .//hello.sh
ubuntu windows mac
ubuntu
0 1 6
3
dasfdsafsadfasdf
dasfdsafsadfasdf
1
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#! /bin/bash
os=('ubuntu' 'windows' 'kali')
os[6]='mac'
unset os[2]
echo "${os[@]}"
echo "${os[0]}"
echo "${!os[@]}"
echo "${#os[@]}"
string=dasfdsafsadfasdf
echo "${string[@]}"
echo "${string[0]}"
echo "${string[1]}"
echo "${#string[@]}"
Ln 15, Col 20
Indentation
Shell Script (Bash)

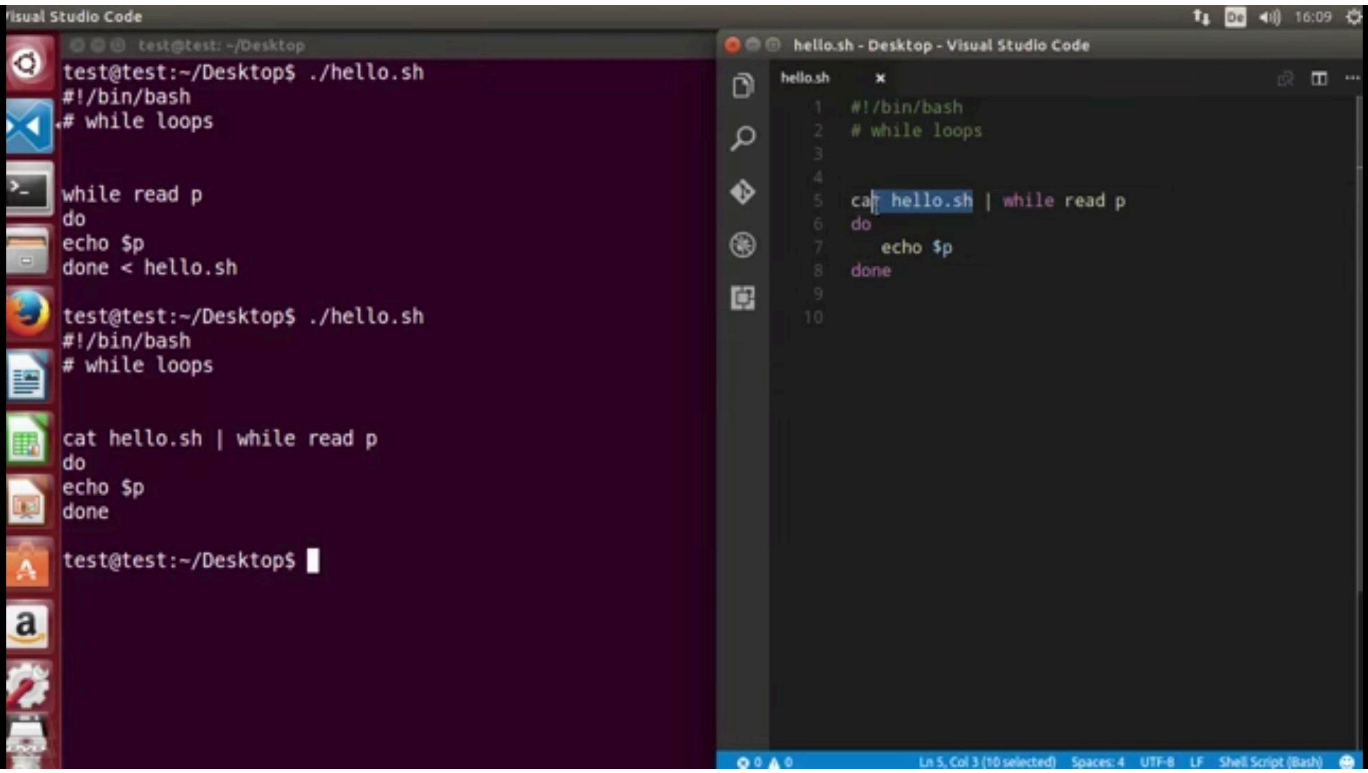

Visual Studio Code
12:46
test@test: ~/Desktop
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
while loops
while [condition]
do
command1
command2
command3
done
Ln 4, Col 6 (5 selected)
Spaces: 4
UTF-8
LF
Shell Script (Bash)

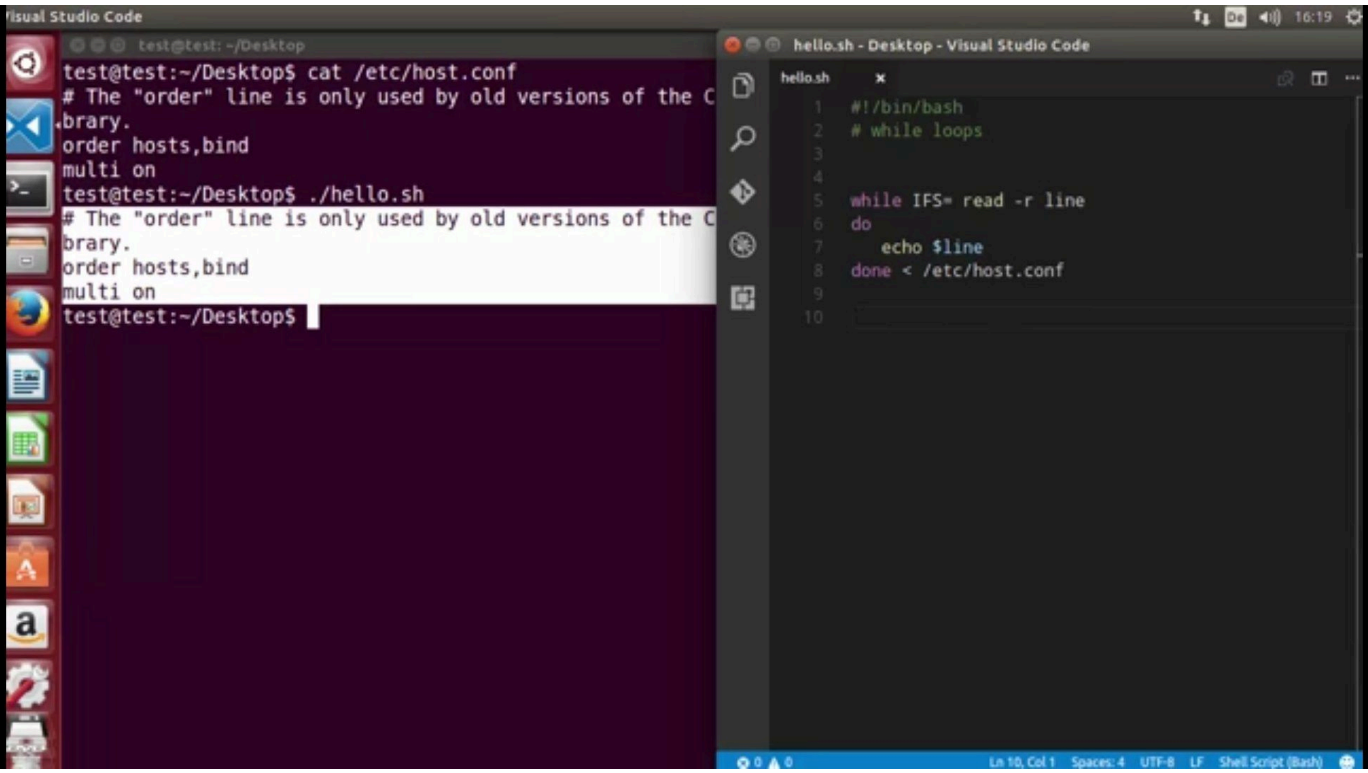

Visual Studio Code
12:48
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
10
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
# while loops
n=1
while [$n -le 10]
do
echo "$n"
n=$((n+1))
done
Ln 3, Col 3 (1 selected) Spaces: 4 UTF-8 LF Shell Script (Bash)

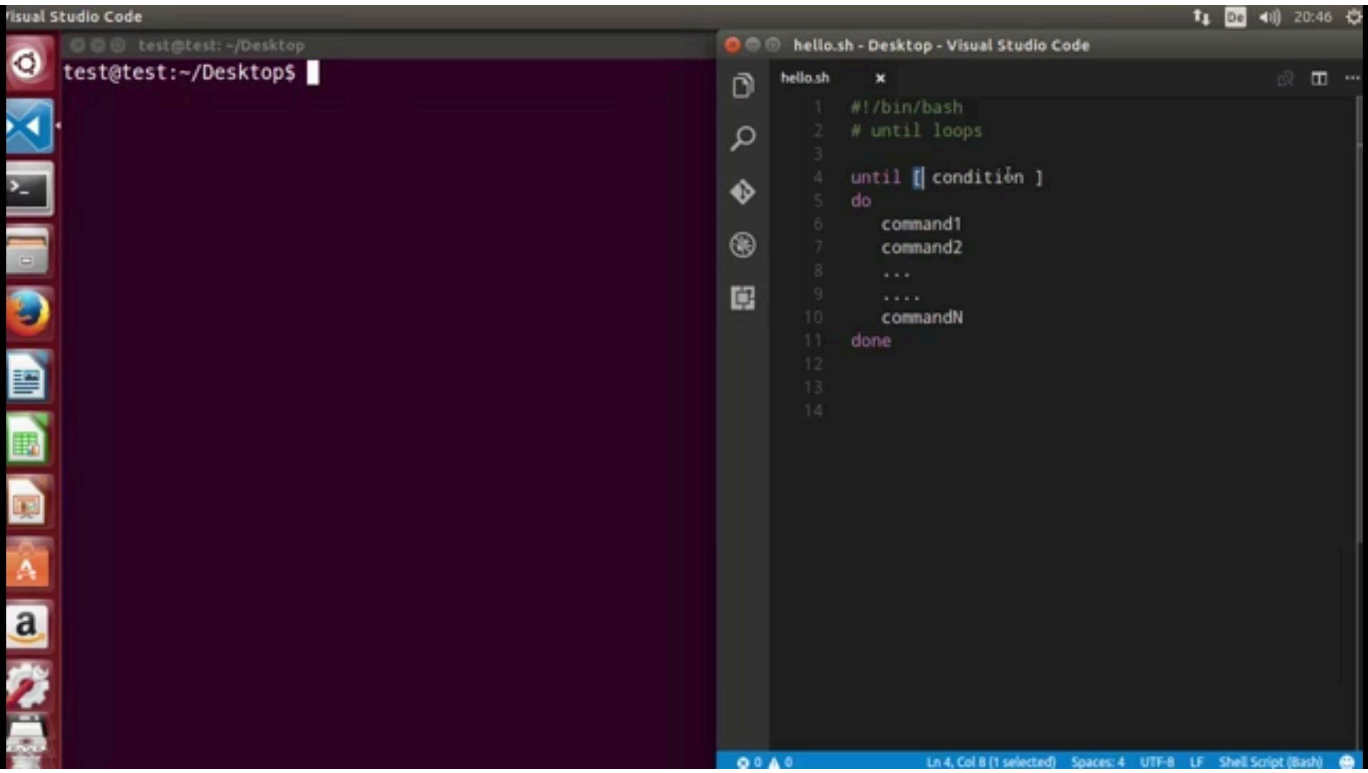

test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
10
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
10
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
# while loops
n=1
while [$n -le 10]
do
echo "$n"
((n++))
done
Ln 8, Col 14
Spaces: 4
UTF-8
LF
Shell Script (Bash)

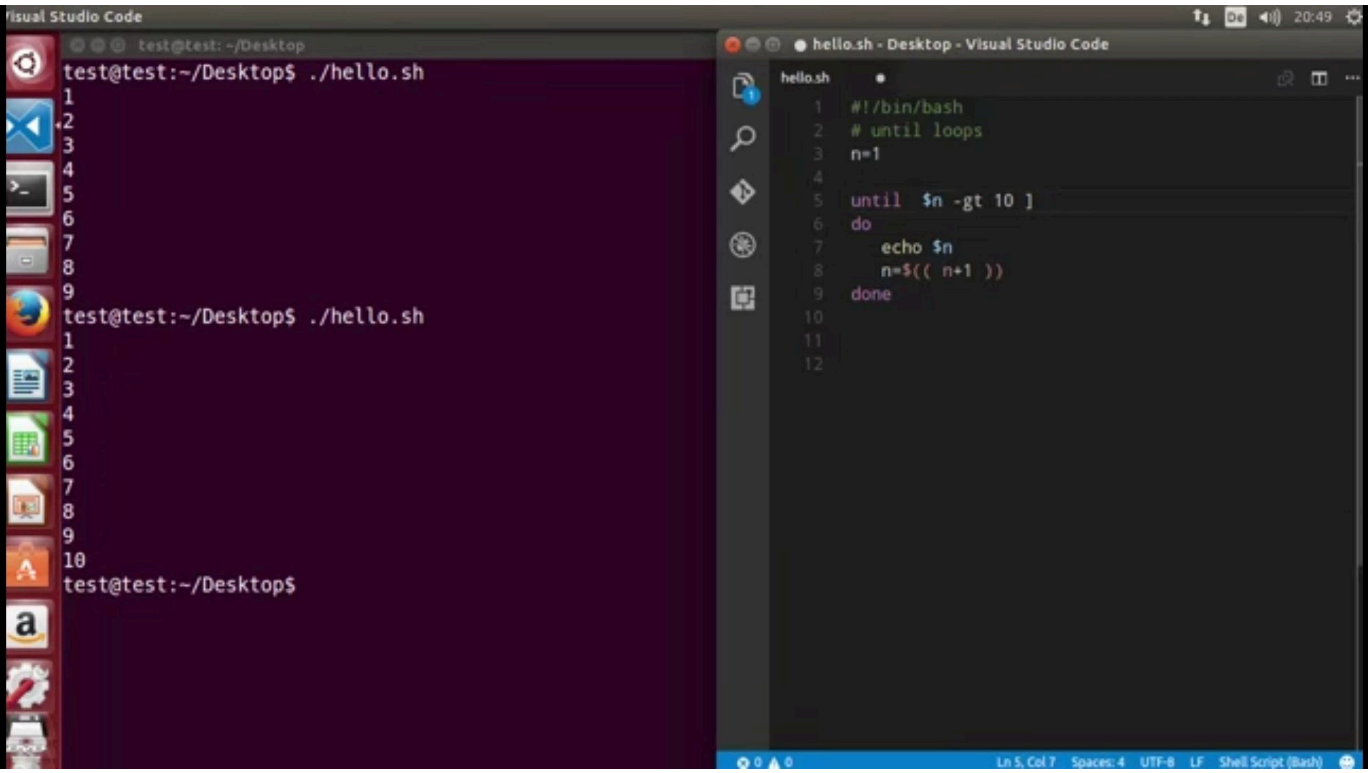

Visual Studio Code
12:52
test@test: ~/Desktop
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
# while loops
n=1
while (($n <= 10))
do
echo "$n"
((++n))
done
Ln 9, Col 5
Spaces: 4
UTF-8
LF
Shell Script (Bash)

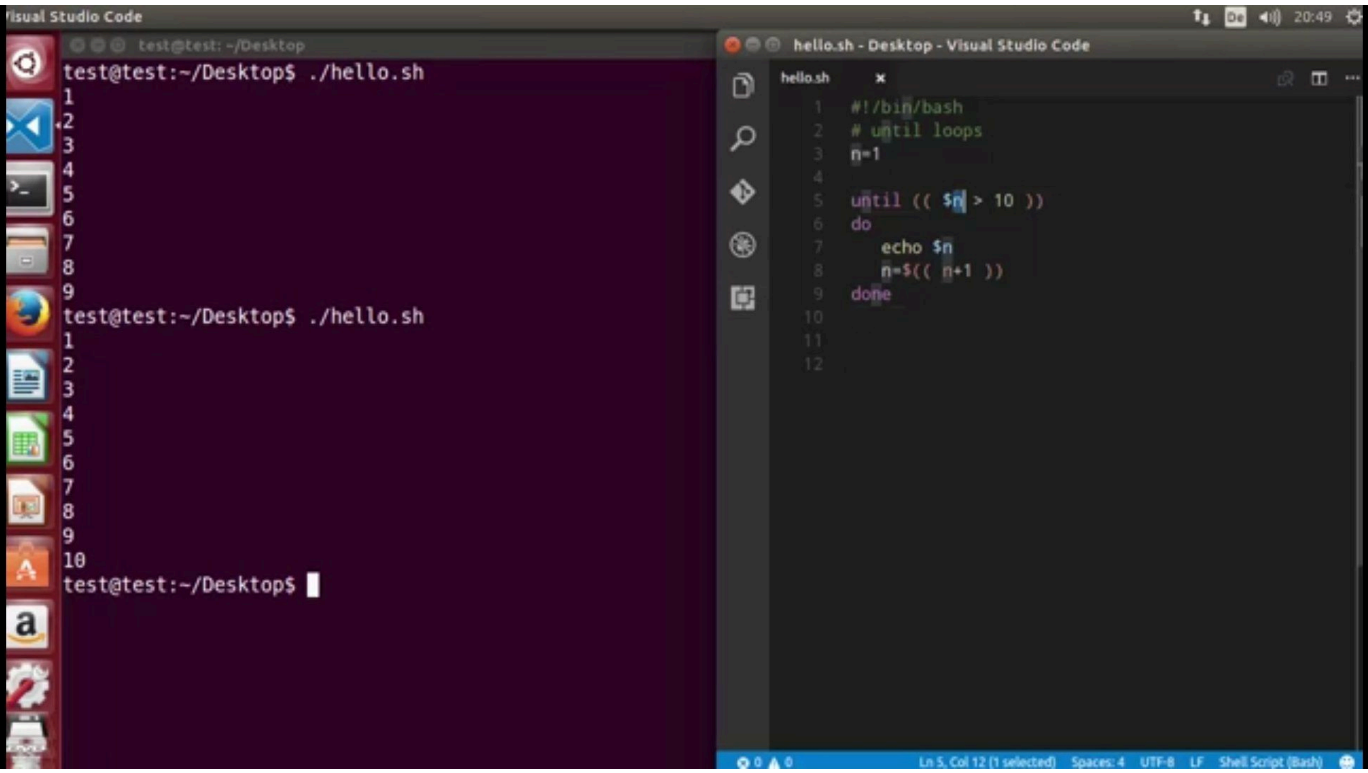

erminal
13:44
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
# while loops
n=1
while [$n -le 10]
do
echo "$n"
((n++))
sleep 1
done
Ln 9, Col 12
Spaces: 4
UTF-8
LF
Shell Script (Bash)

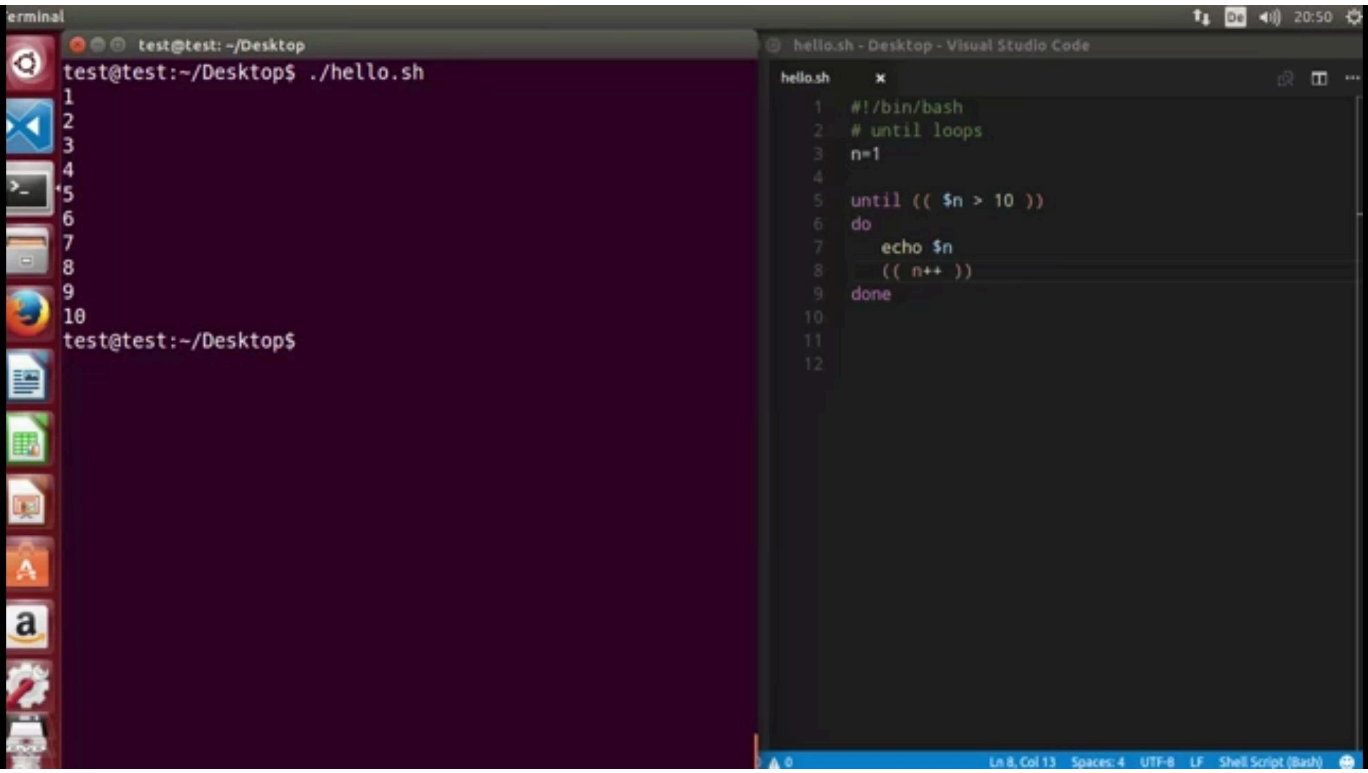

Visual Studio Code
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
#!/bin/bash
while loops
while read p
do
echo $p
done < hello.sh
test@test:~/Desktop$ ./hello.sh
#!/bin/bash
while loops
cat hello.sh | while read p
do
echo $p
done
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
while loops
cat hello.sh | while read p
do
echo $p
done
Ln 5, Col 3 (10 selected) Spaces: 4 UTF-8 LF Shell Script (Bash)

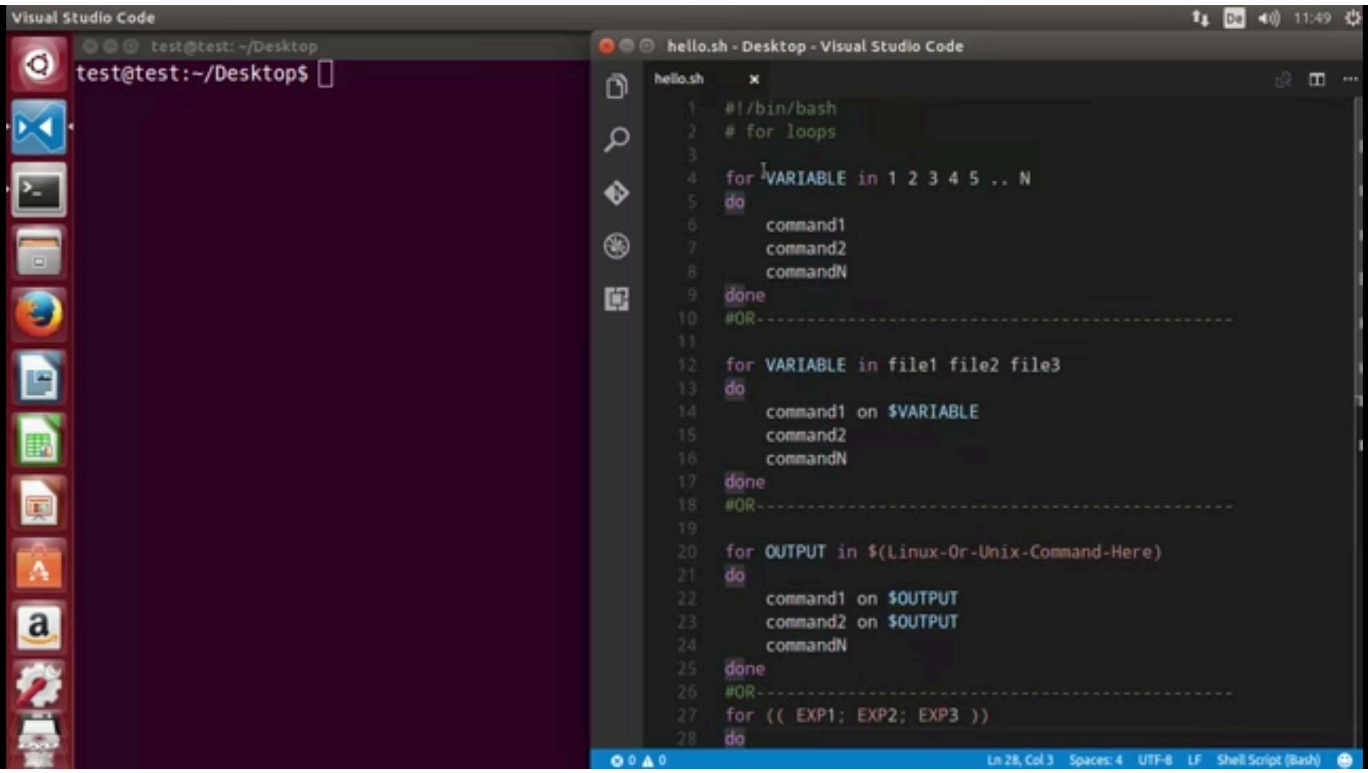

Visual Studio Code
test@test: ~/Desktop
test@test:~/Desktop$ cat /etc/host.conf
The "order" line is only used by old versions of the C
brary.
order hosts,bind
multi on
test@test:~/Desktop$ ./hello.sh
The "order" line is only used by old versions of the C
brary.
order hosts,bind
multi on
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
while loops
while IFS= read -r line
do
echo $line
done < /etc/host.conf
Ln 10, Col 1
Spaces: 4
UTF-8
LF
Shell Script (Bash)

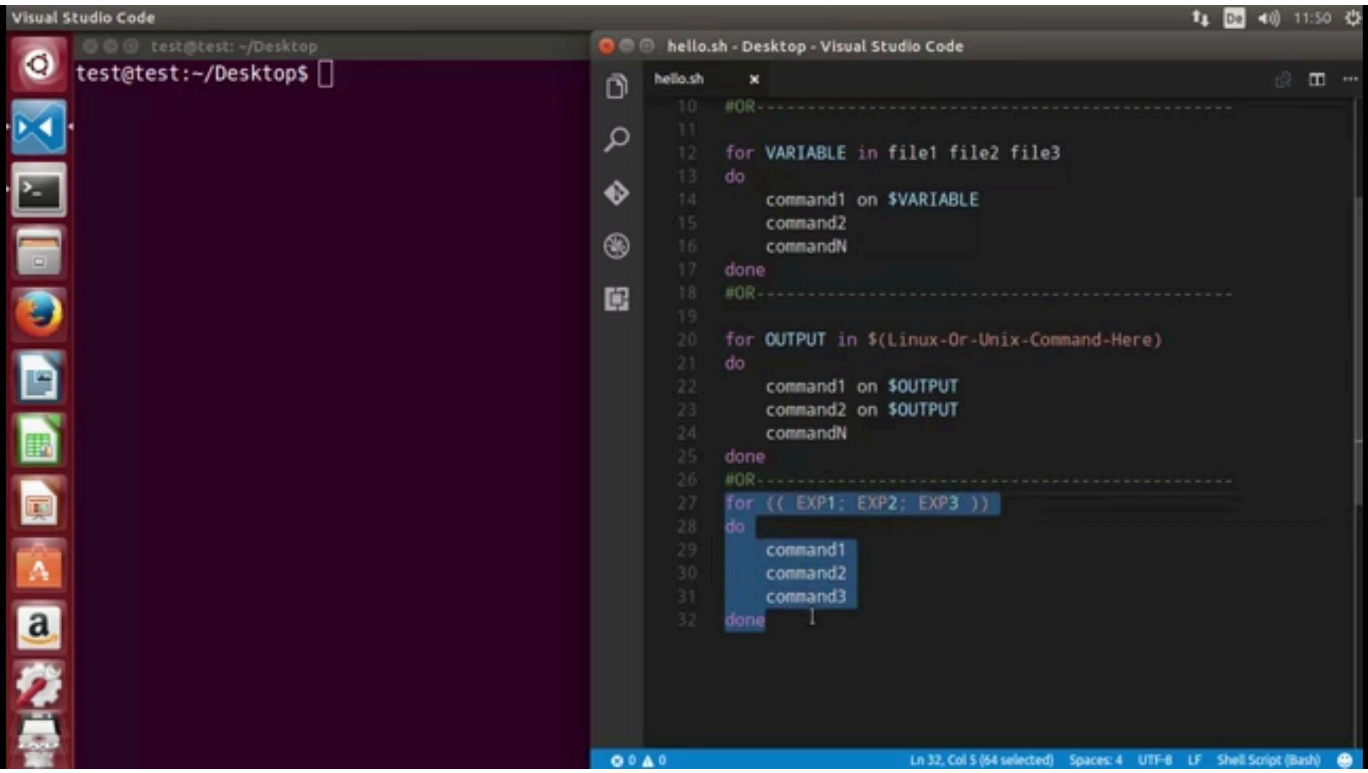

Visual Studio Code
test@test: ~/Desktop
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
until loops
until [condition]
do
command1
command2
...
....
commandN
done
Ln 4, Col 8 (1 selected)
Spaces: 4
UTF-8
LF
Shell Script (Bash)

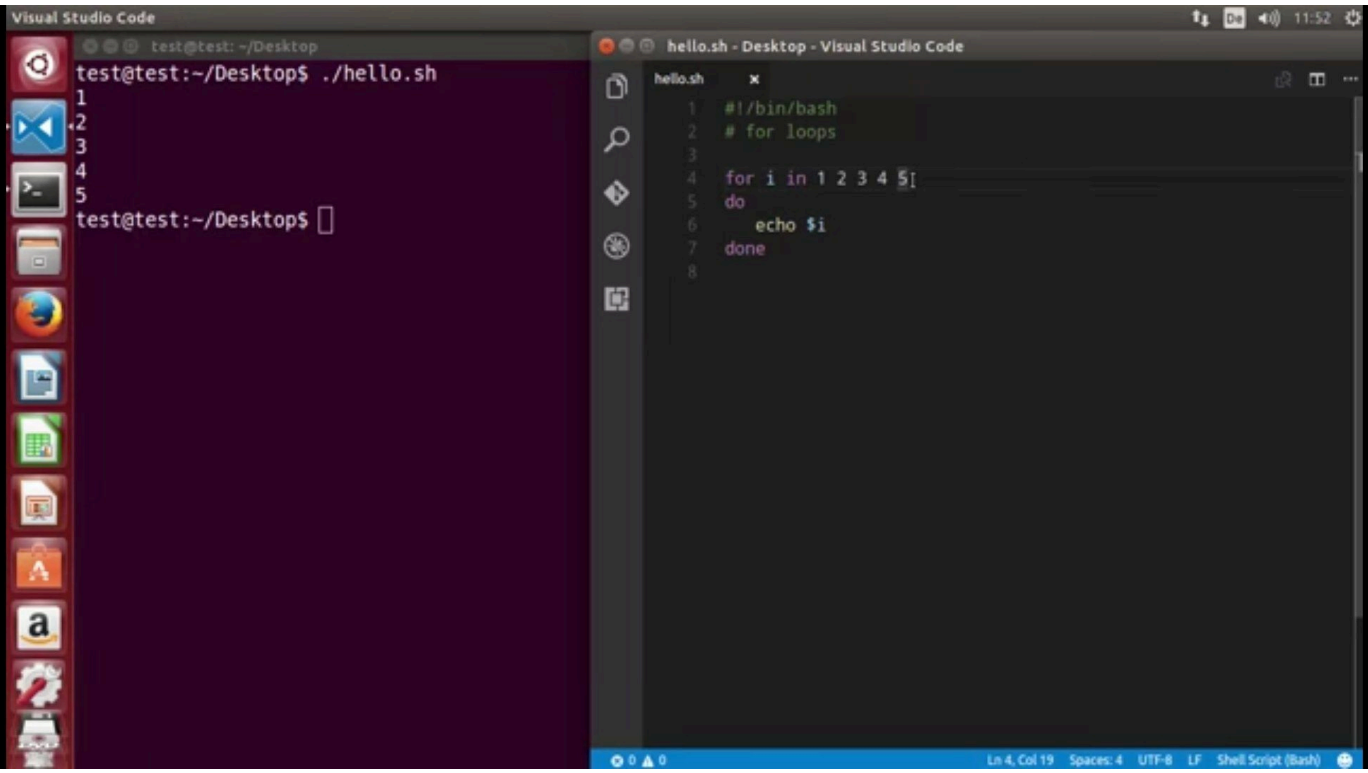

Visual Studio Code
20:49
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
10
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
# until loops
n=1
until $n -gt 10]
do
echo $n
n=$((n+1))
done
Ln 5, Col 7
Spaces: 4
UTF-8
LF
Shell Script (Bash)

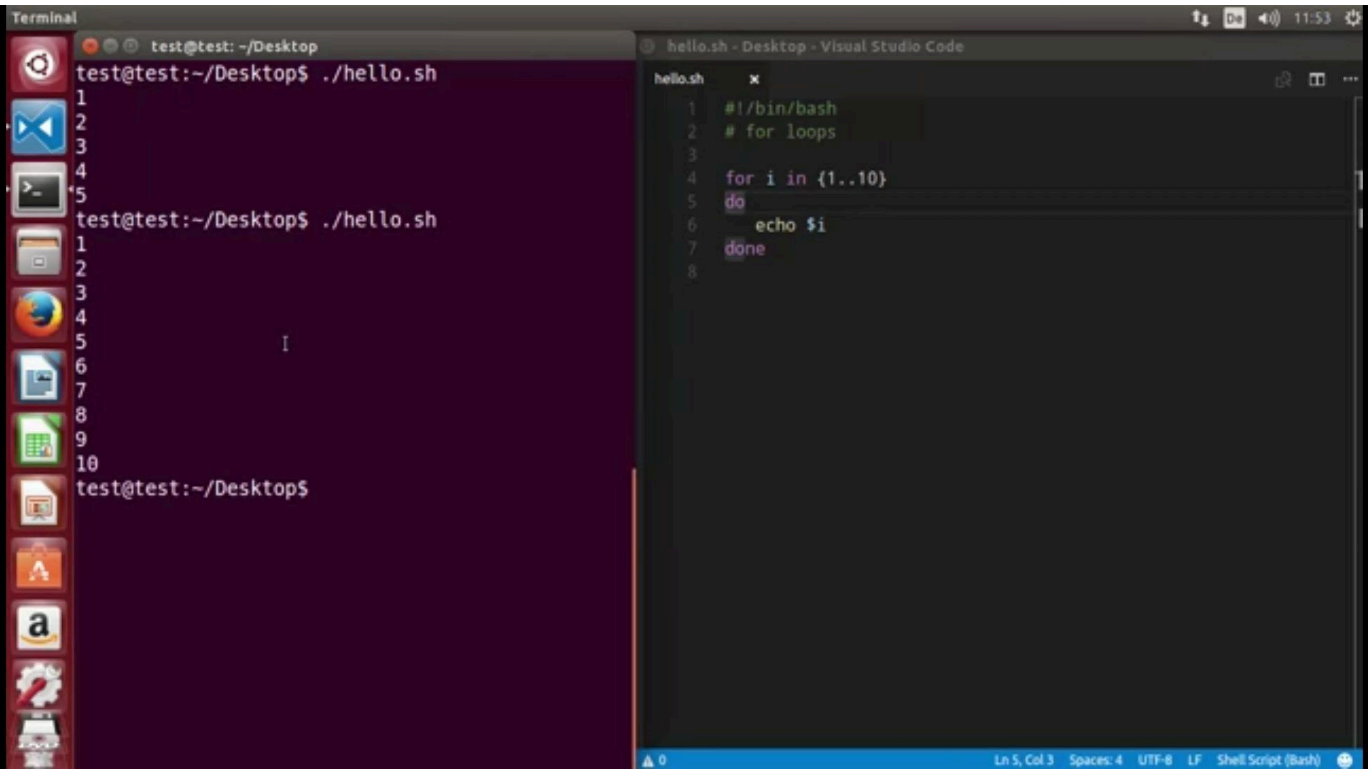

Visual Studio Code
20:49
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
10
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
# until loops
n=1
until (($n > 10))
do
echo $n
n=$((n+1))
done
Ln 5, Col 12 (1 selected) Spaces: 4 UTF-8 LF Shell Script (Bash)

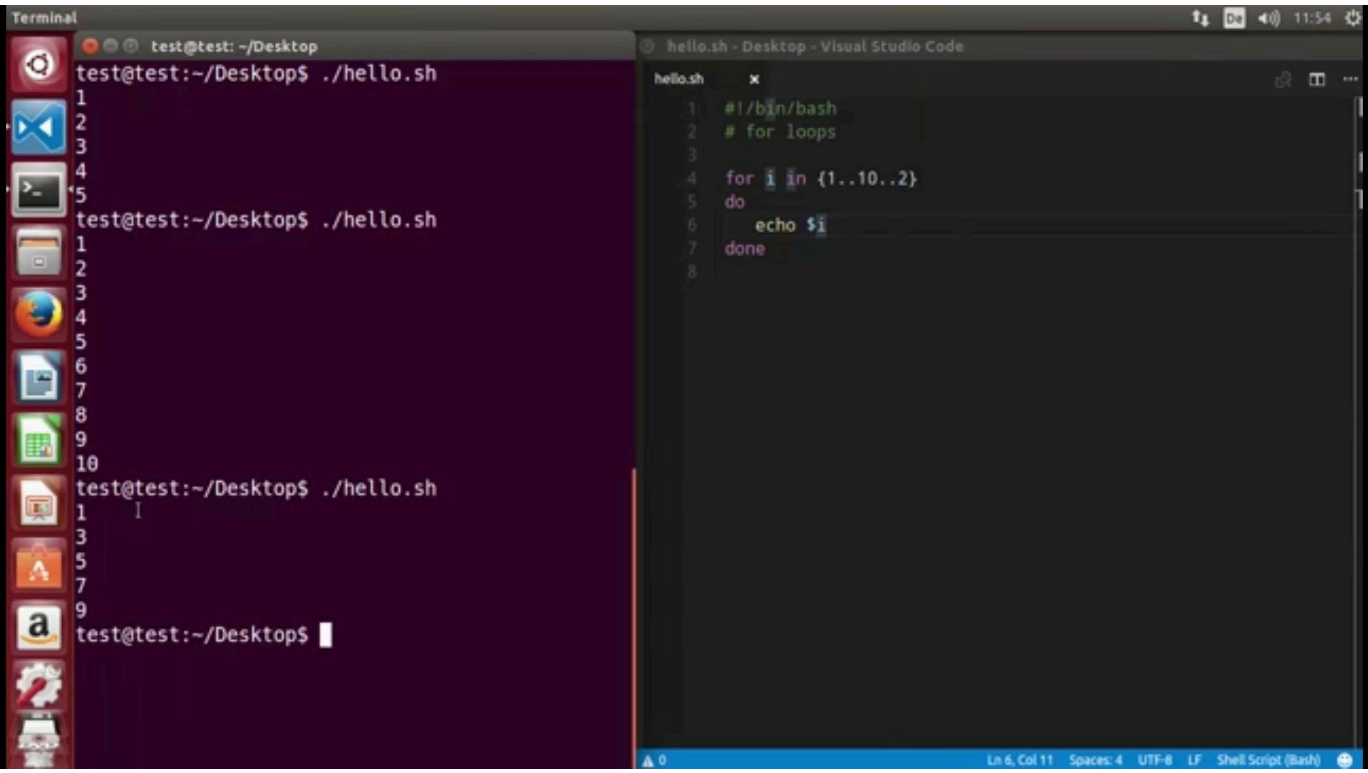

erminal
20:50
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
10
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
# until loops
n=1
until (($n > 10))
do
echo $n
((n++))
done
Ln 8, Col 13
Spaces: 4
UTF-8
LF
Shell Script (Bash)

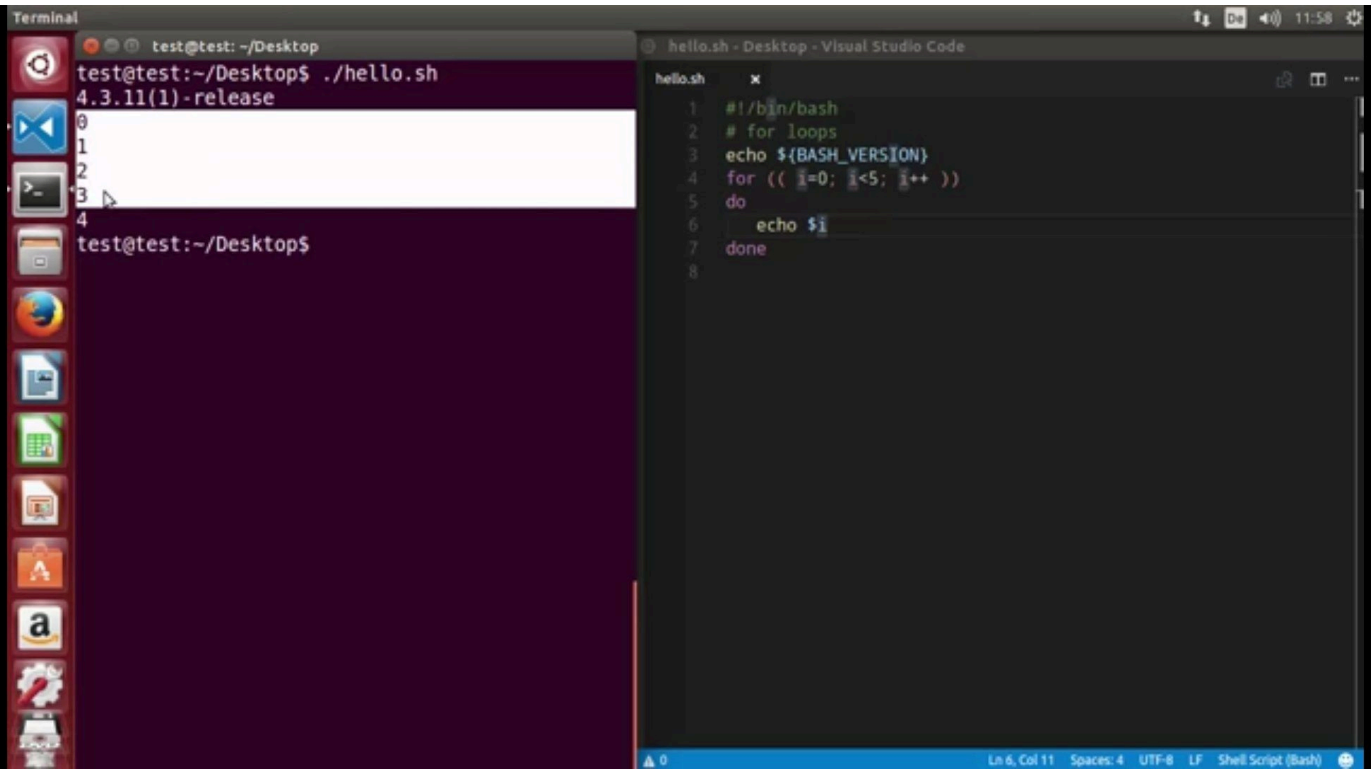

Visual Studio Code
11:49
test@test: ~/Desktop
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
for loops
for VARIABLE in 1 2 3 4 5 .. N
do
command1
command2
commandN
done
#OR----------
for VARIABLE in file1 file2 file3
do
command1 on $VARIABLE
command2
commandN
done
#OR----------
for OUTPUT in $(Linux-Or-Unix-Command-Here)
do
command1 on $OUTPUT
command2 on $OUTPUT
commandN
done
#OR----------
for ((EXP1; EXP2; EXP3))
do
Ln 28, Col 3
Spaces: 4
UTF-8
LF
Shell Script (Bash)

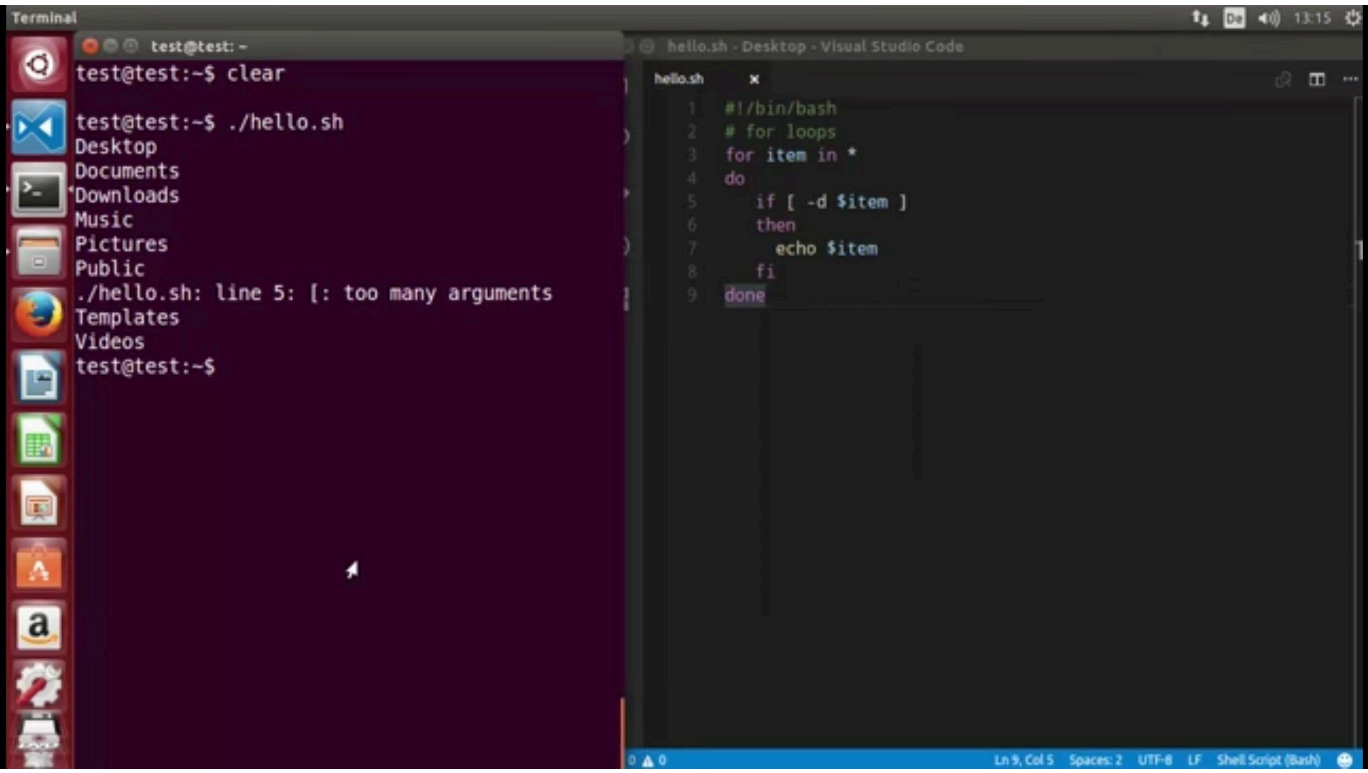

Visual Studio Code
11:50
test@test: ~/Desktop
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#OR----------
for VARIABLE in file1 file2 file3
do
command1 on $VARIABLE
command2
commandN
done
#OR----------
for OUTPUT in $(Linux-Or-Unix-Command-Here)
do
command1 on $OUTPUT
command2 on $OUTPUT
commandN
done
#OR----------
for ((EXP1; EXP2; EXP3))
do
command1
command2
command3
done
Ln 32, Col 5 (64 selected) Spaces: 4 UTF-8 LF Shell Script (Bash)

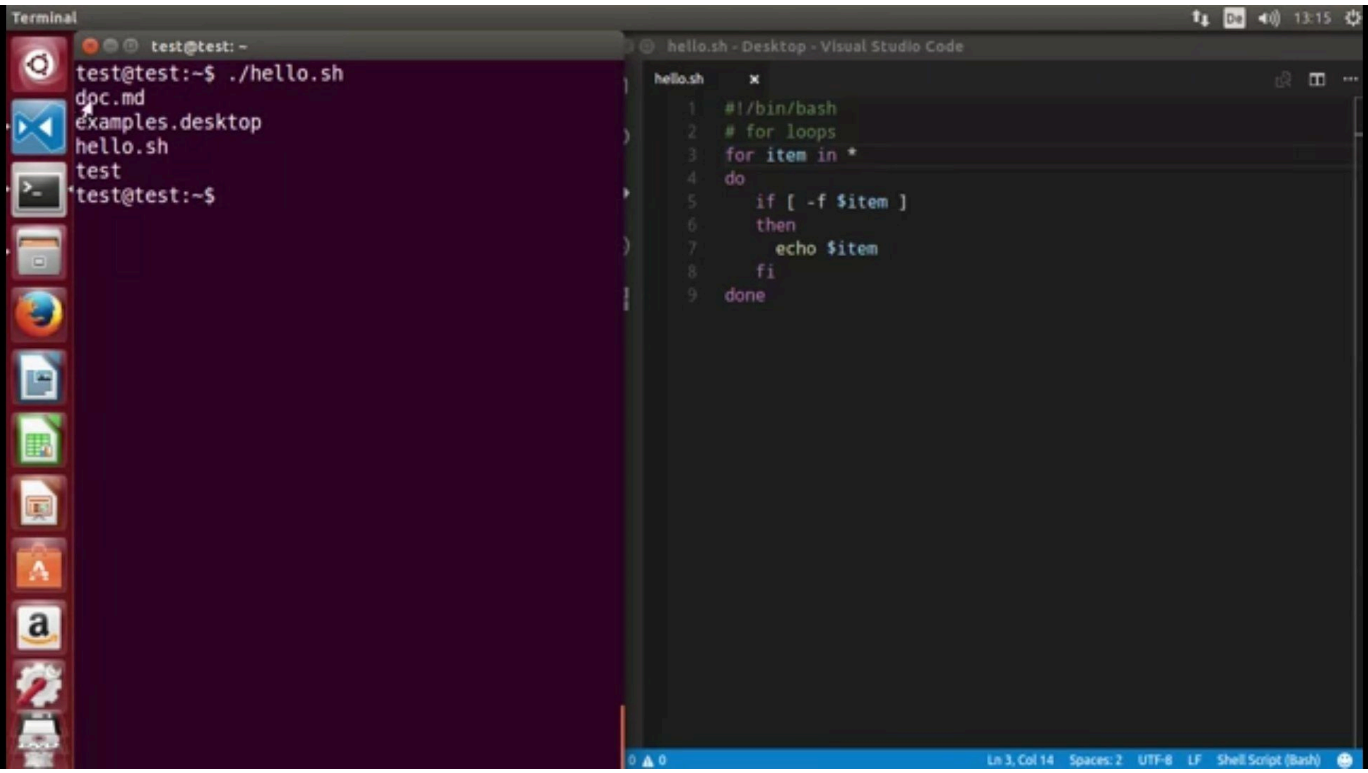

Visual Studio Code
11:52
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
for loops
for i in 1 2 3 4 5
do
echo $i
done
Ln 4, Col 19
Spaces: 4
UTF-8
LF
Shell Script (Bash)

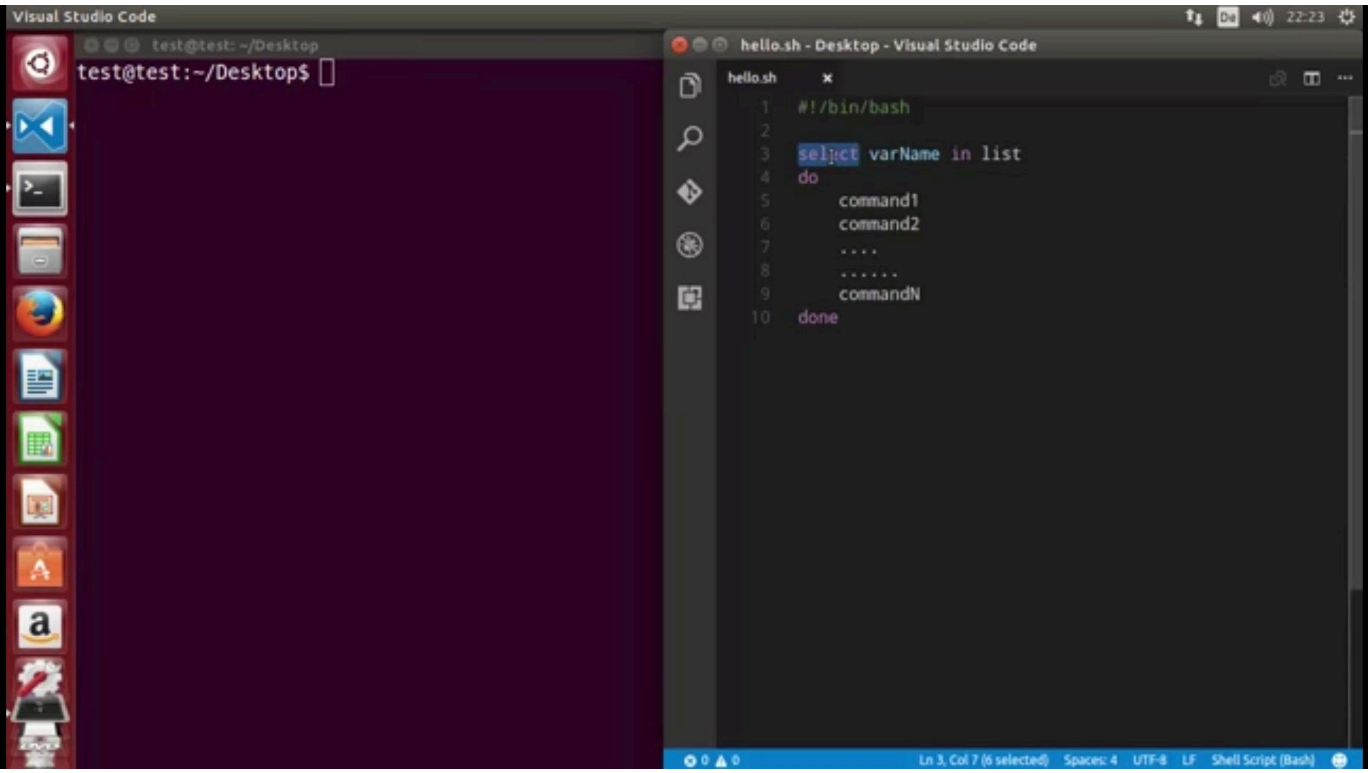

Terminal
11:53
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
10
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
for loops
for i in {1..10}
do
echo $i
done
Ln 5, Col 3
Spaces: 4
UTF-8
LF
Shell Script (Bash)

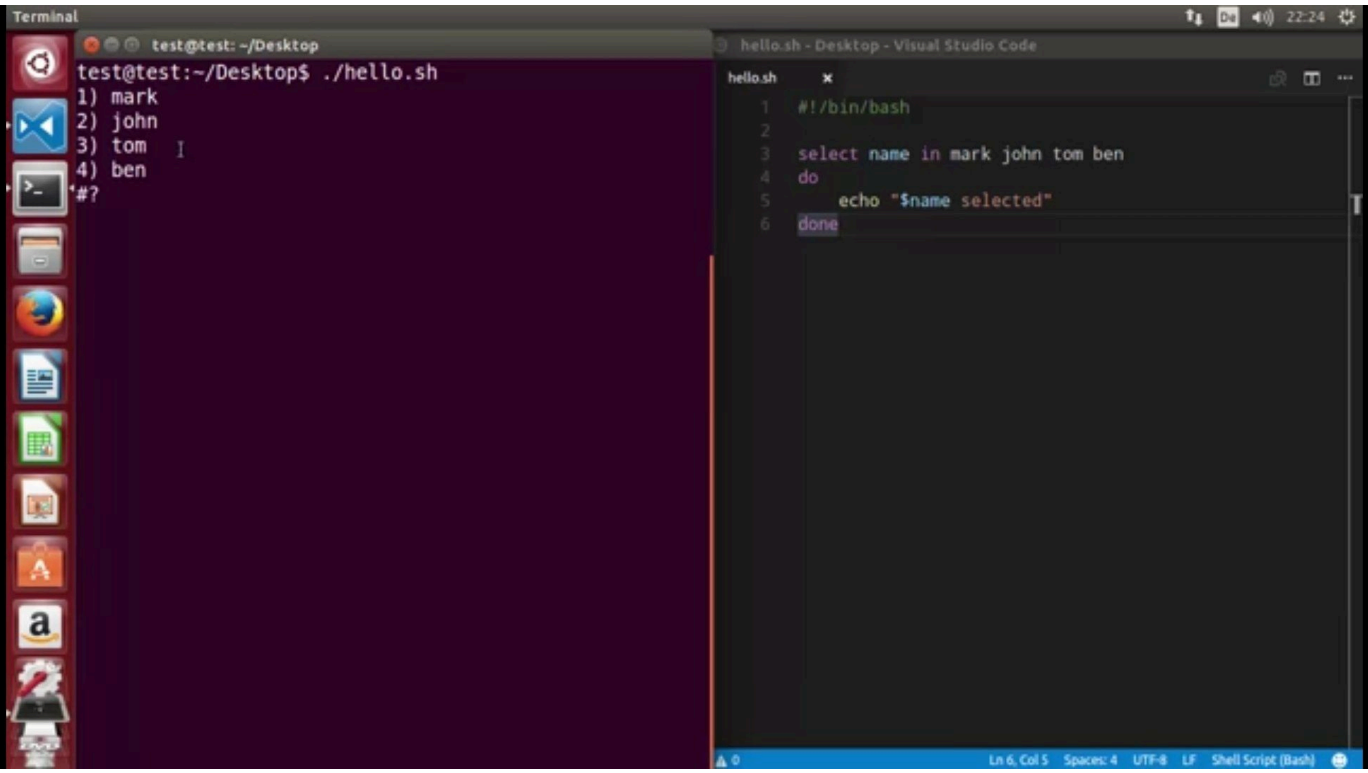

Terminal
11:54
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
6
7
8
9
10
test@test:~/Desktop$ ./hello.sh
1
3
5
7
9
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
for loops
for i in {1..10..2}
do
echo $i
done
Ln 6, Col 11
Spaces: 4
UTF-8
LF
Shell Script (Bash)

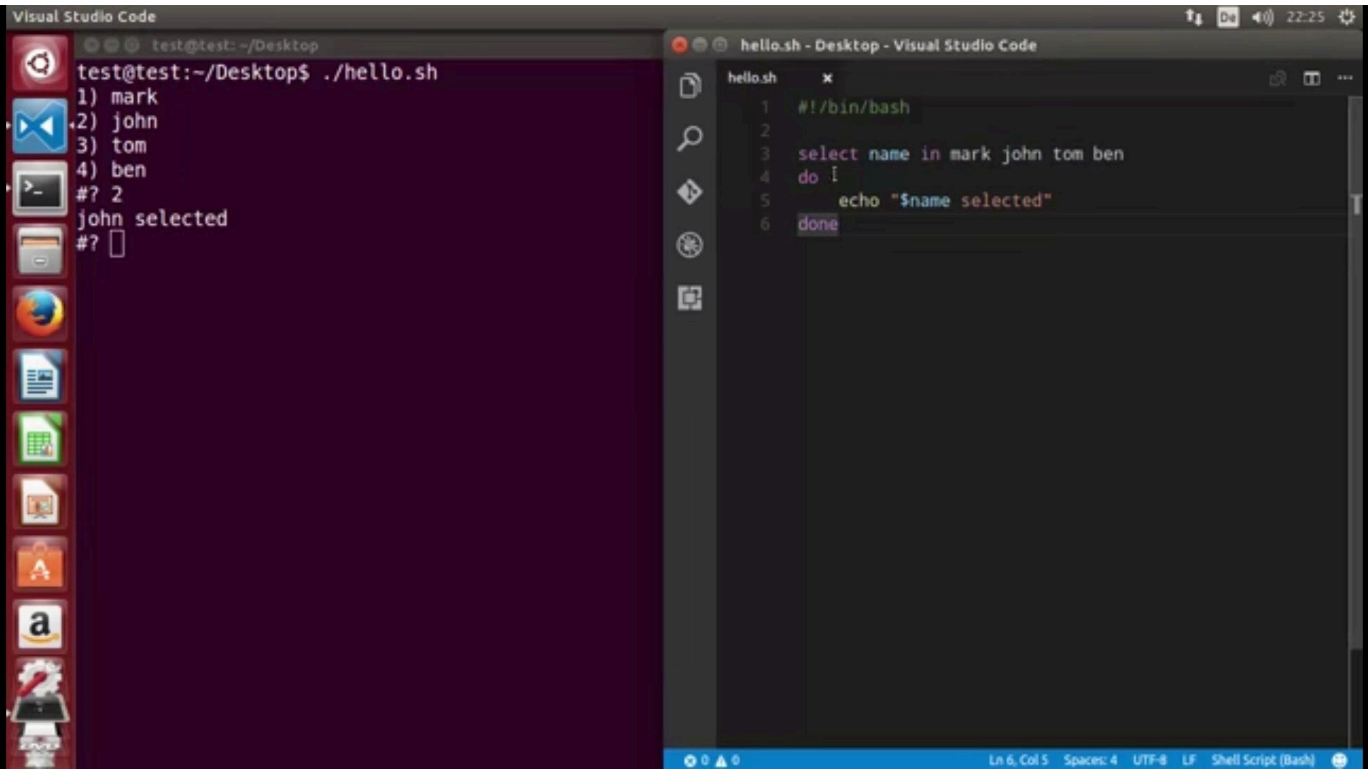

Terminal
11:58
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
4.3.11(1)-release
0
1
2
3
4
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
for loops
echo ${BASH_VERSION}
for ((i=0; i<5; i++))
do
echo $i
done
Ln 6, Col 11
Spaces: 4
UTF-8
LF
Shell Script (Bash)

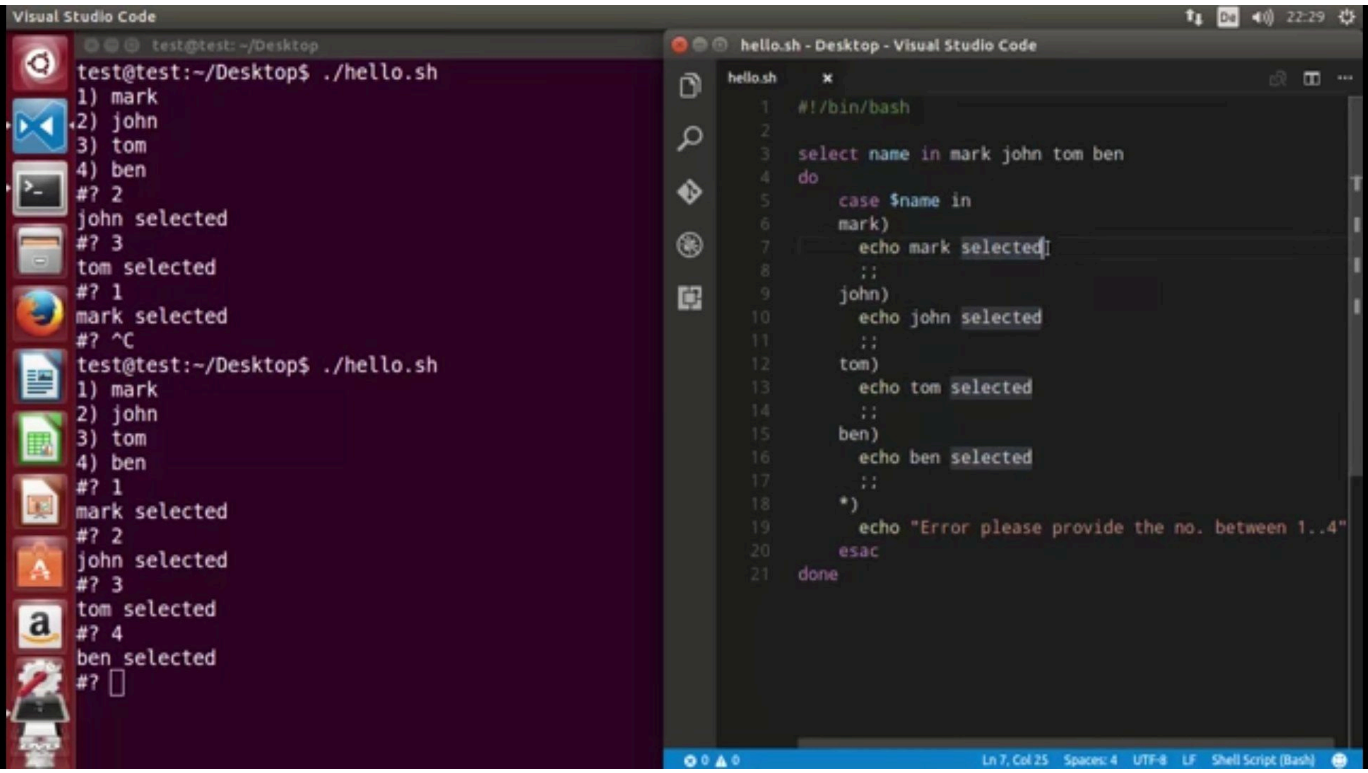

Terminal
13:15
test@test: ~
test@test:~$ clear
test@test:~$ ./hello.sh
Desktop
Documents
Downloads
Music
Pictures
Public
./hello.sh: line 5: [: too many arguments
Templates
Videos
test@test:~$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
for loops
for item in *
do
if [-d $item]
then
echo $item
fi
done
Ln 9, Col 5
Spaces: 2
UTF-8
LF
Shell Script (Bash)

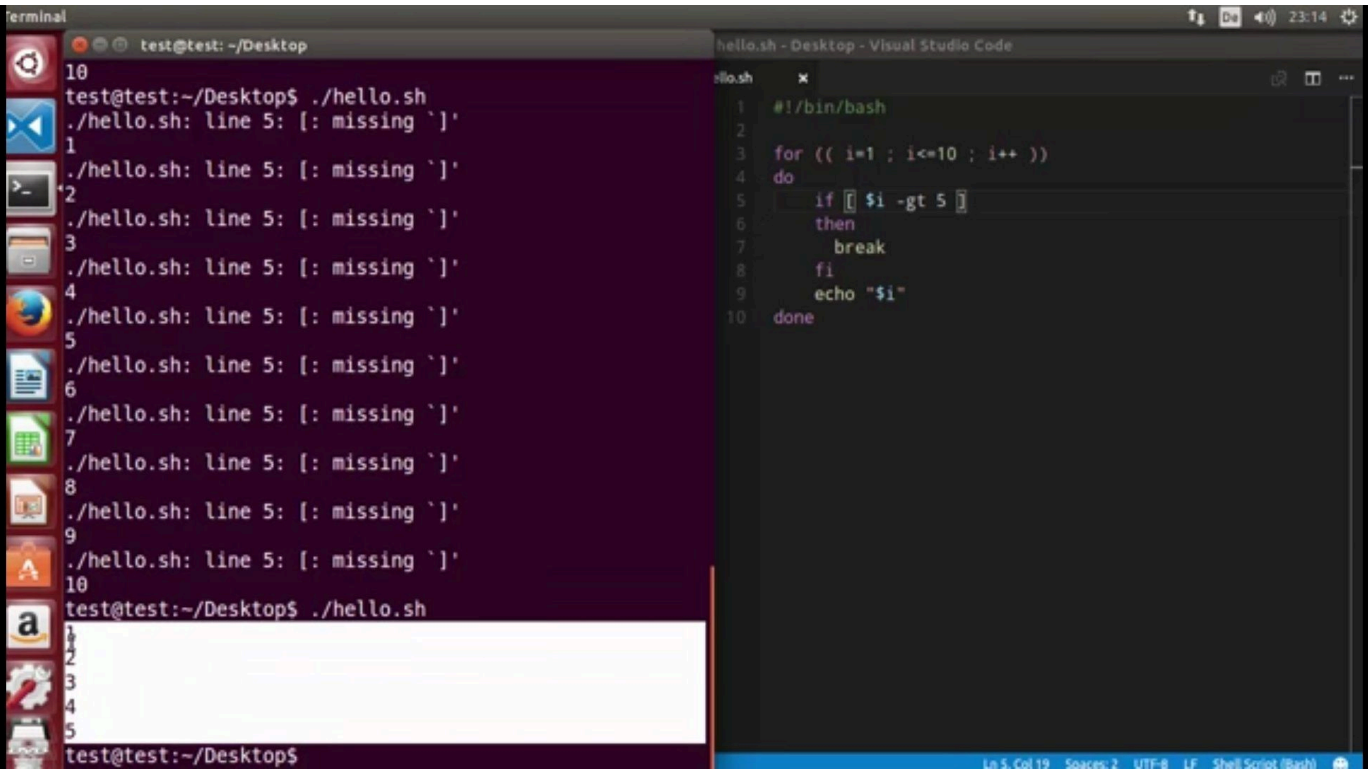

Terminal
13:15
test@test: ~
test@test:~$ ./hello.sh
doc.md
examples.desktop
hello.sh
test
test@test:~$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
for loops
for item in *
do
if [-f $item]
then
echo $item
fi
done
Ln 3, Col 14
Spaces: 2
UTF-8
LF
Shell Script (Bash)

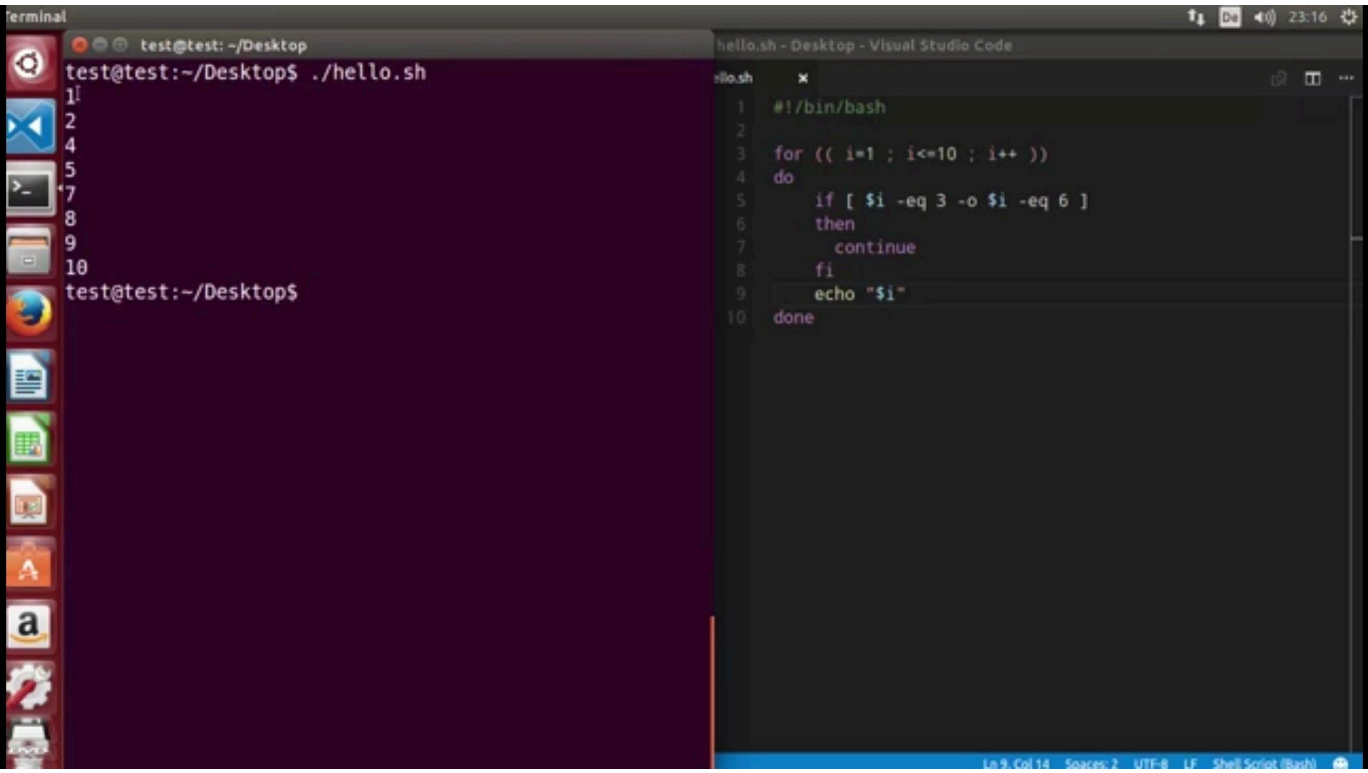

Visual Studio Code
22:23
test@test: ~/Desktop
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
select varName in list
do
command1
command2
....
......
commandN
done
Ln 3, Col 7 (6 selected)
Spaces: 4
UTF-8
LF
Shell Script (Bash)

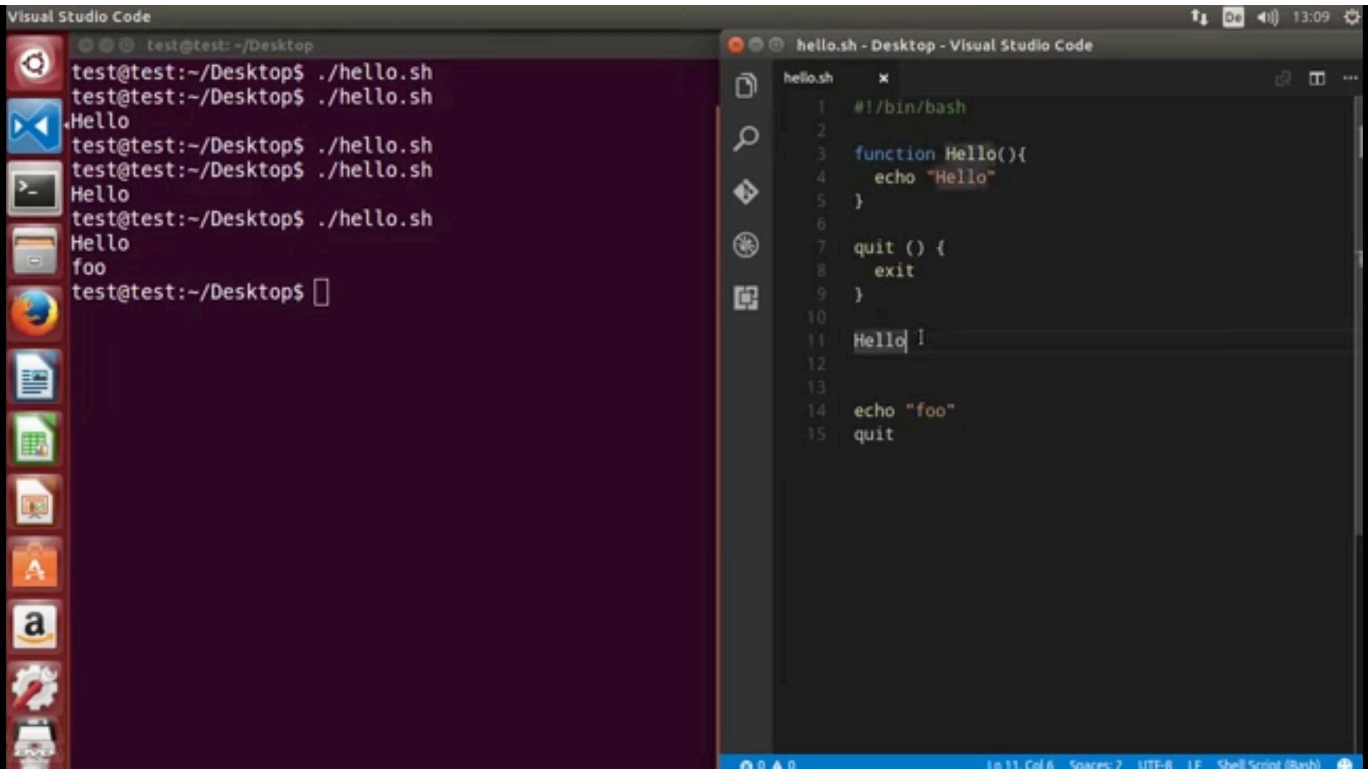

Terminal
22:24
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1) mark
2) john
3) tom
4) ben
#?
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
select name in mark john tom ben
do
echo "$name selected"
done
Ln 6, Col 5
Spaces: 4
UTF-8
LF
Shell Script (Bash)

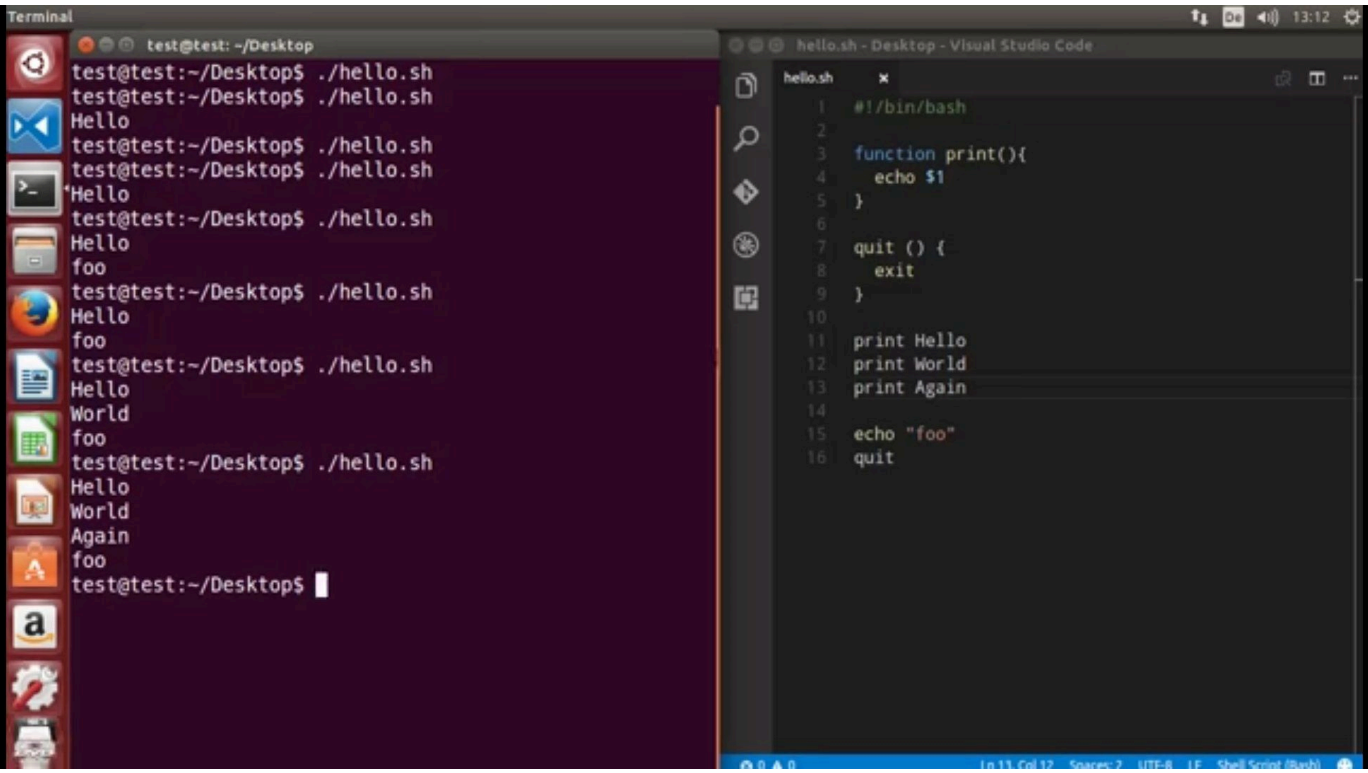

Visual Studio Code
22:25
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1) mark
2) john
3) tom
4) ben
#? 2
john selected
#?
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
select name in mark john tom ben
do
echo "$name selected"
done
Ln 6, Col 5
Spaces: 4
UTF-8
LF
Shell Script (Bash)

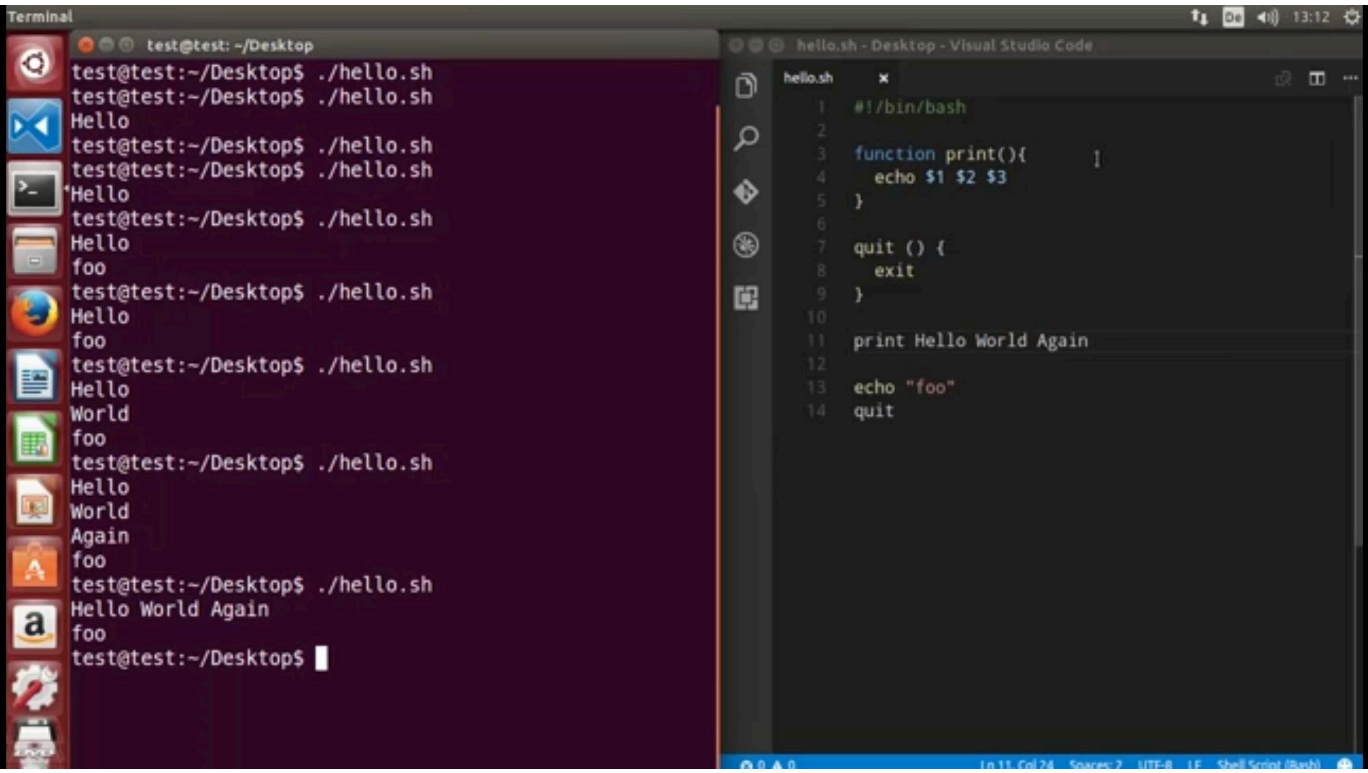

Visual Studio Code
22:29
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1) mark
2) john
3) tom
4) ben
#? 2
john selected
#? 3
tom selected
#? 1
mark selected
#? ^C
test@test:~/Desktop$ ./hello.sh
1) mark
2) john
3) tom
4) ben
#? 1
mark selected
#? 2
john selected
#? 3
tom selected
#? 4
ben selected
#?
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
select name in mark john tom ben
do
case $name in
mark)
echo mark selected
;;
john)
echo john selected
;;
tom)
echo tom selected
;;
ben)
echo ben selected
;;
*)
echo "Error please provide the no. between 1..4"
esac
done
Ln 7, Col 25
Spaces: 4
UTF-8
LF
Shell Script (Bash)

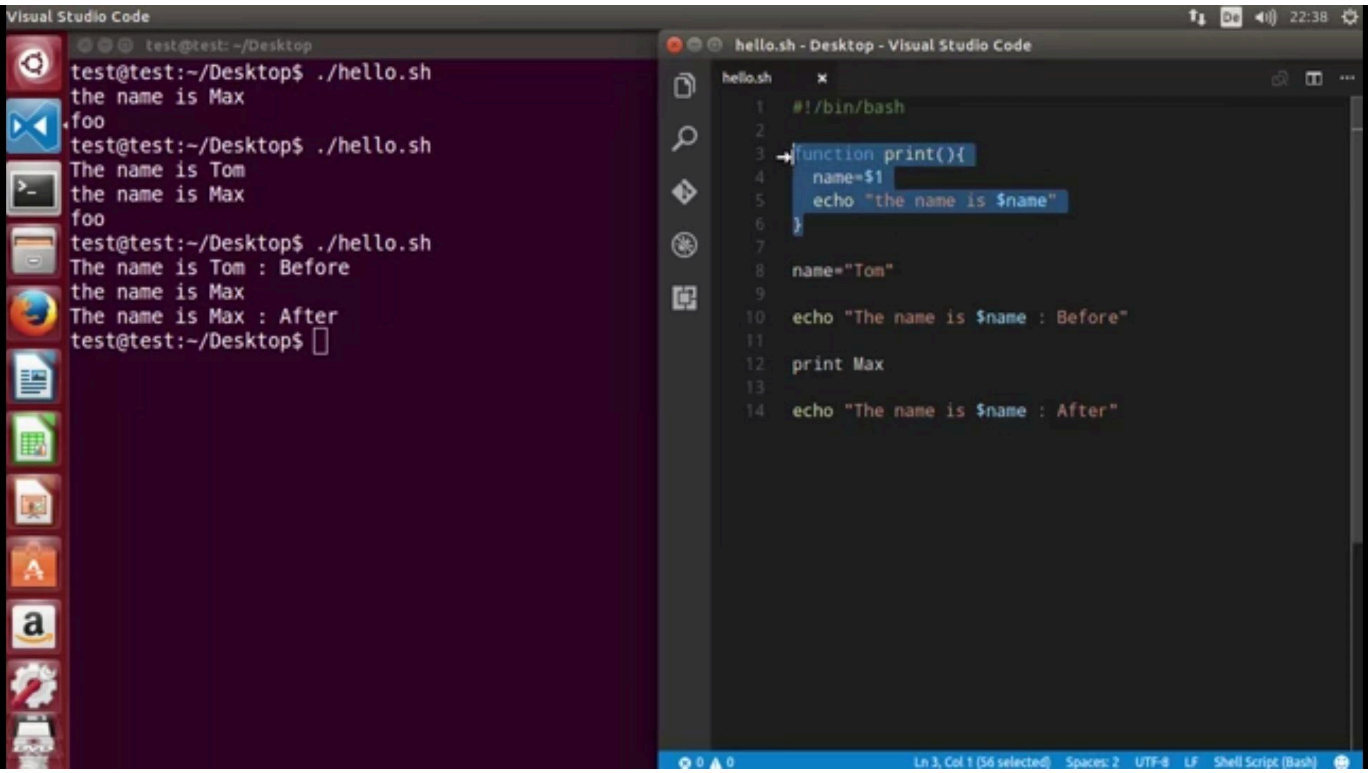

Terminal
test@test: ~/Desktop
10
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 5: [: missing `]'
1
./hello.sh: line 5: [: missing `]'
2
./hello.sh: line 5: [: missing `]'
3
./hello.sh: line 5: [: missing `]'
4
./hello.sh: line 5: [: missing `]'
5
./hello.sh: line 5: [: missing `]'
6
./hello.sh: line 5: [: missing `]'
7
./hello.sh: line 5: [: missing `]'
8
./hello.sh: line 5: [: missing `]'
9
./hello.sh: line 5: [: missing `]'
10
test@test:~/Desktop$ ./hello.sh
1
2
3
4
5
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
#!/bin/bash
for ((i=1 ; i<=10 ; i++))
do
if [$i -gt 5]
then
break
fi
echo "$i"
done

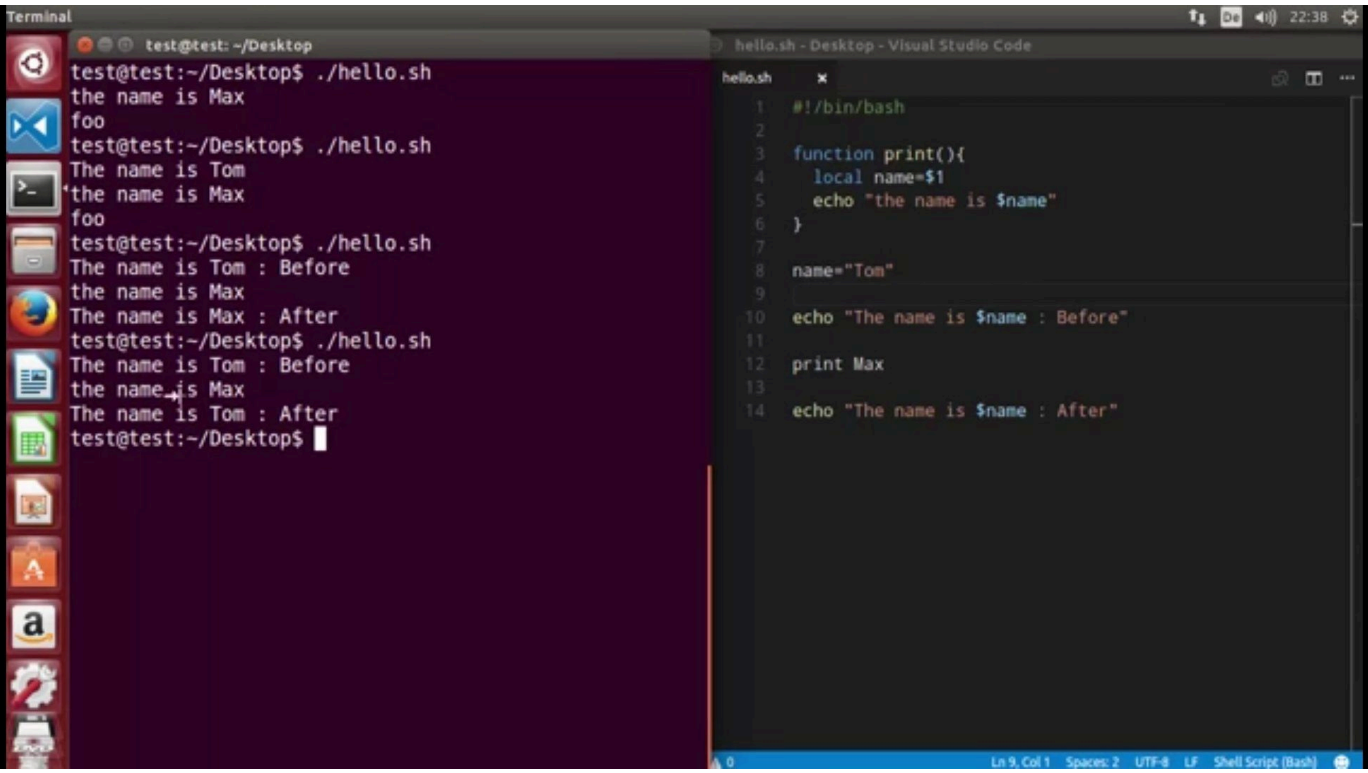

Terminal
23:16
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
1
2
4
5
7
8
9
10
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
#!/bin/bash
for ((i=1 ; i<=10 ; i++))
do
if [$i -eq 3 -o $i -eq 6]
then
continue
fi
echo "$i"
done

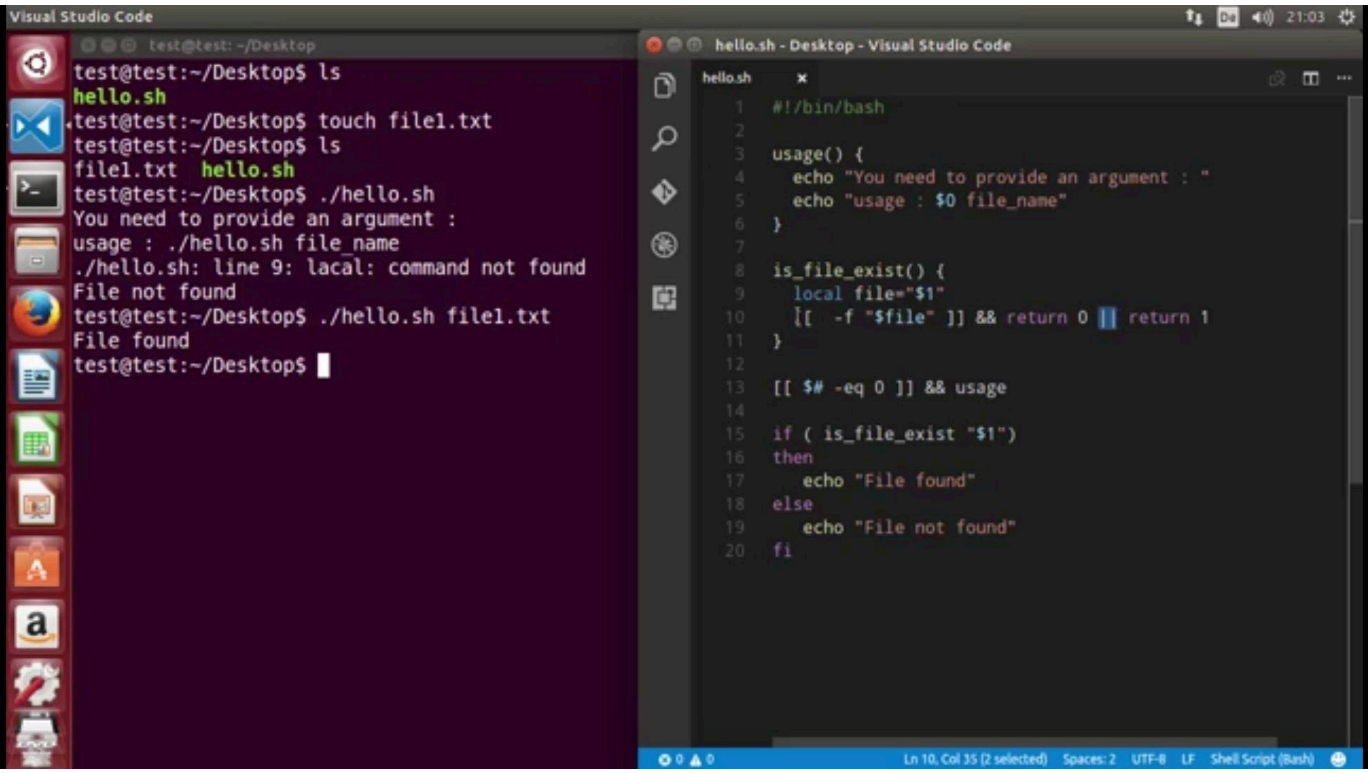

Visual Studio Code
13:09
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
Hello
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
Hello
test@test:~/Desktop$ ./hello.sh
Hello
foo
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
function Hello(){
echo "Hello"
}
quit () {
exit
}
Hello
echo "foo"
quit

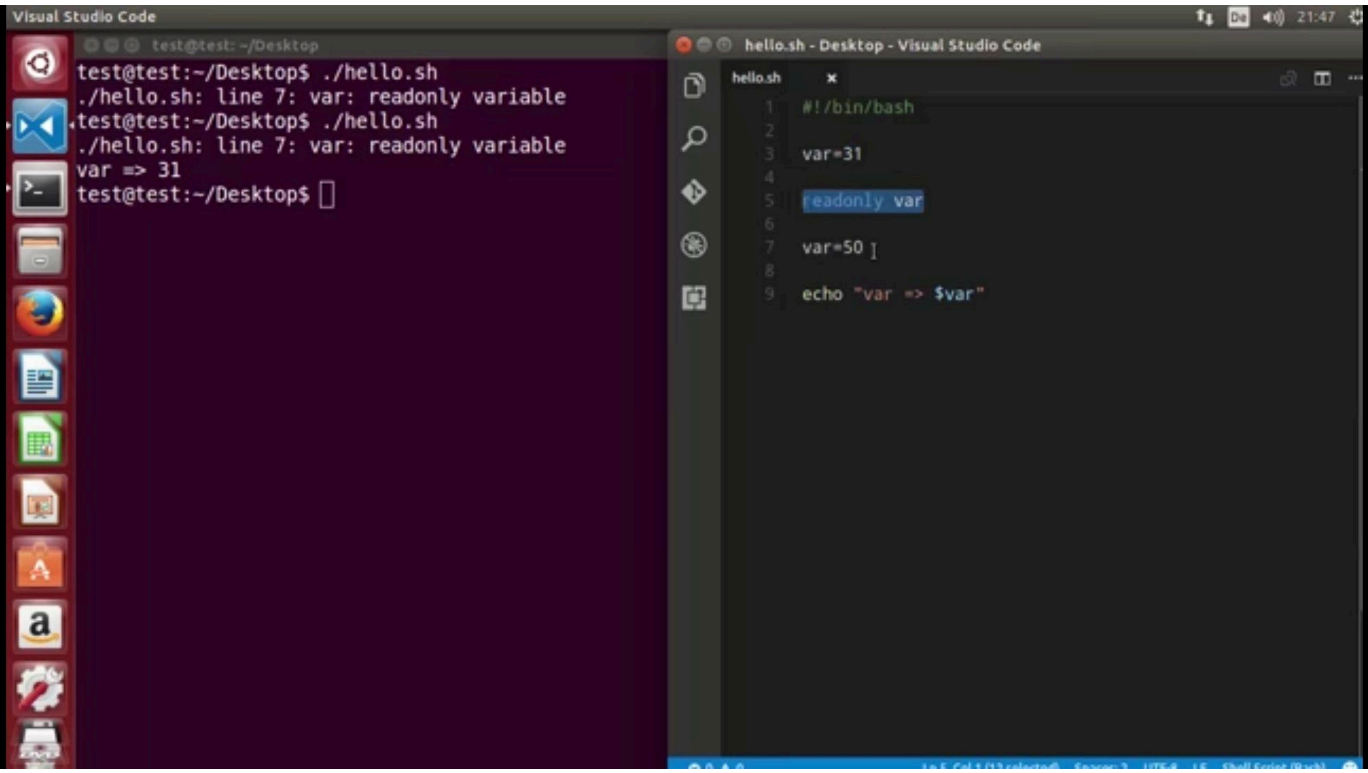

Terminal
13:12
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
Hello
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
Hello
test@test:~/Desktop$ ./hello.sh
Hello
foo
test@test:~/Desktop$ ./hello.sh
Hello
foo
test@test:~/Desktop$ ./hello.sh
Hello
World
foo
test@test:~/Desktop$ ./hello.sh
Hello
World
Again
foo
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
function print(){
echo $1
}
quit () {
exit
}
print Hello
print World
print Again
echo "foo"
quit

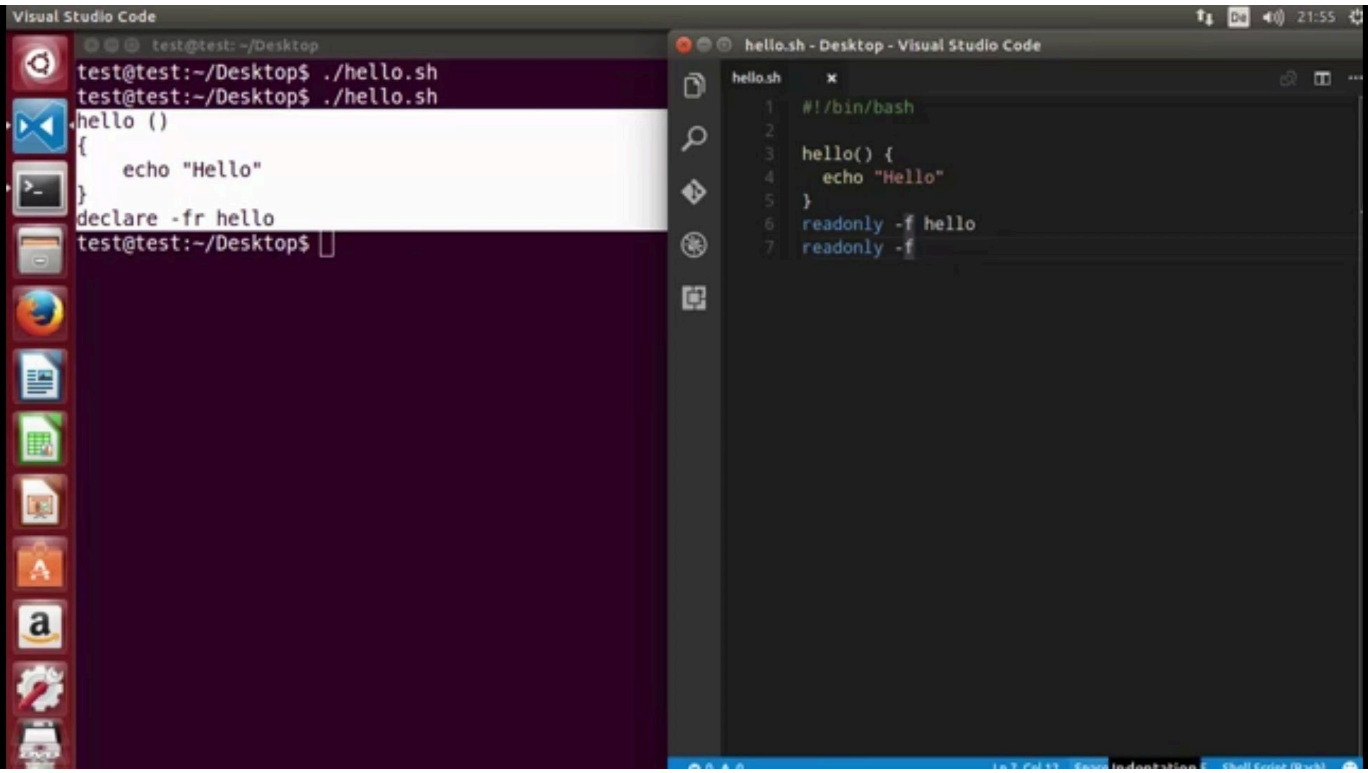

Terminal
13:12
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
Hello
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
Hello
test@test:~/Desktop$ ./hello.sh
Hello
foo
test@test:~/Desktop$ ./hello.sh
Hello
foo
test@test:~/Desktop$ ./hello.sh
Hello
World
foo
test@test:~/Desktop$ ./hello.sh
Hello
World
Again
foo
test@test:~/Desktop$ ./hello.sh
Hello World Again
foo
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
function print(){
echo $1 $2 $3
}
quit () {
exit
}
print Hello World Again
echo "foo"
quit

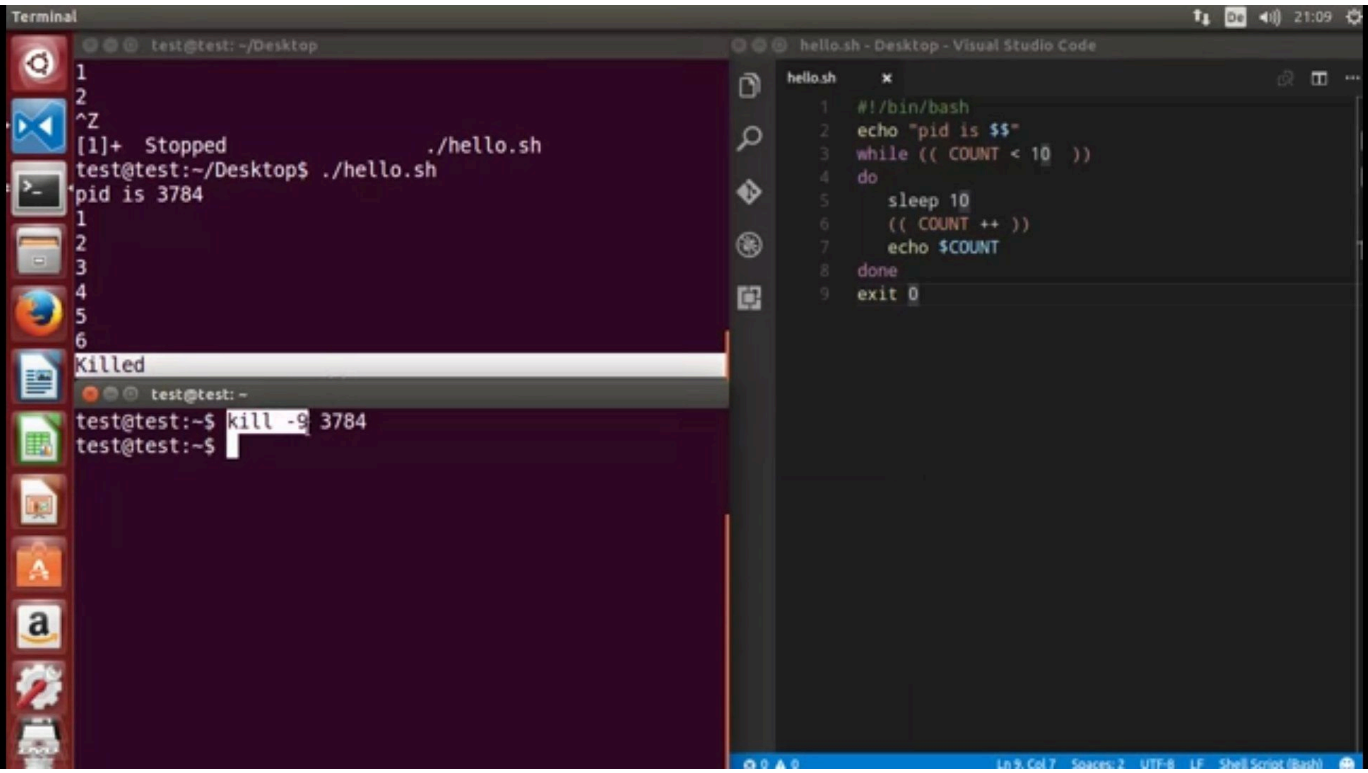

Visual Studio Code
22:38
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
the name is Max
foo
test@test:~/Desktop$ ./hello.sh
The name is Tom
the name is Max
foo
test@test:~/Desktop$ ./hello.sh
The name is Tom : Before
the name is Max
The name is Max : After
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
function print(){
name=$1
echo "the name is $name"
}
name="Tom"
echo "The name is $name : Before"
print Max
echo "The name is $name : After"
Ln 3, Col 1 (56 selected)
Spaces: 2
UTF-8
LF
Shell Script (Bash)

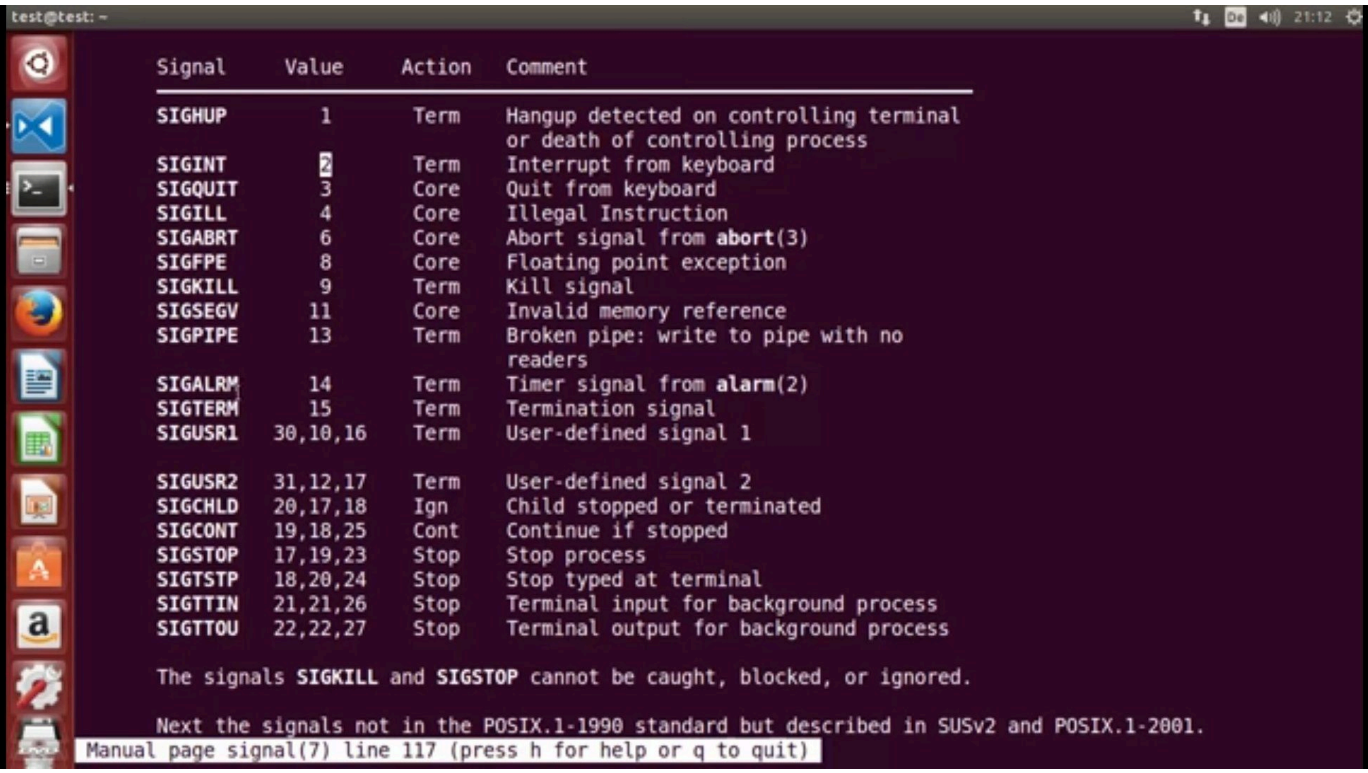

Terminal
22:38
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
the name is Max
foo
test@test:~/Desktop$ ./hello.sh
The name is Tom
the name is Max
foo
test@test:~/Desktop$ ./hello.sh
The name is Tom : Before
the name is Max
The name is Max : After
test@test:~/Desktop$ ./hello.sh
The name is Tom : Before
the name is Max
The name is Tom : After
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
function print(){
local name=$1
echo "the name is $name"
}
name="Tom"
echo "The name is $name : Before"
print Max
echo "The name is $name : After"
Ln 9, Col 1
Spaces: 2
UTF-8
LF
Shell Script (Bash)

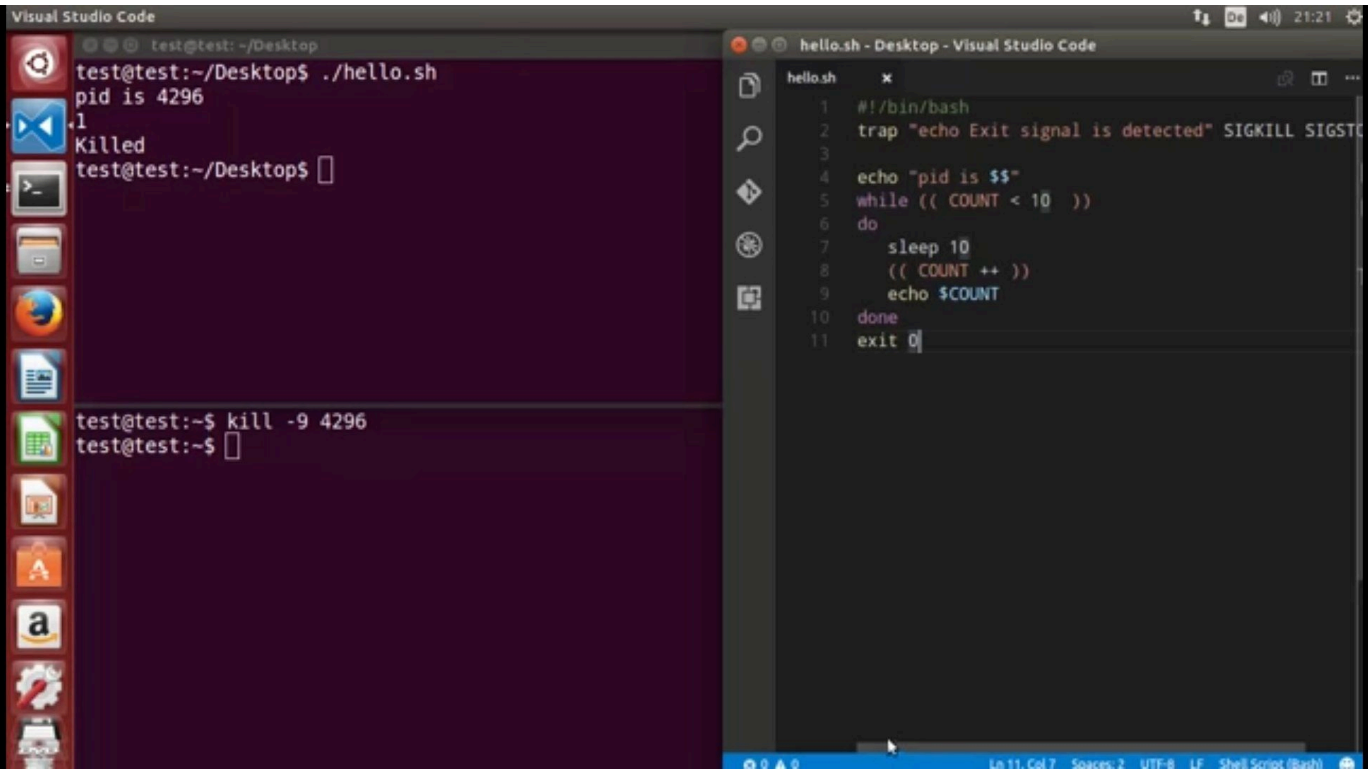

Visual Studio Code
21:03
test@test: ~/Desktop
test@test:~/Desktop$ ls
hello.sh
test@test:~/Desktop$ touch file1.txt
test@test:~/Desktop$ ls
file1.txt hello.sh
test@test:~/Desktop$ ./hello.sh
You need to provide an argument :
usage : ./hello.sh file_name
./hello.sh: line 9: lacal: command not found
File not found
test@test:~/Desktop$ ./hello.sh file1.txt
File found
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
usage() {
echo "You need to provide an argument : "
echo "usage : $0 file_name"
}
is_file_exist() {
local file="$1"
[[-f "$file"]] && return 0 || return 1
}
[[$# -eq 0]] && usage
if (is_file_exist "$1")
then
echo "File found"
else
echo "File not found"
fi
Ln 10, Col 35 (2 selected)
Spaces: 2
UTF-8
LF
Shell Script (Bash)

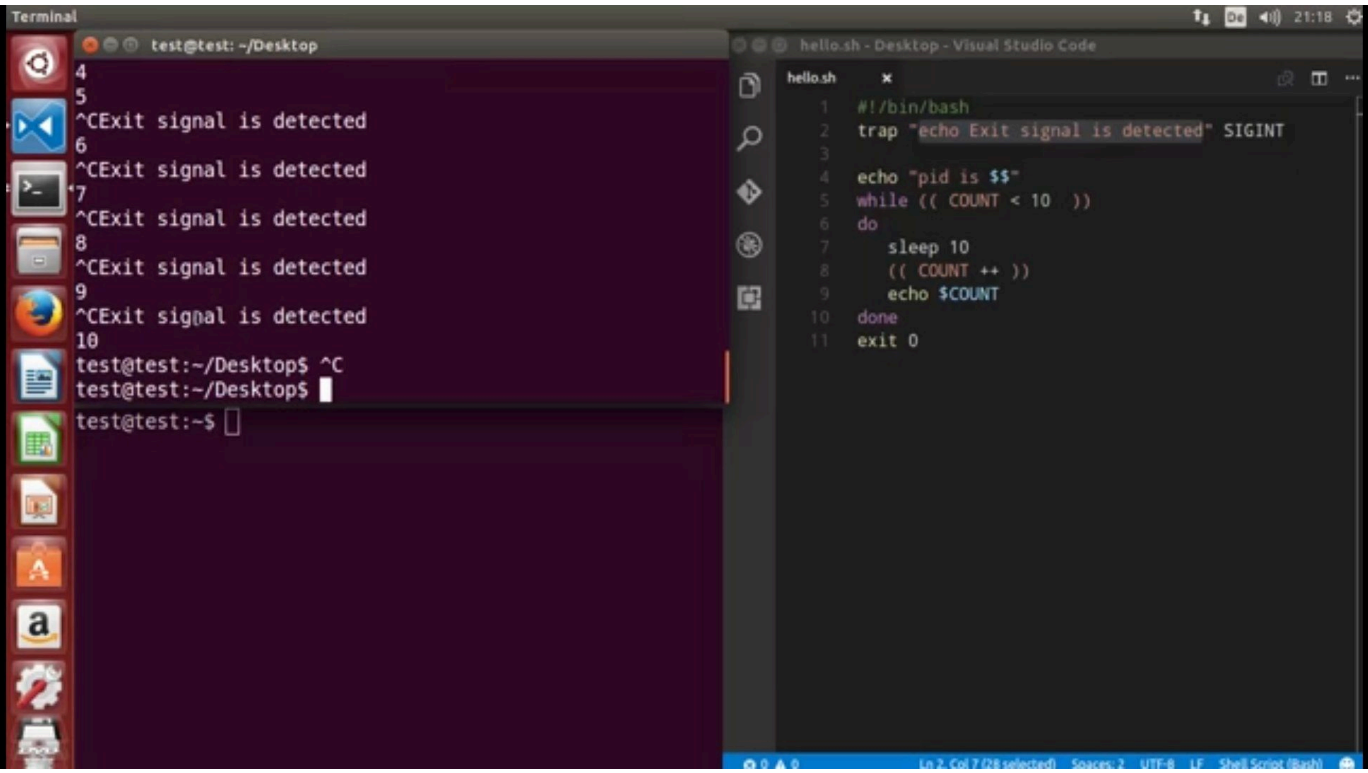

Visual Studio Code
21:47
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 7: var: readonly variable
test@test:~/Desktop$ ./hello.sh
./hello.sh: line 7: var: readonly variable
var => 31
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
1 #!/bin/bash
2
3 var=31
4
5 readonly var
6
7 var=50
8
9 echo "var => $var"

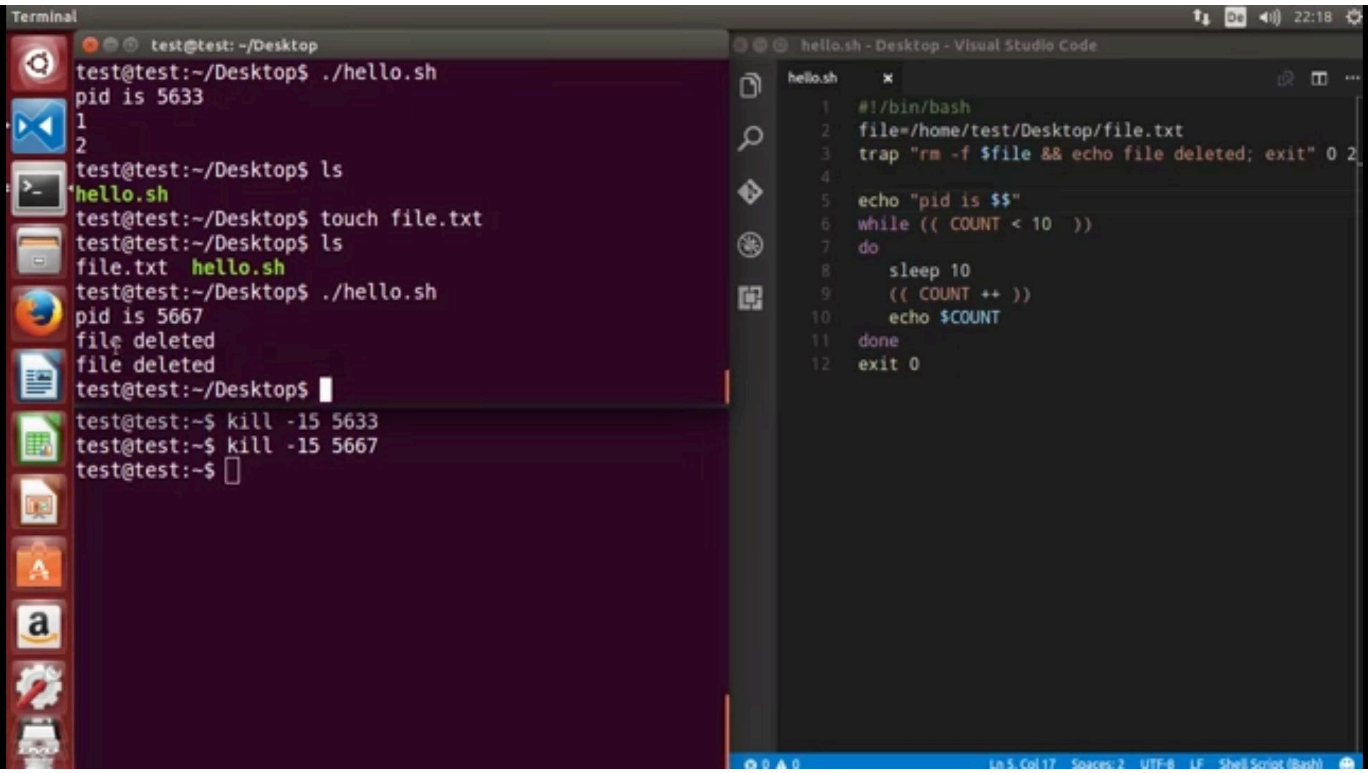

Visual Studio Code
21:55
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
test@test:~/Desktop$ ./hello.sh
hello ()
{
echo "Hello"
}
declare -fr hello
test@test:~/Desktop$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
hello() {
echo "Hello"
}
readonly -f hello
readonly -f

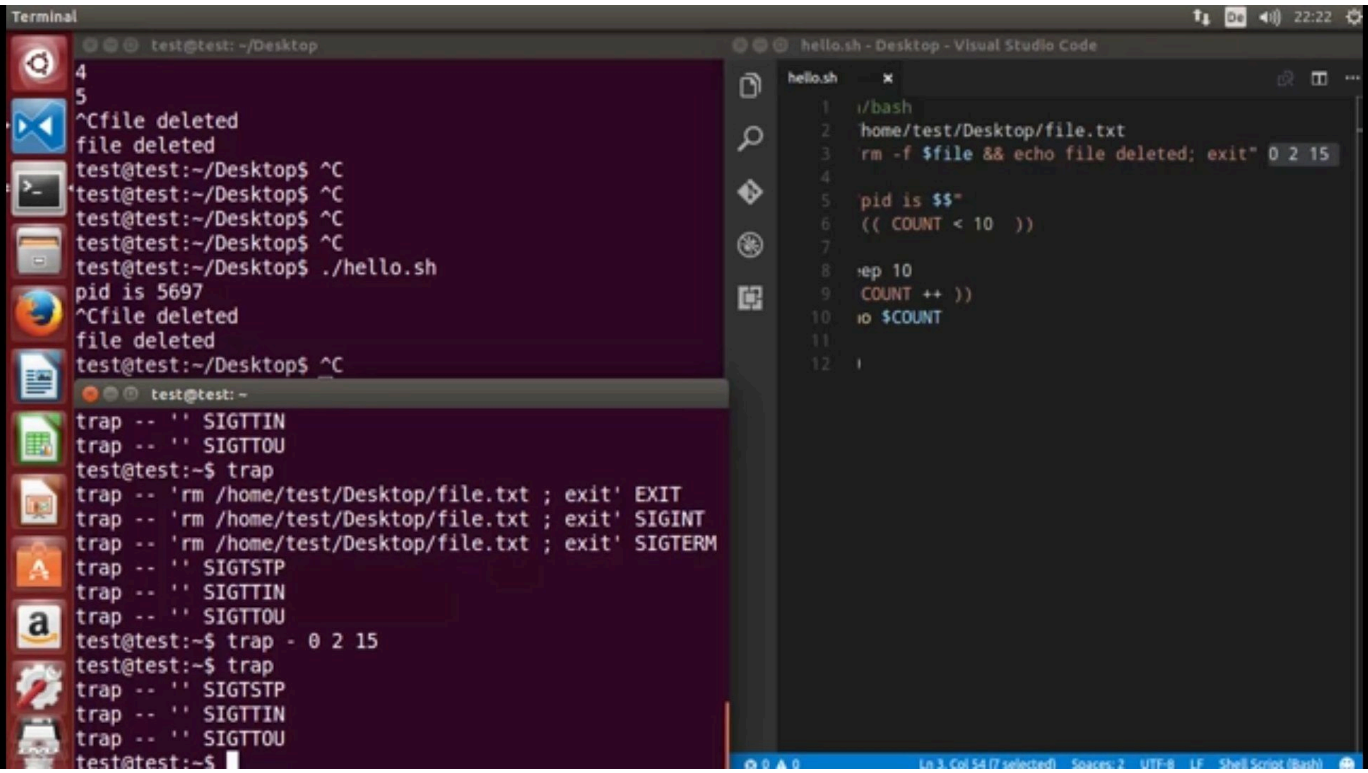

Terminal
21:09
test@test: ~/Desktop
1
2
^Z
[1]+ Stopped ./hello.sh
test@test:~/Desktop$ ./hello.sh
pid is 3784
1
2
3
4
5
6
Killed
test@test: ~
test@test:~$ kill -9 3784
test@test:~$
hello.sh - Desktop - Visual Studio Code
hello.sh
1 #!/bin/bash
2 echo "pid is $$"
3 while ((COUNT < 10))
4 do
5 sleep 10
6 ((COUNT ++))
7 echo $COUNT
8 done
9 exit 0

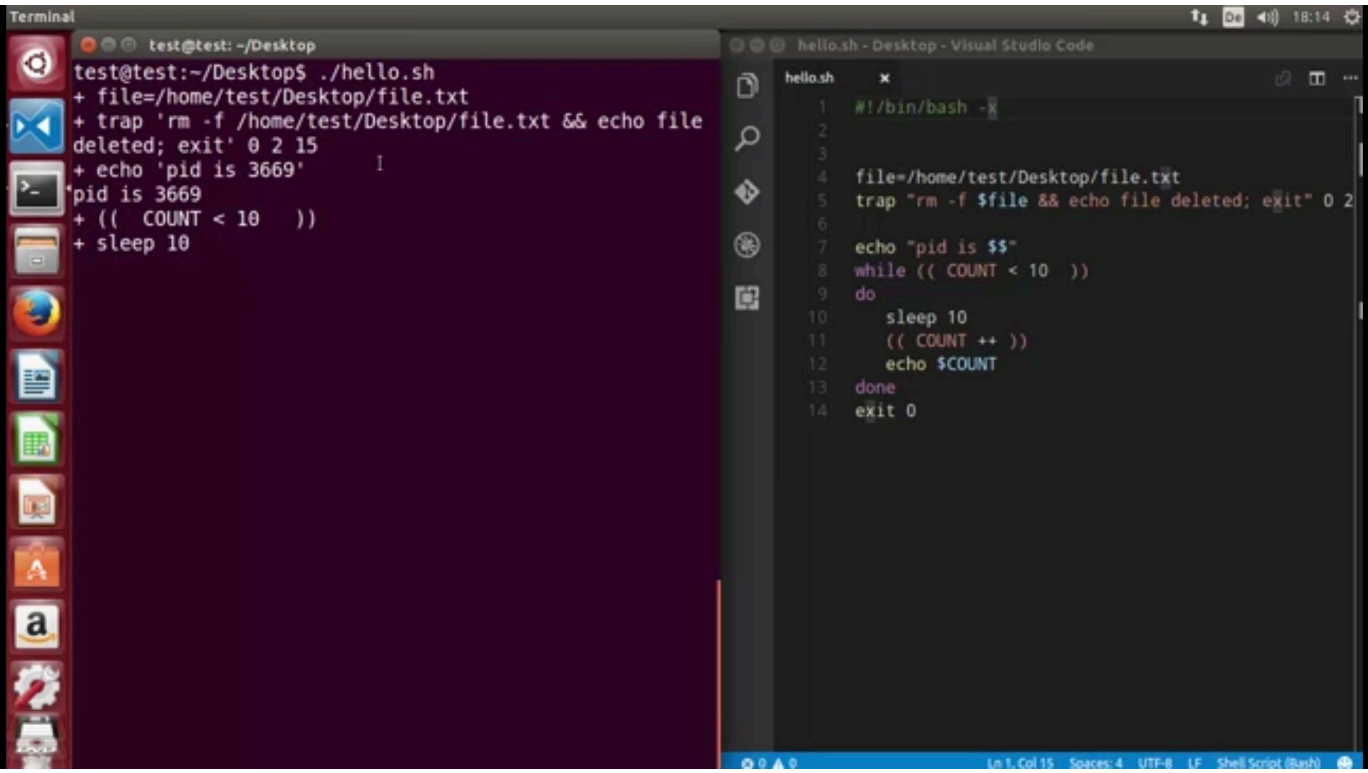

test@test: ~
21:12
Signal Value Action Comment
SIGHUP 1 Term Hangup detected on controlling terminal or death of controlling process
SIGINT 2 Term Interrupt from keyboard
SIGQUIT 3 Core Quit from keyboard
SIGILL 4 Core Illegal Instruction
SIGABRT 6 Core Abort signal from abort(3)
SIGFPE 8 Core Floating point exception
SIGKILL 9 Term Kill signal
SIGSEGV 11 Core Invalid memory reference
SIGPIPE 13 Term Broken pipe: write to pipe with no readers
SIGALRM 14 Term Timer signal from alarm(2)
SIGTERM 15 Term Termination signal
SIGUSR1 30,10,16 Term User-defined signal 1
SIGUSR2 31,12,17 Term User-defined signal 2
SIGCHLD 20,17,18 Ign Child stopped or terminated
SIGCONT 19,18,25 Cont Continue if stopped
SIGSTOP 17,19,23 Stop Stop process
SIGTSTP 18,20,24 Stop Stop typed at terminal
SIGTTIN 21,21,26 Stop Terminal input for background process
SIGTTOU 22,22,27 Stop Terminal output for background process
The signals SIGKILL and SIGSTOP cannot be caught, blocked, or ignored.
Next the signals not in the POSIX.1-1990 standard but described in SUSv2 and POSIX.1-2001.
Manual page signal(7) line 117 (press h for help or q to quit)

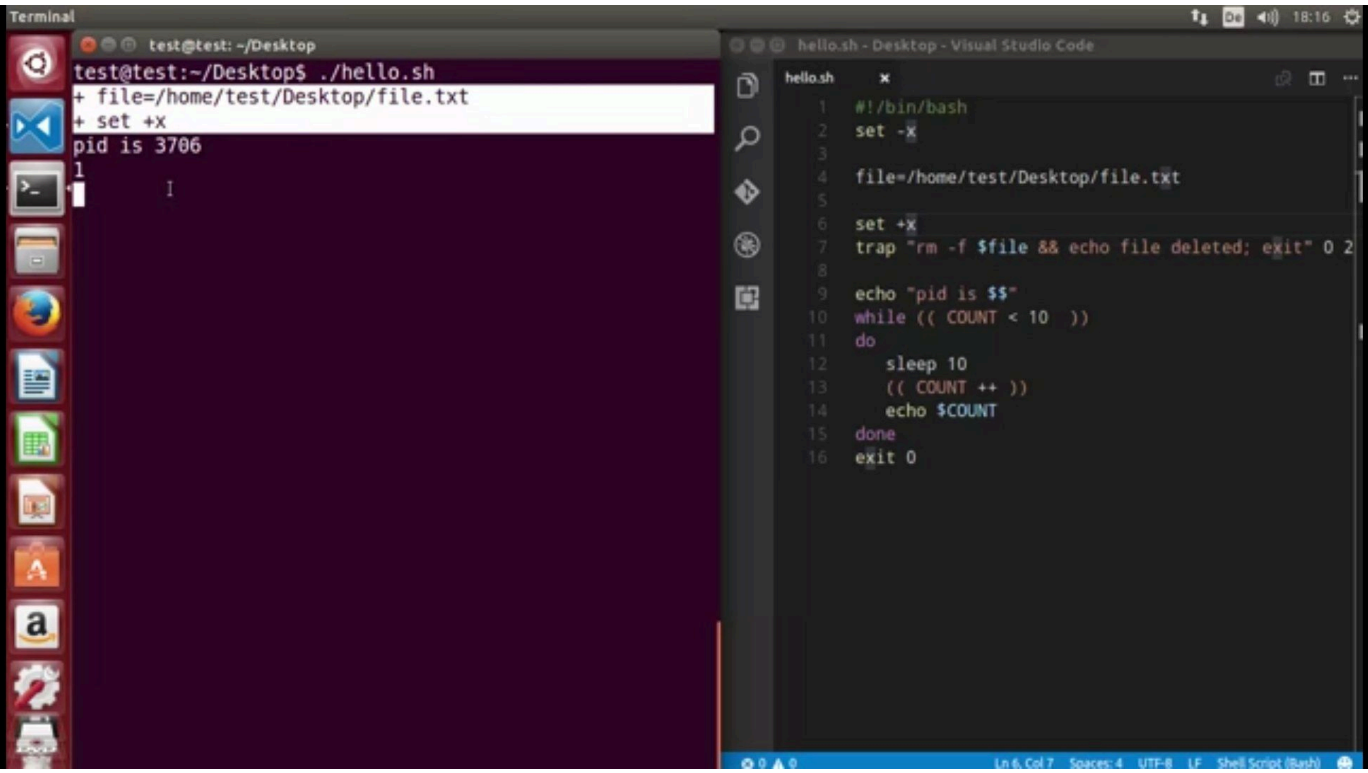

Visual Studio Code
21:21
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
pid is 4296
1
Killed
test@test:~/Desktop$
test@test:~$ kill -9 4296
test@test:~$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
trap "echo Exit signal is detected" SIGKILL SIGST
echo "pid is $$"
while ((COUNT < 10))
do
sleep 10
((COUNT ++))
echo $COUNT
done
exit 0


Terminal
21:18
test@test: ~/Desktop
4
5
^CExit signal is detected
6
^CExit signal is detected
7
^CExit signal is detected
8
^CExit signal is detected
9
^CExit signal is detected
10
test@test:~/Desktop$ ^C
test@test:~/Desktop$
test@test:~$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
trap "echo Exit signal is detected" SIGINT
echo "pid is $$"
while ((COUNT < 10))
do
sleep 10
((COUNT ++))
echo $COUNT
done
exit 0


Terminal
22:18
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
pid is 5633
1
2
test@test:~/Desktop$ ls
hello.sh
test@test:~/Desktop$ touch file.txt
test@test:~/Desktop$ ls
file.txt hello.sh
test@test:~/Desktop$ ./hello.sh
pid is 5667
file deleted
file deleted
test@test:~/Desktop$
test@test:~$ kill -15 5633
test@test:~$ kill -15 5667
test@test:~$
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
file=/home/test/Desktop/file.txt
trap "rm -f $file && echo file deleted; exit" 0 2
echo "pid is $$"
while ((COUNT < 10))
do
sleep 10
((COUNT ++))
echo $COUNT
done
exit 0


Terminal
22:22
test@test: ~/Desktop
4
5
^Cfile deleted
file deleted
test@test:~/Desktop$ ^C
test@test:~/Desktop$ ^C
test@test:~/Desktop$ ^C
test@test:~/Desktop$ ^C
test@test:~/Desktop$ ./hello.sh
pid is 5697
^Cfile deleted
file deleted
test@test:~/Desktop$ ^C
test@test: ~
trap -- '' SIGTTIN
trap -- '' SIGTTOU
test@test:~$ trap
trap -- 'rm /home/test/Desktop/file.txt ; exit' EXIT
trap -- 'rm /home/test/Desktop/file.txt ; exit' SIGINT
trap -- 'rm /home/test/Desktop/file.txt ; exit' SIGTERM
trap -- '' SIGTSTP
trap -- '' SIGTTIN
trap -- '' SIGTTOU
test@test:~$ trap - 0 2 15
test@test:~$ trap
trap -- '' SIGTSTP
trap -- '' SIGTTIN
trap -- '' SIGTTOU
test@test:~$
hello.sh - Desktop - Visual Studio Code
hello.sh
/bash
home/test/Desktop/file.txt
rm -f $file && echo file deleted; exit" 0 2 15
pid is $$"
((COUNT < 10))
ep 10
COUNT ++))
o $COUNT


Terminal
18:14
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
+ file=/home/test/Desktop/file.txt
+ trap 'rm -f /home/test/Desktop/file.txt && echo file deleted; exit' 0 2 15
+ echo 'pid is 3669'
pid is 3669
+ ((COUNT < 10))
+ sleep 10
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash -x
file=/home/test/Desktop/file.txt
trap "rm -f $file && echo file deleted; exit" 0 2
echo "pid is $$"
while ((COUNT < 10))
do
sleep 10
((COUNT ++))
echo $COUNT
done
exit 0
Ln 1, Col 15
Spaces: 4
UTF-8
LF
Shell Script (Bash)


Terminal
18:16
test@test: ~/Desktop
test@test:~/Desktop$ ./hello.sh
+ file=/home/test/Desktop/file.txt
+ set +x
pid is 3706
1
hello.sh - Desktop - Visual Studio Code
hello.sh
#!/bin/bash
set -x
file=/home/test/Desktop/file.txt
set +x
trap "rm -f $file && echo file deleted; exit" 0 2
echo "pid is $$"
while ((COUNT < 10))
do
sleep 10
((COUNT ++))
echo $COUNT
done
exit 0
Ln 6, Col 7
Spaces: 4
UTF-8
LF
Shell Script (Bash)