

Assignment 0

1. Create a simple shell script to tell the user about their session – they need to know:

- What their username is
- What the current date is
- What the time is
- What their current working directory is
- How many files they have in that directory
- What is the biggest file in their current directory

[illegible]

Assignment 1

Create a directory with a few test files in it (the files can be empty). Now write a script that for

every file in that directory you rename it to have an extension of today's date in YYYYMMDD

format.

```
File Edit View Search Terminal Help
[root@Control TestFolder]#
[root@Control TestFolder]# ls -la
total 4
drwxr-xr-x. 2 root root 85 Dec 11 11:20 .
drwxr-xr-x. 3 root root 24 Dec 11 04:55 ..
-rwxr-xr-x. 1 root root 161 Dec 11 11:16 files.sh
-rw-r--r--. 1 root root 0 Dec 11 11:08 test1.txt
-rw-r--r--. 1 root root 0 Dec 11 11:08 test2.txt
-rw-r--r--. 1 root root 0 Dec 11 11:08 test3.txt
-rw-r--r--. 1 root root 0 Dec 11 04:56 .txt
[root@Control TestFolder]#
[root@Control TestFolder]# ./files.sh
[root@Control TestFolder]#
[root@Control TestFolder]# ls -la
total 16
drwxr-xr-x. 2 root root 142 Dec 11 11:29 .
drwxr-xr-x. 3 root root 24 Dec 11 04:55 ..
-rwxr-xr-x. 1 root root 161 Dec 11 11:16 files_20201211.sh
-rw-r--r--. 1 root root 12288 Dec 11 11:27 .files.sh.swp
-rw-r--r--. 1 root root 0 Dec 11 11:08 test1_20201211.txt
-rw-r--r--. 1 root root 0 Dec 11 11:08 test2_20201211.txt
-rw-r--r--. 1 root root 0 Dec 11 11:08 test3_20201211.txt
-rw-r--r--. 1 root root 0 Dec 11 04:56 .txt
[root@Control TestFolder]#
```

Assignment 2

Write a script that takes a number as an input and reverses it out to the user.
For example, if the
original number is 74985, the output should be 58947.

File Edit View Search Terminal Help	File Edit View Search Terminal Help
[root@Control ~]# ./reverse.sh	#!/bin/bash
Enter a number: 74985	echo " "
Reverse number is:	read -p "Enter a number: " num
58947	echo " "
[root@Control ~]#	echo "Reverse number is: "
	echo \$num rev
	echo " "
	~
	~
	~

Assignment 3

Write a script to validate how secure someone's password is. Things you would care about: - Length should be 8 or more characters - The password should contain numbers and letters - There should be both uppercase and lowercase letters

File Edit View Search Terminal Help	File Edit View Search Terminal Help
[root@Control ~]# ./password.sh	#!/bin/bash
Enter the Password	echo ""
123	echo "Enter the Password"
Weak Password!!!... Password length must be >= 8	echo ""
[root@Control ~]# ./password.sh	read password
Enter the Password	len="\${#password}"
12345678	if [\$len -ge 8]; then
Weak Password!!!... Password must contain Numbers, Uppercase, Lowecase, and Special Character	echo "\$password" grep -q [0-9]
[root@Control ~]# ./password.sh	if [\$? -eq 0]; then
Enter the Password	echo "\$password" grep -q [A-Z]
12345678A	if [\$? -eq 0]; then
Weak Password!!!... Password must contain Numbers, Uppercase, Lowecase, and Special Character	echo "\$password" grep -q [a-z]
[root@Control ~]# ./password.sh	if [\$? -eq 0]; then
Enter the Password	echo "\$password" grep -q [0-9]
12345678Aa	echo "\$password" grep -q [0-9]
Weak Password!!!... Password must contain Numbers, Uppercase, Lowecase, and Special Character	if [\$? -eq 0]; then
[root@Control ~]# ./password.sh	echo "Cool this is a Strong Password..."
Enter the Password	else
12345678Aa@	echo ""
Cool this is a Strong Password...	echo "Weak Password!!!..."
[root@Control ~]#	echo "Password must contain Numbers, Uppercase, Lowecase, and Special Character"
	fi
	else
	echo ""
	echo "Weak Password!!!..."
	echo "Password must contain Numbers, Uppercase, Lowecase, and Special Character"
	fi
	else
	echo ""
	echo "Weak Password!!!... Password length must be >= 8"
	fi
	echo ""
	~
	~
	"password.sh" 41L, 952C