| **Assignment Case** | Description: LogoBINUS-University |
| --- | --- |
| OSH1 Special |
| **Periode Berlaku** Semester Genap 2020/2021  ***Valid on*** *Even Semester Year 2020/2021* | **Software Laboratory Center**  **Assistant Recruitment 21-2** |

## Materi

*Material*

* Linux Command Console

## Soal

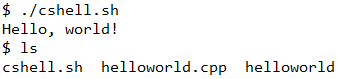
*Case*

**OSH1 Special**

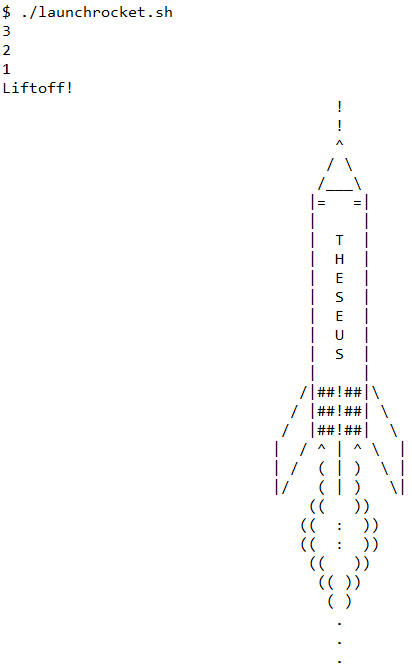
1. **Create**  a new user called **“Theseus”**
2. **Set** Theseus’ Password as **“welcomehometheseus”**
3. **Grant** Theseus his **sudo privileges**
4. **Switch user** to Theseus in the same terminal
5. **Create a new group** called **“Kings”**
6. **Add** Theseus into the **Kings** group
7. Using an absolute path from / directory, **create a new directory** in **/home** called **“Athens”**
8. **Set Athens ownership** to the **user Theseus** and the **group Kings**
9. As a superuser, change Theseus’ **default home directory** to **Athens**
10. Create the following directory structure in one line:

|  |
| --- |

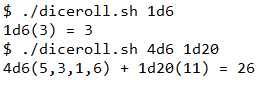
1. Create a shell script that does the following:
   1. create a .cpp file and fill it with a simple hello world code in C or C++ language
   2. compile the .cpp file into an executable
   3. run the executable



1. Create a shell script that does the following:
   1. print a countdown from 3 to 1, with 1-second intervals
   2. print each line from ‘rocket.txt’ with 0.1-second intervals
   3. clear the screen



1. Install and run an SSH Server with the following configurations:
   1. Listening Port: 515
   2. Root login: Disabled
   3. Login using password: Disabled
   4. Whitelist:
      1. prk: Can login using password
   5. Make sure to restart the SSH Server after configuring it
2. Setup an SSH Key-based authentication between your host machine and the user ‘Theseus’, with the following configuration:
   1. Cryptographic algorithm: RSA
   2. Length of bits: 4096
   3. Key passphrase: “Minotaur”
   4. Make sure your host machine can connect to your Ubuntu machine via the user ‘Theseus’
3. Create a shell script called ‘diceroll’ that does the following:
   1. receives at least 1 argument: xdy
      1. where x is an Integer denoting the amount of dice rolls
      2. where d is just the character ‘d’
      3. where y is an Integer denoting the amount of sides of the dice
   2. roll the dice using this formula:
      1. for each amount of dice rolls, randomize between 1 and the amount of sides (inclusive), with equal chances
   3. print each rolls, and its grand total



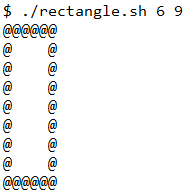
1. Create a shell script that does the following:
   1. read ‘donald.txt’
   2. count and print how many the character ‘2’ occurs



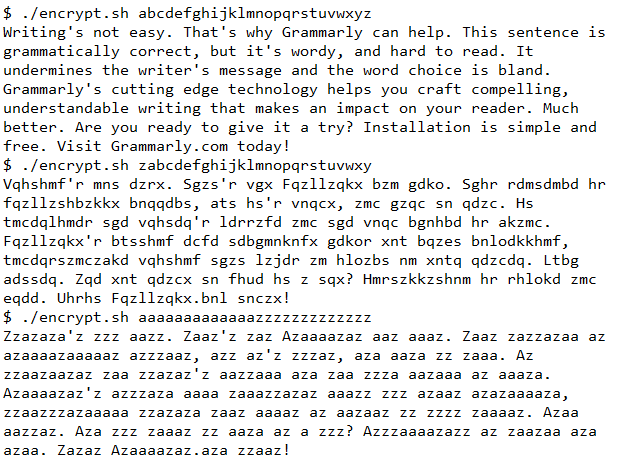
1. Create a shell script named ‘timer.sh’ that does the following:
   1. count up by 1 from 0, with 1-second intervals
   2. on receiving SIGINT, print how many seconds have elapsed, and terminate the process



1. Run ‘timer.sh’ in the background
2. Terminate ‘timer.sh’ from background processes using SIGINT signal
3. Create a shell script that does the following:
   1. receives 2 arguments: x and y
      1. where x is the width of the rectangle
      2. where y is the height of the rectangle
   2. draw a hollow rectangle with width x and height y using any visible character



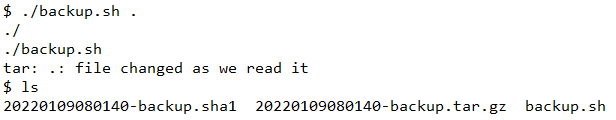
1. Create a shell script called ‘encrypt.sh’ that does the following:
   1. receive 1 argument: repr
      1. where repr has exactly 26 of any alphabetical characters
      2. each character will represent each alphabet from A to Z
   2. encrypt and print the contents of ‘grammarly.txt’ by swapping the alphabets with their representatives, preserving non-alphabetical characters



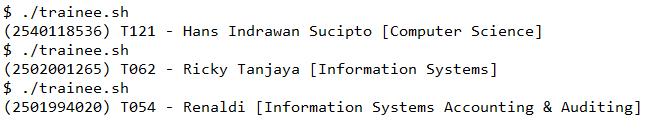
1. Create a shell script called ‘bubble.sh’ that does the following:
   1. receives at least 1 argument: x
      1. where x is an Integer
   2. sort and print all of the arguments in ascending order using bubble sort implemented within the shell script



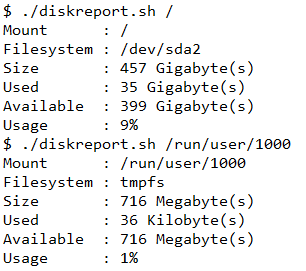
1. Create a shell script that does the following:
   1. receives 1 argument: path
      1. where path is a directory path
   2. compress the directory with filename format ’[year][month][day][hour][minute][second]-backup.tar.gz’ using tar, with gzip compression
   3. calculate sha1 checksum of the compressed file and output it to a file named ’[year][month][day][hour][minute][second]-backup.sha1’



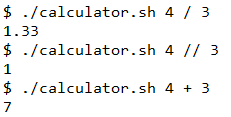
1. Create a shell script that does the following:
   1. read ‘trainees.csv’
   2. print a random trainee with the format: ‘({NIM}) {TXXX} {Full Name} [{Major}]’



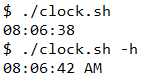
1. Create a shell script that does the following:
   1. receives 1 argument: mount location
      1. where mount location is the volume mount location, such as ‘/’ or ‘/boot/efi’ or etc.
   2. print the disk information of the mount location, including:
      1. mount location
      2. filesystem location
      3. size of volume
      4. amount of volume used
      5. amount of volume available
      6. usage percentage
   3. print size units in human-readable format (Kilobytes, Megabytes, Gigabytes, etc.)



1. Create a shell script that does the following:
   1. receive 3 arguments: number1, operator, and number2
      1. where number1 and number2 are Integers
      2. where operator is one of the following: +. -, \*, \*\*, /, //
   2. print the result of the arithmetic operation between 2 numbers
      1. where + is addition
      2. where - is subtraction
      3. where \* is multiplication
      4. where \* is power operation
      5. where / is floating point division (2 decimal places)
      6. where // is Integer division
   3. if the operator is anything other than the 6 above, print an error message “operator not supported”



1. Create a shell script called ‘clock.sh’ that does the following:
   1. can receive 1 argument: -h
   2. print the current time, updating each second
   3. if the argument -h is applied, print the time in human-readable format



1. Show commands history
2. Clear commands history
3. Schedule shutdown in 5 minutes