



**CSE 3033 – OPERATING SYSTEMS
PROJECT1**

Authors:

- 1) 150117005 Kadir ACUN
- 2) 150117905 Erhan YALNIZ

CSE3063: Operating Systems, Fall 2020

Date Submitted: Nov 23, 2020

PROBLEM DEFINITIONS AND IMPLEMENTATION DETAILS

- 1) Firstly, we take a command line argument as a file and we defined an array which contains all numbers between 0 and 9 to system. After we execute the reading file statement line by line, we check the required statements which controls whether the number is between 0 and 9. Furthermore, we defined to the system a '*' sign so that this program can execute the printing histogram operation and we defined an error part for the case where the numbers are not between 0 and 9 and printing operation for this. Finally, after the program reads file, prints the histogram requested from us.

```
acun@acun-VirtualBox:~/Masaüstü$ ./ex1.sh file.txt
0
1 **
2 ***
3
4 *
5 **
6 ***
7 *
8 *
9
acun@acun-VirtualBox:~/Masaüstü$
```

```
acun@acun-VirtualBox:~/Masaüstü$ ./ex1.sh file.txt
There is invalid input: -1
There is invalid input: 10
0
1 **
2 ***
3
4 *
5 **
6 ***
7 *
8 *
9
acun@acun-VirtualBox:~/Masaüstü$
```

- 2) Firstly, we take a command line arguments as a word and number and we defined an array which contains all letters in the alphabet. After these definitions, we split word and number character by character and we stored the result of this splitting operation in arrays one by one. Furthermore, we defined required informations to system to hold and store length of word, number and alphabet and we defined a string variable which holds and stores the encrypted word and called as final.

We use if statements to avoid occuring an error and controlling all operations such as checking length of word and number. In first if part, we check the length of word and number and if these are equal, the program will calculate encrypted word character by character and print the result by using final keyword we defined previously. In second if part, we check length of number and if it is equal to 1, again, the program will calculate encrypted word character by character and print the result by using final keyword we defined previously for just one number. As Finally, we defined required informations to the system so that it can print error message for error state which is the state of the length of the number is not equals to length of word or 1.

```
acun@acun-VirtualBox:~/Masaüstü$ ./ex2.sh apple 12345
brspj
acun@acun-VirtualBox:~/Masaüstü$ ./ex2.sh zoo 8
hww
acun@acun-VirtualBox:~/Masaüstü$ ./ex2.sh apple 12
Numbers length should be equals to word length or 1.
acun@acun-VirtualBox:~/Masaüstü$
```

- 3) We defined the problem two system by 2 part. If there is a command line argument, first part is executed but if there is not a command line argument, second part is executed.

In first part, we changed the directory as a command line argument and listed all files according to modification time by using usefull commands and took the oldest one. Furthermore, we used '-i' to delete oldest one by asking to the user.

In second part, by this time, we do not changed the directory but our all operations are the same except this. If we want to mention these again, these operations are listing all files according to modification time by using usefull commands, taking the oldest one and using '-i' to delete oldest one by asking to the user.

```
acun@acun-VirtualBox:~/Masaüstü/example3$ ls -l
toplama 16
drwxrwxr-x 2 acun acun 4096 Kas 20 12:08 directory1
-rwxr-xr-x 1 acun acun 613 Kas 20 12:10 ex3.sh
-rw-rw-r-- 1 acun acun 9 Kas 20 12:07 file3_1.txt
-rw-rw-r-- 1 acun acun 7 Kas 20 12:07 file3_2.txt
acun@acun-VirtualBox:~/Masaüstü/example3$ ls -l directory1
toplama 12
-rw-rw-r-- 1 acun acun 10 Kas 14 16:37 file4.txt
-rw-rw-r-- 1 acun acun 17 Kas 14 16:37 file5.txt
-rw-rw-r-- 1 acun acun 5 Kas 20 12:08 file6.txt
acun@acun-VirtualBox:~/Masaüstü/example3$ ./ex3.sh
rm: normal dosya 'file3_1.txt' silinsin mi? n
acun@acun-VirtualBox:~/Masaüstü/example3$ ./ex3.sh directory1
rm: normal dosya 'file4.txt' silinsin mi? n
acun@acun-VirtualBox:~/Masaüstü/example3$
```

- 4) Firstly, we took a command line argument as a file and defined text variable as a empty string to store final text. After reading the file character by character, we defined if statements to the system so that the program can change the number to word version and add to text variable, if the number is detected. If there is blank, this program adds blank character to text variable by using '-z' keyword. Furthermore, for the other characters, it adds the character directly to text variable. After execute these all operations, it deletes the context of the txt file and prints final text which we defined previously to the system into the txt file.

```
acun@acun-VirtualBox:~/Masaüstü$ cat file4.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit. 7 Suspendisse vitae odio blandit, commodo
nisl dignissim, 9 commodo est. Quisque blandit laoreet ante id tincidunt. Vivamus in vestibulum s
em. Duis ac faucibus quam. Mauris posuere, sapien quis elementum porttitor, leo turpis finibus era
t, vel dapibus 00 lorem mauris in elit. Curabitur quis massa sit amet ligula suscipit pulvinar.
acun@acun-VirtualBox:~/Masaüstü$ ./ex4.sh file4.txt
Context of file4.txt is changed successfully
acun@acun-VirtualBox:~/Masaüstü$ cat file4.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit. seven Suspendisse vitae odio blandit, com
modo nisl dignissim, nine commodo est. Quisque blandit laoreet ante id tincidunt. Vivamus in vesti
bulum sem. Duis ac faucibus quam. Mauris posuere, sapien quis elementum porttitor, leo turpis fini
bus erat, vel dapibus zerozero lorem mauris in elit. Curabitur quis massa sit amet ligula suscipit
pulvinar.
acun@acun-VirtualBox:~/Masaüstü$
```


- 5) In this part, if there is one command line argument, this program will create a new directory which named as copied and copies the given file into the copied file which created firstly by the program.

If there are two command line argument and the first one of this command line is equal to “-R”, this program does the copying operation recursively.

Firstly, by using find keyword, all subdirectories are stored in an array and by using for loop, this program will create new directory and copies the files in this new directories for all subdirectories.

```
acun@acun-VirtualBox:~/Masaüstü/example5$ ls
cask-of-amontillado.txt  french.txt  trees-and-other-poems.txt
ex5.sh                  shakespeare
acun@acun-VirtualBox:~/Masaüstü/example5$ ls shakespeare
barleby-scrivener.txt  calaveras-county.txt
acun@acun-VirtualBox:~/Masaüstü/example5$ ./ex5.sh -R "c*.txt"
acun@acun-VirtualBox:~/Masaüstü/example5$ ls copied
cask-of-amontillado.txt
acun@acun-VirtualBox:~/Masaüstü/example5$ ls shakespeare
barleby-scrivener.txt  calaveras-county.txt  copied
acun@acun-VirtualBox:~/Masaüstü/example5$ ls shakespeare/copied
calaveras-county.txt
acun@acun-VirtualBox:~/Masaüstü/example5$
```