

**Linux & bash script**  
**Book– Exam #1**  
**Bash command Interpreter**

Instructor: Fernando Crozetta

November 16, 2023

**Name:** \_\_\_\_\_  
**Email:** \_\_\_\_\_

---

This exam contains 8 pages (including this cover) and 3 questions. The total of points is 50.  
Good luck.

**Points distribution**

Questions	Points	Grade
1	15	
2	13	
3	22	
Total:	50	

1. In relation to the Operational system basic commands, answer:

(a) (1 point) How can we list all the contents of a file?

**Solution:**

```
cat file.txt
```

(b) (1 point) How to list the contents of a file, from the last line to the first line?

**Solution:**

```
tac arquivo.txt
```

(c) (1 point) How can we list the contents of a file, in an ordered manner?

**Solution:**

```
cat arquivo | sort
```

(d) (1 point) What is the command used to turn off the computer?

**Solution:**

```
halt
```

(e) (1 point) What is the command used to restart the computer?

**Solution:**

```
reboot
```

(f) (1 point) What is the function of the command *cal*?

**Solution:**

display a calendar in the screen

(g) (1 point) What is the function of the command *date*?

**Solution:**

Show the date of the system. the command also allow us to modify the system date.

- (h) (1 point) What is the command line we have to execute in order to see the manual of the command *date*?

**Solution:**

```
man date
```

- (i) (1 point) What is the purpose of *head* and *tail* commands ? What is the difference between them?

**Solution:**

*head*: show the first lines of a file.

*tail*: show the last lines of a file.

Both are used to filter a specified number of lines of the file.

- (j) (1 point) How can we filter the lines of a file that contains a specified word?

**Solution:**

```
grep "word" file.txt
```

- (k) (1 point) What is the difference between *more* and *less*?

**Solution:**

*more*: Makes a simple pagination, where *<enter>* moves one line forward, and *<space>* moves one page forward

*less*: Does the same as *more*, however it allows the use of arrow keys to roll forward/backward in the page. It also allows search in the file.

- (l) (1 point) Considering *A.txt* and *B.txt* two files, how can we check the difference in their contents?

**Solution:**

```
diff A.txt B.txt
```

- (m) (1 point) Which commands makes a pause, before moving into the next instruction?

**Solution:**

```
sleep and usleep.
```

- (n) (1 point) Which command splits a line of text, delimited by a character?

**Solution:**

cut

- (o) (1 point) Which command is used to make text replacements in files?

**Solution:**

sed

2. Considering the structure of *etc/passwd* below, where the fields are separated by colon (:)

0	1	2	3	4	5	6
<i>user</i>	<i>x</i>	<i>UID</i>	<i>GID</i>	<i>Extra</i>	<i>Home</i>	<i>Shell</i>

Answer:

- (a) (1 point) What is the command to list the entire file?

**Solution:**

```
cat /etc/passwd
```

- (b) (2 points) What is the command line to list only the usernames?

**Solution:**

```
cat /etc/passwd | cut -d':' -f1
```

- (c) (2 points) What is the command line to list only *UID* and *home* of each user?

**Solution:**

```
cat /etc/passwd | cut -d':' -f1,6
```

- (d) (2 points) What is the command line to print only the three first characters of each user's *home* ?

**Solution:**

```
cat /etc/passwd | cut -d':' -f6 | cut -c1-3
```

- (e) (2 points) What is the command line to load only *user*, *extra* from each user, replacing the field *extra* for "NOT INFORMED", if it is empty, or contains only commas?

**Solution:**

```
sed -r 's/(: *)/NOT INFORMED' | cut -d':' -f1,5
```

- (f) (2 points) What is the command line to check how many users are registered in the file?

**Solution:**

```
cat /etc/passwd | wc -l
```

- (g) (2 points) What is the command line to show all users that start with consonants, and their respective *home* folders, alphabetically ordered?

**Solution:**

```
grep -e '^[^aeiouAEIOU]' | cut -d':' -f1,6
```

3. (22 points) You received a file with the registered users to import in your system. The file, however, is not in a valid format. Your job is to remove the invalid entries that could potentially break your system. Create a valid output, considering the following:
- (a) The header line in the input must be removed.
  - (b) all usernames must be lowercase.
  - (c) All emails must have a user with more than 3 characters.  
(*Ex:* 1234@example.com)
  - (d) All emails must end with *.com.br* or *.com*
  - (e) Only active users must be parsed.

Below, you can see the first lines of this file:

```
user,email,active
usuario1,user1@mycompany.com,true
usuario2,user2@mycompany.net,true
usuario3,3@mycompany.com.br,true
usuario4,user4@mycompany.com,false
USUARIO,user4@mycompany.com,false
falseUser,falseUser@company.com,true
trueUser,trueUser@company.com,false
usuario5,usuariojoao@company.com.br,true
usuario6,usr@company.com,true
usuario7,usuario@minicompany.net.br,true
```

**Solution:**

```
sed -r '1d' file.txt |
sed -r '[A-Z],/d' |
sed -r ',[a-Z0-9]{1,3}@/d' |
sed -r '(\.com|\.com\.br),/d' |
sed -r ',false$/d'
```

This page can be used to draft your answers