0x04. Loops, conditions and parsing

DevOpsShellBashScripting

- By: Sylvain Kalache
- Weight: 1
- Project over - took place from Jul 21, 2022 6:00 AM to Jul 22, 2022 6:00 AM
- An auto review will be launched at the deadline

In a nutshell…

- **Auto QA review:** 41.5/83 mandatory & 15.0/30 optional
- **Altogether: 75.0%**
  - Mandatory: 50.0%
  - Optional: 50.0%
  - Calculation: 50.0% + (50.0% * 50.0%) == **75.0%**

About Bash projects

Unless stated, all your projects will be auto-corrected with Ubuntu 20.04 LTS.

Background Context

Resources

Read or watch:

- Loops sample
- Variable assignment and arithmetic
- Comparison operators
- File test operators
- Make your scripts portable

man or help:

- `env`
- `cut`
- `for`
- `while`
- `until`
- `if`

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone, **without the help of Google**:

General

- How to create SSH keys
- What is the advantage of using `#!/usr/bin/env bash` over `#!/bin/bash`
- How to use `while`, `until` and `for` loops
- How to use `if`, `else`, `elif` and `case` condition statements
- How to use the `cut` command
- What are files and other comparison operators, and how to use them

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted on Ubuntu 20.04 LTS
- All your files should end with a new line
- A `README.md` file, at the root of the folder of the project, is mandatory
- All your Bash script files must be executable
- You are not allowed to use `awk`
- Your Bash script must pass `Shellcheck` (version `0.7.0`) without any error
- The first line of all your Bash scripts should be exactly `#!/usr/bin/env bash`
- The second line of all your Bash scripts should be a comment explaining what is the script doing

Copyright - Plagiarism

- You are tasked to come up with solutions for the tasks below yourself to meet with the above learning objectives.
- You will not be able to meet the objectives of this or any following project by copying and pasting someone else's work.
- You are not allowed to publish any content of this project.
- Any form of plagiarism is strictly forbidden and will result in removal from the program.

More Info

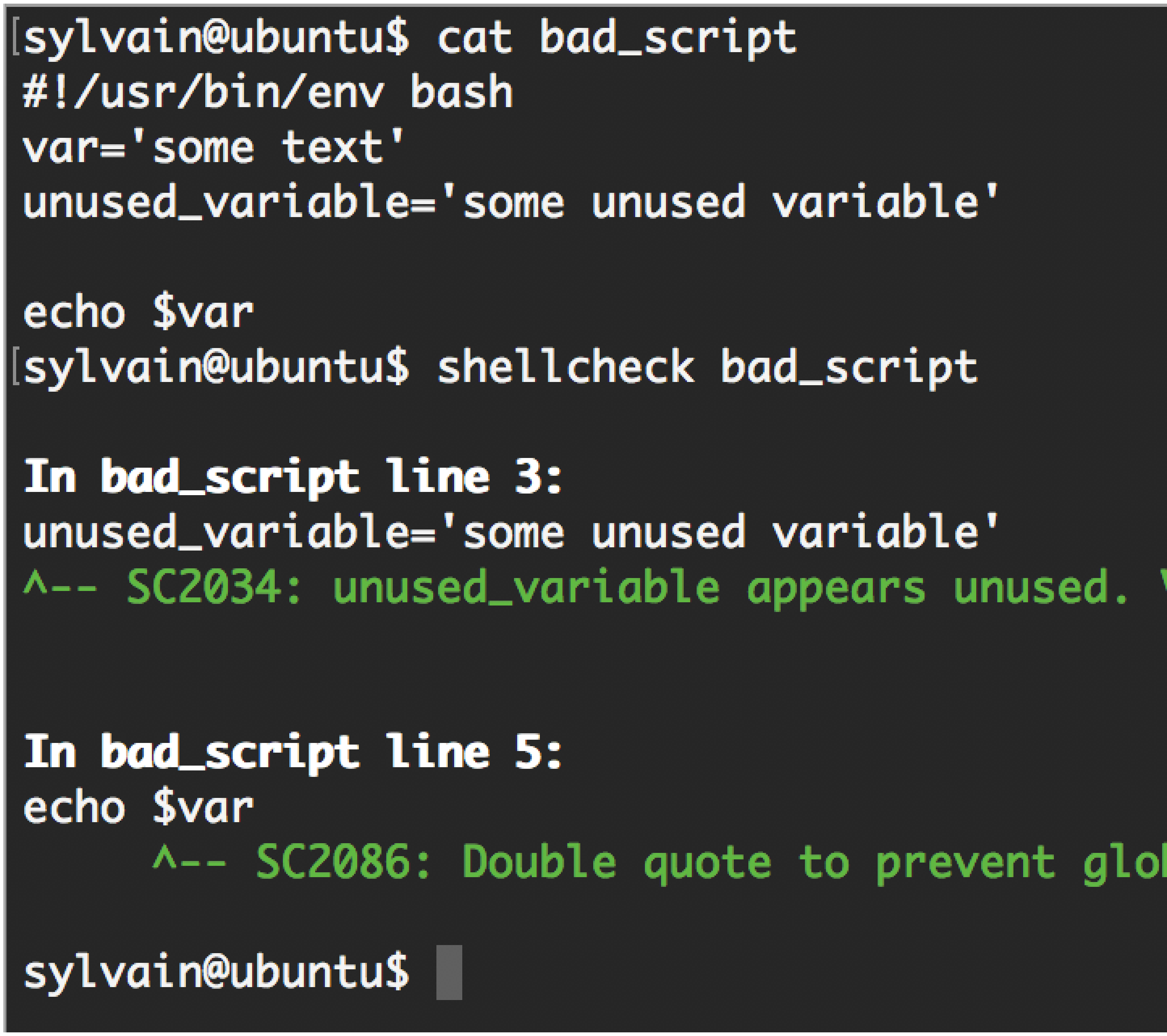
Shellcheck

Shellcheck is a tool that will help you write proper Bash scripts. It will make recommendations on your syntax and semantics and provide advice on edge cases that you might not have thought about. `Shellcheck` is your friend! **All your Bash scripts must pass `Shellcheck` without any error or you will not get any points on the task**.

`Shellcheck` is available on the school's computers. If you want to use it on your own computer, here is how to install it.

Examples:

Not passing `Shellcheck`:



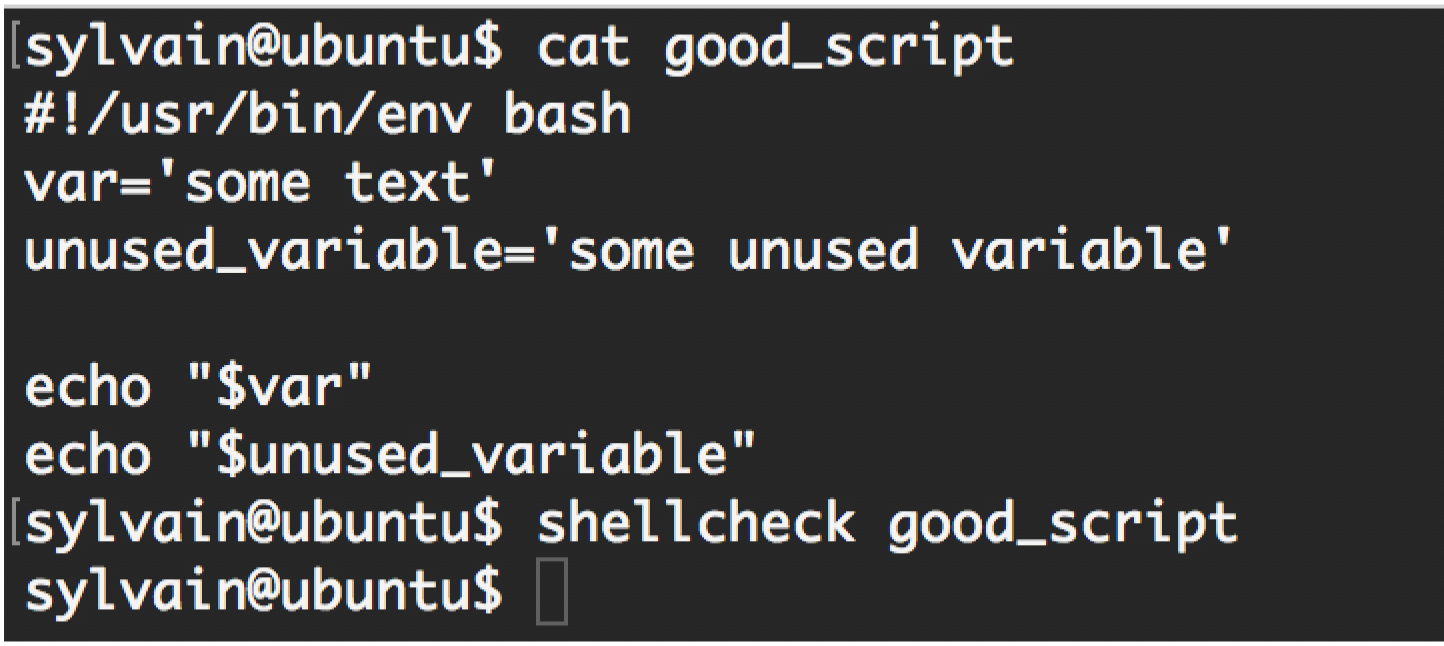
```
sylvain@ubuntu$ cat bad_script
#!/usr/bin/env bash
var='some text'
unused_variable='some unused variable'

echo $var
sylvain@ubuntu$ shellcheck bad_script

In bad_script line 3:
unused_variable='some unused variable'
^-- SC2034: unused_variable appears unused.

In bad_script line 5:
echo $var
     ^-- SC2086: Double quote to prevent glo
sylvain@ubuntu$
```

Passing `Shellcheck`:



```
[sylvain@ubuntu$ cat good_script
#!/usr/bin/env bash
var='some text'
unused_variable='some unused variable'

echo "$var"
echo "$unused_variable"
[sylvain@ubuntu$ shellcheck good_script
sylvain@ubuntu$ 
```

For every feedback, Shellcheck will provide a code that you can use to get more information about the issue, for example for code `SC2034`, you can browse https://github.com/koalaman/shellcheck/wiki/SC2034.

Tasks

0. Create a SSH RSA key pair

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Read for this task:

- Linux and Mac OS users
- Windows users

man: `ssh-keygen`

You will soon have to manage your own **servers** concept page hosted on remote data centers. We need to set them up with your RSA public key so that you can access them via SSH.

Create a RSA key pair.

Requirements:

- Share your **public key** in your answer file `0-RSA_public_key.pub`
- Fill the `SSH public key` field of your intranet profile with the public key you just generated

- **Keep the private key to yourself in a secure location**, you will use it later to connect to your servers using SSH. Some storing ideas are Dropbox, Google Drive, password manager, USB key. Failing to do so will prevent you to access your servers, which will prevent you from doing your projects
- If you decide to add a passphrase to your key, make sure to save this passphrase in a secure location, you will not be able to use your keys without the passphrase

SSH and RSA keys will be covered in depth in a later project.

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `0-RSA_public_key.pub`

Done! Help Check your code Get a sandbox QA Review

1. For Best School loop

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that displays `Best School` 10 times.

Requirement:

- You must use the `for` loop (`while` and `until` are forbidden)

```
sylvain@ubuntu$ head -n 2 1-for_best_school
#!/usr/bin/env bash
# This script is displaying "Best School" 10 times
sylvain@ubuntu$ ./1-for_best_school
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
```

```
sylvain@ubuntu$
```

Note that:

- The first line of my Bash script starts with `#!/usr/bin/env bash`
- The second line of my Bash scripts is a comment explaining what it is doing

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `1-for_best_school`

Done! Help Check your code Get a sandbox QA Review

2. While Best School loop

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that displays `Best School` 10 times.

Requirements:

- You must use the `while` loop (`for` and `until` are forbidden)

```
sylvain@ubuntu$ ./2-while_best_school
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`

- File: `2-while_best_school`

Done! Help Check your code Get a sandbox QA Review

3. Until Best School loop

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that displays `Best School` 10 times.

Requirements:

- You must use the `until` loop (`for` and `while` are forbidden)

```
sylvain@ubuntu$ ./3-until_best_school
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `3-until_best_school`

Done! Help Check your code Get a sandbox QA Review

4. If 9, say Hi!

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that displays `Best School` 10 times, but for the 9th iteration, displays `Best School` *and then* `Hi` on a new line.

Requirements:

- You must use the `while` loop (`for` and `until` are forbidden)
- You must use the `if` statement

```
sylvain@ubuntu$ ./4-if_9_say_hi
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Hi
Best School
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `4-if_9_say_hi`

Done! Help Check your code Get a sandbox QA Review

5. 4 bad luck, 8 is your chance

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that loops from 1 to 10 and:

- displays `bad luck` for the 4th loop iteration
- displays `good luck` for the 8th loop iteration
- displays `Best School` for the other iterations

Requirements:

- You must use the `while` loop (`for` and `until` are forbidden)
- You must use the `if`, `elif` and `else` statements

```
sylvain@ubuntu$ ./5-4_bad_luck_8_is_your_chance
```

```
Best School
Best School
Best School
bad luck
Best School
Best School
Best School
good luck
Best School
Best School
sylvain@ubuntu$
```

For the most curious:

- 8 in the Chinese culture
- 4 in the Chinese culture

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `5-4_bad_luck_8_is_your_chance`

Done! Help Check your code Get a sandbox QA Review

6. Superstitious numbers

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that displays numbers from 1 to 20 and:

- displays `4` *and then* `bad luck from China` for the 4th loop iteration
- displays `9` *and then* `bad luck from Japan` for the 9th loop iteration
- displays `17` *and then* `bad luck from Italy` for the 17th loop iteration

Requirements:

- You must use the `while` loop (`for` and `until` are forbidden)
- You must use the `case` statement

```
sylvain@ubuntu$ ./6-superstitious_numbers
1
```

```
2
3
4
bad luck from China
5
6
7
8
9
bad luck from Japan
10
11
12
13
14
15
16
17
bad luck from Italy
18
19
20
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `6-superstitious_numbers`

Done! Help Check your code Get a sandbox QA Review

7. Clock

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that displays the time for 12 hours and 59 minutes:

- display hours from 0 to 12
- display minutes from 1 to 59

Requirements:

- You must use the `while` loop (`for` and `until` are forbidden)

Note that in this example, we only display the first 70 lines using the `head` command.

```
sylvain@ubuntu$ ./7-clock | head -n 70

Hour: 0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25
```

```
59
Hour: 1
1
2
3
4
5
6
7
8
9
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `7-clock`

Done! Help Check your code Get a sandbox QA Review

8. For ls

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that displays:

- The content of the current directory
- In a list format
- Where only the part of the name after the first dash is displayed (refer to the example)

Requirements:

- You must use the `for` loop (`while` and `until` are forbidden)
- Do not display hidden files

```
sylvain@ubuntu$ ls
100-read_and_cut              1-for_best_school        6-superstitious_numbers
101-tell_the_story_of_passwd  2-while_best_school      7-clock
102-lets_parse_apache_logs    3-until_best_school      8-for_ls
```

```
103-dig_the-data              4-if_9_say_hi                9-to_file_or_not_to_file
10-fizzbuzz                   5-4_bad_luck_8_is_your_chance
sylvain@ubuntu$  ./8-for_ls
read_and_cut
tell_the_story_of_passwd
lets_parse_apache_logs
dig_the-data
fizzbuzz
for_best_school
while_best_school
until_best_school
if_9_say_hi
4_bad_luck_8_is_your_chance
superstitious_numbers
clock
for_ls
to_file_or_not_to_file
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `8-for_ls`

Done! Help Check your code Get a sandbox QA Review

9. To file, or not to file

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that gives you information about the `school` file.

Requirements:

- You must use `if` and, `else` (`case` is forbidden)
- Your Bash script should check if the file exists and print:
  - if the file exists: `school file exists`
  - if the file does not exist: `school file does not exist`
- If the file exists, print:

   - if the file is empty: `school file is empty`
   - if the file is not empty: `school file is not empty`
   - if the file is a regular file: `school is a regular file`
   - if the file is not a regular file: (nothing)

```
sylvain@ubuntu$ file school
school: cannot open `school' (No such file or directory)
sylvain@ubuntu$ ./9-to_file_or_not_to_file
school file does not exist
sylvain@ubuntu$ touch school
sylvain@ubuntu$ ./9-to_file_or_not_to_file
school file exists
school file is empty
school is a regular file
sylvain@ubuntu$ echo 'betty' > school
sylvain@ubuntu$ ./9-to_file_or_not_to_file
school file exists
school file is not empty
school is a regular file
sylvain@ubuntu$ rm school
sylvain@ubuntu$ mkdir school
sylvain@ubuntu$ ./9-to_file_or_not_to_file
school file exists
school file is not empty
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `9-to_file_or_not_to_file`

Done! Help Check your code Get a sandbox QA Review

10. FizzBuzz

mandatory

Score: 50.0% (*Checks completed: 100.0%*)

Write a Bash script that displays numbers from 1 to 100.

Requirements:

- Displays `FizzBuzz` when the number is a multiple of 3 and 5
- Displays `Fizz` when the number is multiple of 3
- Displays `Buzz` when the number is a multiple of 5
- Otherwise, displays the number
- In a list format

```
sylvain@ubuntu$ ./10-fizzbuzz | head -20
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `10-fizzbuzz`

Done! Help Check your code Get a sandbox QA Review

11. Read and cut

#advanced

Score: 50.0% (*Checks completed: 100.0%*)

help: `read`

Write a Bash script that displays the content of the file `/etc/passwd`.

Your script should only display:

- username
- user id
- Home directory path for the user

Requirements:

- You must use the `while` loop (`for` and `until` are forbidden)

```
sylvain@ubuntu$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
```

```
messagebus:x:102:106::/var/run/dbus:/bin/false
landscape:x:103:109::/var/lib/landscape:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
vagrant:x:1000:1000::/home/vagrant:/bin/bash
colord:x:106:112:colord colour management daemon,,,:/var/lib/colord:/bin/false
statd:x:107:65534::/var/lib/nfs:/bin/false
sylvain:98:99:Sylvain:/home/sylvain:/bin/bash
puppet:x:108:114:Puppet configuration management daemon,,,:/var/lib/puppet:/bin/false
ubuntu:x:1001:1001:Ubuntu:/home/ubuntu:/bin/bash
sylvain@ubuntu$ ./100-read_and_cut
root:0:/root
daemon:1:/usr/sbin
bin:2:/bin
sys:3:/dev
sync:4:/bin
games:5:/usr/games
man:6:/var/cache/man
lp:7:/var/spool/lpd
mail:8:/var/mail
news:9:/var/spool/news
uucp:10:/var/spool/uucp
proxy:13:/bin
www-data:33:/var/www
backup:34:/var/backups
list:38:/var/list
irc:39:/var/run/ircd
gnats:41:/var/lib/gnats
nobody:65534:/nonexistent
libuuid:100:/var/lib/libuuid
syslog:101:/home/syslog
messagebus:102:/var/run/dbus
landscape:103:/var/lib/landscape
```

```
sshd:104:/var/run/sshd
pollinate:105:/var/cache/pollinate
vagrant:1000:/home/vagrant
colord:106:/var/lib/colord
statd:107:/var/lib/nfs
sylvain:99:/bin/bash
puppet:108:/var/lib/puppet
ubuntu:1001:/home/ubuntu
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `100-read_and_cut`

Done! Help Check your code Get a sandbox QA Review

12. Tell the story of passwd

#advanced

Score: 50.0% (*Checks completed: 100.0%*)

Read:

- IFS (internal field separator)
- Understanding /etc/passwd

The file `/etc/passwd` has already been covered in a previous project and you should be familiar with it. Today we will make up a story based on it.

Write a Bash script that displays the content of the file `/etc/passwd`, using the `while` loop + IFS.

Format: `The user USERNAME is part of the GROUP_ID gang, lives in HOME_DIRECTORY and rides COMMAND/SHELL. USER ID's place is protected by the passcode PASSWORD, more info about the user here: USER ID INFO`

Requirements:

- You must use the `while` loop (`for` and `until` are forbidden)

```
sylvain@ubuntu$ ./101-tell_the_story_of_passwd
The user root is part of the 0 gang, lives in /root and rides /bin/bash. 0's place is
protected by the passcode x, more info about the user here: root
```

```
The user daemon is part of the 1 gang, lives in /usr/sbin and rides /usr/sbin/nologin
. 1's place is protected by the passcode x, more info about the user here: daemon

The user bin is part of the 2 gang, lives in /bin and rides /usr/sbin/nologin. 2's pl
ace is protected by the passcode x, more info about the user here: bin

The user sys is part of the 3 gang, lives in /dev and rides /usr/sbin/nologin. 3's pl
ace is protected by the passcode x, more info about the user here: sys

The user sync is part of the 65534 gang, lives in /bin and rides /bin/sync. 4's place
is protected by the passcode x, more info about the user here: sync

The user games is part of the 60 gang, lives in /usr/games and rides /usr/sbin/nologi
n. 5's place is protected by the passcode x, more info about the user here: games

The user man is part of the 12 gang, lives in /var/cache/man and rides /usr/sbin/nolo
gin. 6's place is protected by the passcode x, more info about the user here: man

The user lp is part of the 7 gang, lives in /var/spool/lpd and rides /usr/sbin/nologi
n. 7's place is protected by the passcode x, more info about the user here: lp

The user mail is part of the 8 gang, lives in /var/mail and rides /usr/sbin/nologin.
8's place is protected by the passcode x, more info about the user here: mail

The user news is part of the 9 gang, lives in /var/spool/news and rides /usr/sbin/nol
ogin. 9's place is protected by the passcode x, more info about the user here: news

The user uucp is part of the 10 gang, lives in /var/spool/uucp and rides /usr/sbin/no
login. 10's place is protected by the passcode x, more info about the user here: uucp

The user proxy is part of the 13 gang, lives in /bin and rides /usr/sbin/nologin. 13'
s place is protected by the passcode x, more info about the user here: proxy

The user www-data is part of the 33 gang, lives in /var/www and rides /usr/sbin/nolog
in. 33's place is protected by the passcode x, more info about the user here: www-dat
a

The user backup is part of the 34 gang, lives in /var/backups and rides /usr/sbin/nol
ogin. 34's place is protected by the passcode x, more info about the user here: backu
p

The user list is part of the 38 gang, lives in /var/list and rides /usr/sbin/nologin.
38's place is protected by the passcode x, more info about the user here: Mailing Lis
t Manager

The user irc is part of the 39 gang, lives in /var/run/ircd and rides /usr/sbin/nolog
in. 39's place is protected by the passcode x, more info about the user here: ircd

The user gnats is part of the 41 gang, lives in /var/lib/gnats and rides /usr/sbin/no
login. 41's place is protected by the passcode x, more info about the user here: Gnat
s Bug-Reporting System (admin)

The user nobody is part of the 65534 gang, lives in /nonexistent and rides /usr/sbin/
nologin. 65534's place is protected by the passcode x, more info about the user here:
nobody

The user libuuid is part of the 101 gang, lives in /var/lib/libuuid and rides . 100's
place is protected by the passcode x, more info about the user here:

The user syslog is part of the 104 gang, lives in /home/syslog and rides /bin/false.
101's place is protected by the passcode x, more info about the user here:
```

```
The user messagebus is part of the 106 gang, lives in /var/run/dbus and rides /bin/fa
lse. 102's place is protected by the passcode x, more info about the user here:

The user landscape is part of the 109 gang, lives in /var/lib/landscape and rides /bi
n/false. 103's place is protected by the passcode x, more info about the user here:

The user sshd is part of the 65534 gang, lives in /var/run/sshd and rides /usr/sbin/n
ologin. 104's place is protected by the passcode x, more info about the user here:

The user pollinate is part of the 1 gang, lives in /var/cache/pollinate and rides /bi
n/false. 105's place is protected by the passcode x, more info about the user here:

The user vagrant is part of the 1000 gang, lives in /home/vagrant and rides /bin/bash
. 1000's place is protected by the passcode x, more info about the user here:

The user colord is part of the 112 gang, lives in /var/lib/colord and rides /bin/fals
e. 106's place is protected by the passcode x, more info about the user here: colord
colour management daemon,,,

The user statd is part of the 65534 gang, lives in /var/lib/nfs and rides /bin/false.
107's place is protected by the passcode x, more info about the user here:

The user puppet is part of the 114 gang, lives in /var/lib/puppet and rides /bin/fals
e. 108's place is protected by the passcode x, more info about the user here: Puppet
configuration management daemon,,,

The user ubuntu is part of the 1001 gang, lives in /home/ubuntu and rides /bin/bash.
1001's place is protected by the passcode x, more info about the user here: Ubuntu

sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `101-tell_the_story_of_passwd`

Done! Help Check your code Get a sandbox QA Review

13. Let's parse Apache logs

#advanced

Score: 50.0% (*Checks completed: 100.0%*)

Apache is among the most popular web servers in the world, serving 50% of all active websites, no doubt that you will have to interact with it within your career.

As a Full-Stack Software Engineer, you have to master the art of parsing log files. Today we will do a simple parsing of Apache log access files.

Today the Customer Support department reported that a user reported that the site is being “buggy”. Not being a detailed description, you want to have a look at the Apache logs and gather data about the traffic.

Write a Bash script that displays the visitor IP along with the HTTP status code from the Apache log file.

Requirement:

- Format: IP HTTP_CODE
  - in a list format
  - See example
- You must use `awk`
- You are not allowed to use `while`, `for`, `until` and `cut`
- Download and commit the apache-access.log file along with your answers files

```
sylvain@ubuntu$ ./102-lets_parse_apache_logs | tail -n 10
185.130.5.207 301
185.130.5.207 301
91.224.140.223 200
```

```
62.210.142.23 304
92.222.20.166 304
180.76.15.19 200
2.1.201.36 304
198.58.99.82 304
50.116.30.23 304
209.133.111.211 200
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `102-lets_parse_apache_logs`

Done! Help Check your code Get a sandbox QA Review

14. Dig the data

#advanced

Score: 50.0% (*Checks completed: 100.0%*)

Now that you've parsed the Apache log file, let's sort the data so you can get a better idea of what is going on.

Using what you did in the previous exercise, write a Bash script that groups visitors by IP and HTTP status code, and displays this data.

Requirements:

- The exact format must be:
  - OCCURENCE_NUMBER IP HTTP_CODE
  - In list format
- Ordered from the greatest to the lowest number of occurrences
  - See example
- You must use `awk`
- You are not allowed to use `while`, `for`, `until` and `cut`

```
sylvain@ubuntu$ ./103-dig_the-data | head -n 10
    141 130.0.236.153 200
    140 62.210.250.66 200
    117 103.243.26.232 404
    67 195.154.172.143 200
    60 78.154.190.29 200
    45 195.154.172.59 200
    41 62.210.248.185 200
    41 167.114.156.198 200
    36 2.1.201.36 304
    36 195.154.172.53 200
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `103-dig_the-data`

Done! Help Check your code Get a sandbox QA Review