

## **Lab 5:** awk and pipe operator

**Due date: 10/19/2021**

In this lab assignment we will be practicing with **awk**. Some of the features used in this assignment are explained in the textbook and slides but there might be a few things that you will need to learn from other sources. For additional information, you can see web sites

[https://spoken-tutorial.org/tutorial-search/?search\\_foss=Linux+A WK&search\\_language=English](https://spoken-tutorial.org/tutorial-search/?search_foss=Linux+A WK&search_language=English) and <http://www.zytrax.com/tech/web/regex.htm> among many others.

## List of Tasks

For this lab assignment you will create 4 awk scripts. The first three will be entered in a shell script file named **script1\_tXX** ( XX are the 2 digits corresponding to your team number) that you will use to perform tasks 1 through 3. The commands must be entered in full (I will execute your scripts to test them). The last awk script will be entered in a file named **script2\_tXX**.

To begin, get into your home directory and copy file `/files/labs/lab5/lab5.tar` into it. Remember a 'tar' file is an archive of many files and directories. In this case, I have created a directory structure for you to do some exercises in. To expand the archive, type the command **`tar -xvf lab5.tar`**. List the contents of the current directory. You should see a directory `lab5` with several files and directories in it.

For the first three questions you must use the pipe operator to redirect the output of the `ls -l` command (use more options for the command if necessary) to the input of your `awk` scripts.

For example: `ls -l | awk ' <your awk script here> '`

1. Write an awk script to print just the name and size of **ordinary** files (do not include directories), one on each line. The output must be aligned as shown in the example below. First, analyze a typical output of the `ls -l` command to see what differentiates an ordinary file from a directory and which fields contain the file name and size information. Then, determine what the field separator is and whether the default field separator of awk will work, or if you have to specify it using the `FS=` command. Insert this awk script in **script1.tXX** (preceded by `ls -l |`) and save it.

awk4	178
awk5	68
pr1.c	368
pr2.c	234
qwbre	47

2. In directory lab5 there are some hidden files and directories. Write an awk script to print just the name and size of **ordinary hidden** files (do not include directories), one on each line. Use a logical AND (&&) to achieve this effect. Only after both conditions are satisfied, proceed to print the information. Add to your listing a header like the one shown below. Add this awk script (preceded by `ls -l |`) to the content of **script1 tXX** and save it.

Name	Size
------	------

```
.321up      0
.awk2       91
.awk4       178
```

3. Now you want to list your C programs only (files with extension .c). Do not include hidden files in your listing. Add the files' access rights to your output as shown in the example below. Add this awk script (preceded by `ls -l |`) to the content of **script1\_tXX** and save it.

Name	Size	Access
pr1.c	368	-rw-r--r--
pr2.c	234	-rw-r--r--

Edit **script1\_tXX** and add the necessary shell commands so that when it is executed its output looks like Figure 1 shown below. Pay attention to the alignment of the displayed information.

```
===== Script 1A =====
```

```
awk4      178
awk5       68
pr1.c     368
pr2.c     234
qwbre      47
```

```
===== Script 1B =====
```

```
Name      Size
.321up      0
.awk2       91
.awk4       178
```

```
===== Script 1C =====
```

```
Name      Size  Access
pr1.c     368  -rw-r--r--
pr2.c     234  -rw-r--r--
```

```
=====
```

Test this script using: `./script1_tXX`

4. Write an awk script that finds the averages of lists of numbers. For example, see the following list:

```
10
200
3000
First separator
4
5
=====
```

Your awk script, when given the above input, should output:

```
===== Script 2 =====
```

```
10
200
3000
Average 1070.00
4
5
Average 4.50
=====
```

**Important:** the set of numbers will be separated one from another by lines containing any characters but digits (analyze the two input files provided please).

Insert your awk script in **script2\_tXX**.

Copy the input files named **lab5input1** and **lab5input2** from **/files/labs/lab5/** to your directory to test your scripts using:

```
awk -f script2_tXX lab5input1
```

and

```
awk -f script2_tXX lab5input2
```

***When done, put both scripts in a compressed folder named lab5\_2344\_tXX.zip and submit it through Blackboard using the “Assignments” tool. Do Not email it.***

**IMPORTANT:**

- 1) Make sure your script file names do NOT have any extension
- 2) Your compressed folder should have ONLY the two files requested (they MUST NOT be inside of a folder)
- 3) Make sure you specify your team number with TWO digits
- 4) Copy files named script1\_t00\_example and script2\_t00\_example from ~kpv468/labs/lab5/ to your directory to see what your scripts should look like