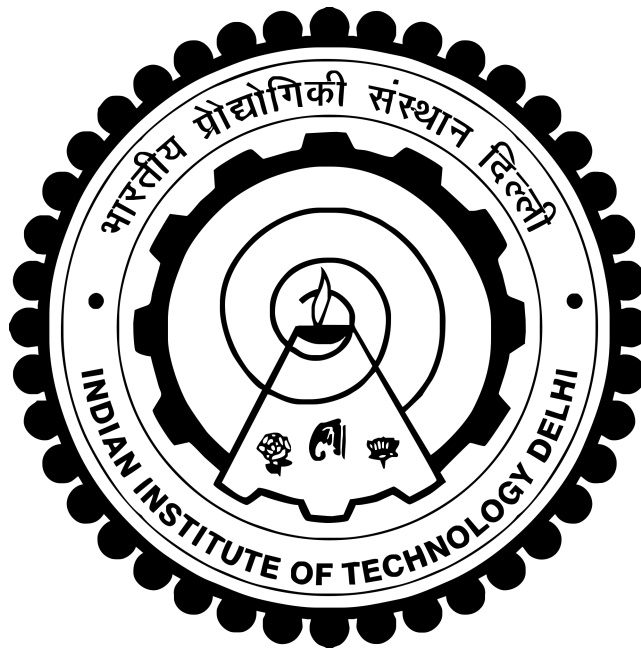


ELP718 Telecom Software Laboratory
1st Semester, 2016-18
Abhishek Mishra
23 Aug 2016, 5pm
Assignment-5



Contents

0.1	Introduction	3
0.2	Problem Statement 1	4
0.2.1	Assumptions	4
0.2.2	Structure Chart and Implementation	5
0.2.3	Screenshots	6
0.3	Problem Statement 2	7
0.3.1	Assumptions	7
0.3.2	Structure Chart	8
0.3.3	Screenshots	9
0.4	Problem Statement 3	10
0.4.1	Assumptions	10
0.4.2	Structure Chart	11
0.5	Epilogue	12

0.1 Introduction

This assignment aims to provide a better understanding of the following topics:

1. Shell Scripting

A shell script is a computer program designed to be run by the Unix shell, a command-line interpreter. The various dialects of shell scripts are considered to be scripting languages. The typical Unix/Linux/Posix-compliant installation includes the Korn Shell (ksh) in several possible versions such as ksh88, Korn Shell '93 and others.

The oldest shell still in common use is the Bourne shell (sh); Unix systems invariably include also the C Shell (csh), Bourne Again Shell (bash), a remote shell (rsh), a secure shell for SSL telnet connections (ssh), and a shell which is a main component of the Tcl/Tk installation usually called tclsh; wish is a GUI-based Tcl/Tk shell. The C and Tcl shells have syntax quite similar to that of said programming languages, and the Korn shells and Bash are developments of the Bourne shell, which is based on the ALGOL language with elements of a number of others added as well.

On the other hand, the various shells plus tools like awk, sed, grep, and BASIC, Lisp, C and so forth contributed to the Perl programming language

2. Vim Editor

Vim (Vi IMproved) is a clone of Bill Joy's vi text editor program for Unix. It was written by Bram Moolenaar based on source for a port of the Stevie editor to the Amiga and first released publicly in 1991.

Vim is designed for use both from a command-line interface and as a standalone application in a graphical user interface. Vim is free and open source software and is released under a license that includes some charityware clauses, encouraging users who enjoy the software to consider donating to children in Uganda. The license is compatible with the GNU General Public License.

0.2 Problem Statement 1

This problem requires us to take input through command line argument from the user and then process it similar to following outputs:

```
Change file extensions in folder mydir
$./lowercasefe.sh mydir
Number of file extensions changed -
```

```
Change file extensions in folder mydir and its sub folders recursively
$./lowercasefe.sh -r mydir
Number of file extensions changed -
```

```
Change file extensions in folder mydir, but folder name is wrong
$./lowercasefe.sh -r mydiir
There is no folder mydiir
```

```
Change file extensions in present working directory
$./lowercasefe.sh
Number of file extensions changed -
```

```
Change file extensions in present working directory and its sub folders
$./lowercasefe.sh
Number of file extensions changed -
```

0.2.1 Assumptions

The user shall input either no name to a directory or he will enter an existing directory.

0.2.2 Structure Chart and Implementation

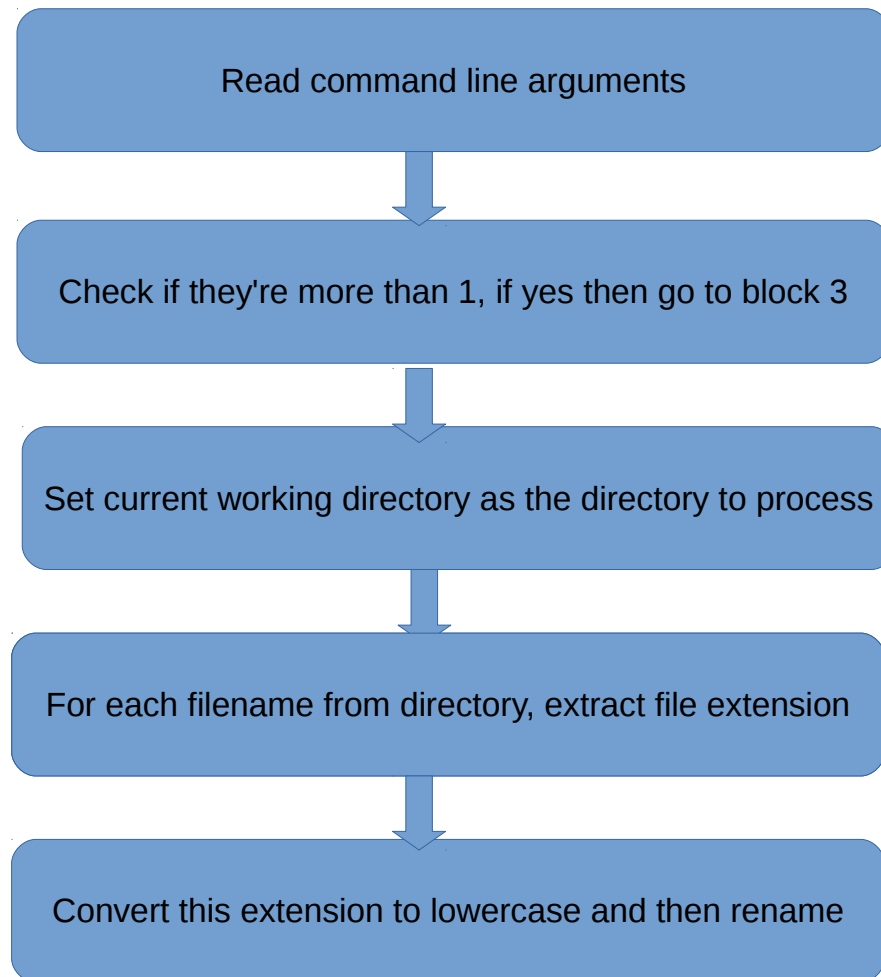


Figure 1: Structure chart for problem 1

0.2.3 Screenshots

```
dash
-eq 0 ] ; then
for filename in *
do
    file_ext="$($_CMD echo $filename |awk -F . '{if (NF>1) {print $NF}}')"
    n="$($_CMD echo $file_ext | tr A-Z a-z )"
    if [ "$file_ext" != "$n" ] ; then
        mv $filename `echo $filename | sed 's/\(.*\.$)file_ext /\1$
    fi
done
```

Figure 2: Screenshot for problem statement 1

0.3 Problem Statement 2

The problem requires you to display system information in the following manner.

```
./systeminfo.sh
===== System Information =====
Processor - Core i7
RAM - 4096 MB
Architecture - X86_64
MAC Address - 10:1e:23:da:12:c2
Linux Kernel - 3.13.0-93-generic
Ubuntu Release - 14.04.4
Username - upendra
Gcc version - 4.4.0
Python version - 2.7.3
=====
```

0.3.1 Assumptions

The output is to be processed in the exact manner as shown in the image

0.3.2 Structure Chart

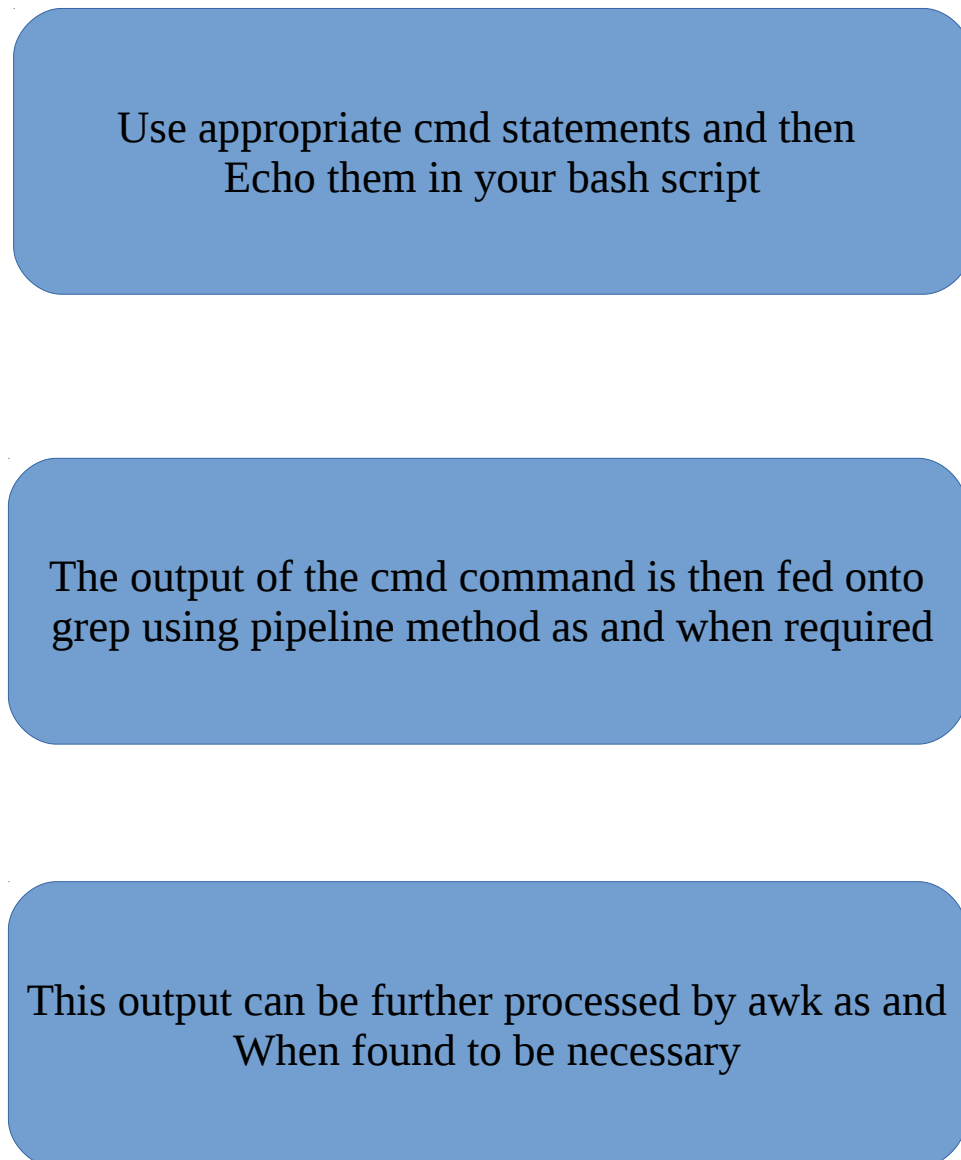


Figure 3: Structure chart for problem 2

0.3.3 Screenshots

```
abhisheks@administrator:~/bin$ lowercasefe.sh
===== System Information =====
Processor - Intel(R) Pentium(R) 4 CPU 3.06GHz
RAM - 3026 MB
Architecture: i686
MAC Address - f0:7d:68:fb:f0:32
Linux Kernel - 4.4.0-34-generic
Ubuntu Release - Ubuntu 14.04.5 LTS
Python version - Python 2.7.6
=====
```

Figure 4: Screenshot for problem statement 2

0.4 Problem Statement 3

Part - 1 -

Write shell script to emulate behaviour of bash, but also record every command put up on terminal on a logger.txt file along with time stamp. Your command prompt should look like normal bash command prompt.

0.4.1 Assumptions

The output is to be processed in the exact manner as shown in the image

```
upendra@admin108-OptiPlex-9020:~$ ./fakeshell.sh
upendra@admin108-OptiPlex-9020:~$ uname -m
x86_64
upendra@admin108-OptiPlex-9020:~$ cd test
upendra@admin108-OptiPlex-9020:~/test$ pwd
/home/upendra/test
upendra@admin108-OptiPlex-9020:~/test$ date
Fri Aug 19 15:42:48 IST 2016
upendra@admin108-OptiPlex-9020:~/test$ ls
File1.txt  file2.txt  file3.txt
upendra@admin108-OptiPlex-9020:~/test$ cp file3.txt ../myfolder/new
upendra@admin108-OptiPlex-9020:~/test$ mv File1.txt file1.txt
upendra@admin108-OptiPlex-9020:~/test$ ls
file1.txt  file2.txt
upendra@admin108-OptiPlex-9020:~/test$ exit_fakeshell
upendra@admin108-OptiPlex-9020:~/test$
```

0.4.2 Structure Chart

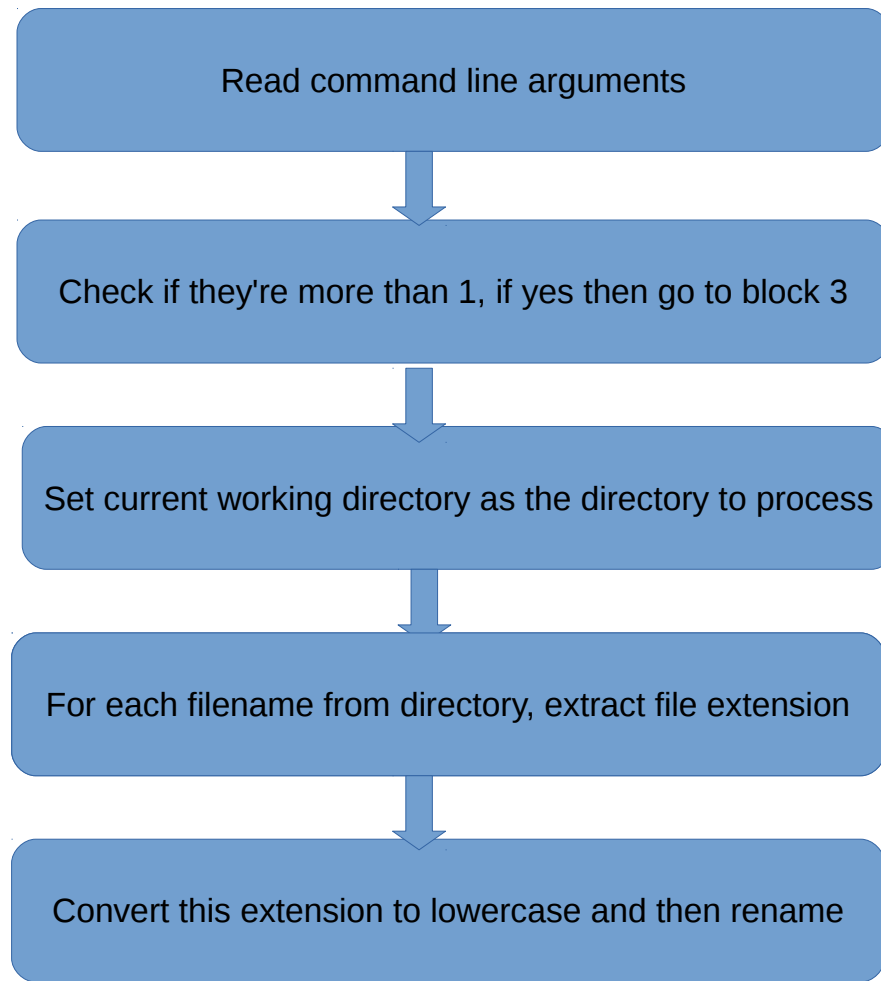


Figure 5: Structure chart for problem 3

0.5 Epilogue

The execution of the first problem involved me to break it down and then try to figure out solution to each and every part of it. To some extent I feel I have been successful in justifying the problem.

Whereas for problem statement 2, it was a different but not so difficult problem statement compared to problem 1.

Then i found the problem 3 to be the toughest of all problems.

This week's assignment too has taught me a lot of things on which I shall further improve upon in the next assignment.

Bibliography

- [1] “Heiner’s shelldorado.” <http://www.shelldorado.com/>.
- [2] “Advanced bash scripting guide.” <http://tldp.org/ldp/abs/html/>.