CS 3360: Bash Assignment

Kevin Apodaca

October 20, 2019

Secret Message

Using the files ’innocuous.txt’ and ’codebook.txt’ the secret message that I extracted was **they.mgiht.be.giants**

Report

Something clever that was used in this program was piping outputs to other functions which saves lines of code as well as having to save outputs to some temporary. This did not work in all instances, but was pretty useful such as when I had to remove the even numbers from the list. I used awk, piped its result into tr to remove the dashes and replace with spaces, then piped that output into cut so that I can put the columns together. Something else that was useful and clever is this use of the awk command

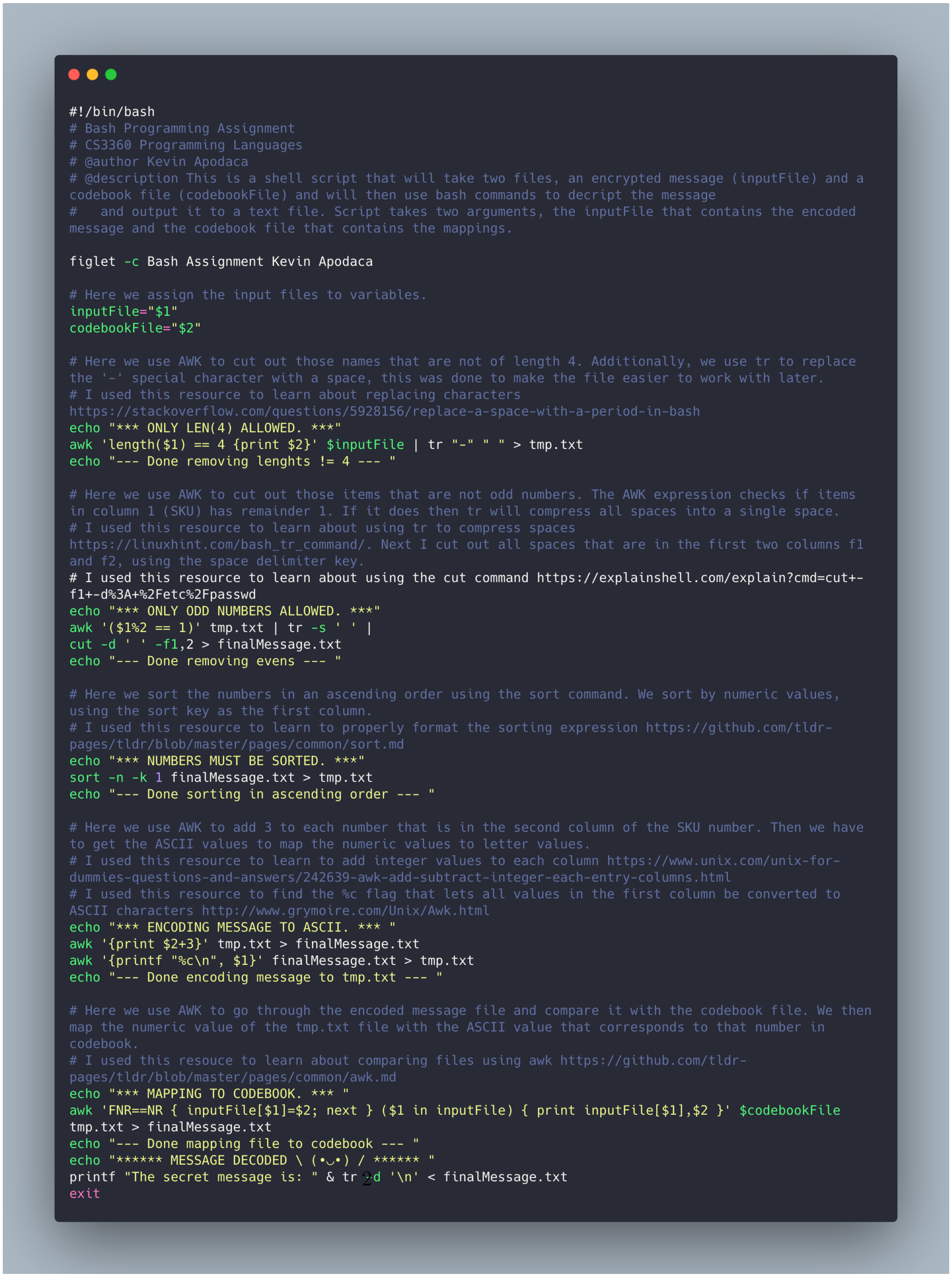
```
awk’{printf ”%c\n”, $1}’
```

The %c was used to map values to ASCII characters, making my job easier. Instead of hard-coding the names of the text files into the program, I decided to allow the user to feed arguments. The first argument is the file that contains the hidden message, and the second is going to be used as the codebook. This way the script can also work for the mini-fishlist and mini-codebook files. However, there is no checking that the files are the proper ones, meaning you can use codebook as the first argument and not get a valid result. There is also no checking that the files were supplied, so if just the shell script is executed then the shell will just be stuck infinitely until killed by the user. One last note, the line

```
figlet −c Bash Assignment Kevin Apodaca
```

Does not serve any purpose other than visual aesthetic, it uses the Figlet package that must be installed manually beforehand, therefore when running the script this line will produce a ’figlet: command not found’ error but it does NOT affect the functionality of this script in any way.

Bash Script



```
#!/bin/bash
# Bash Programming Assignment
# CS3360 Programming Languages
# @author Kevin Apodaca
# @description This is a shell script that will take two files, an encrypted message (inputFile) and a
codebook file (codebookFile) and will then use bash commands to decript the message
#   and output it to a text file. Script takes two arguments, the inputFile that contains the encoded
message and the codebook file that contains the mappings.

figlet -c Bash Assignment Kevin Apodaca

# Here we assign the input files to variables.
inputFile="$1"
codebookFile="$2"

# Here we use AWK to cut out those names that are not of length 4. Additionally, we use tr to replace
the '-' special character with a space, this was done to make the file easier to work with later.
# I used this resource to learn about replacing characters
https://stackoverflow.com/questions/5928156/replace-a-space-with-a-period-in-bash
echo "*** ONLY LEN(4) ALLOWED. ***"
awk 'length($1) == 4 {print $2}' $inputFile | tr "-" " " > tmp.txt
echo "--- Done removing lenghts != 4 --- "

# Here we use AWK to cut out those items that are not odd numbers. The AWK expression checks if items
in column 1 (SKU) has remainder 1. If it does then tr will compress all spaces into a single space.
# I used this resource to learn about using tr to compress spaces
https://linuxhint.com/bash_tr_command/. Next I cut out all spaces that are in the first two columns f1
and f2, using the space delimiter key.
# I used this resource to learn about using the cut command https://explainshell.com/explain?cmd=cut+-
f1+-d%3A+%2Fetc%2Fpasswd
echo "*** ONLY ODD NUMBERS ALLOWED. ***"
awk '($1%2 == 1)' tmp.txt | tr -s ' ' |
cut -d ' ' -f1,2 > finalMessage.txt
echo "--- Done removing evens --- "

# Here we sort the numbers in an ascending order using the sort command. We sort by numeric values,
using the sort key as the first column.
# I used this resource to learn to properly format the sorting expression https://github.com/tldr-
pages/tldr/blob/master/pages/common/sort.md
echo "*** NUMBERS MUST BE SORTED. ***"
sort -n -k 1 finalMessage.txt > tmp.txt
echo "--- Done sorting in ascending order --- "

# Here we use AWK to add 3 to each number that is in the second column of the SKU number. Then we have
to get the ASCII values to map the numeric values to letter values.
# I used this resource to learn to add integer values to each column https://www.unix.com/unix-for-
dummies-questions-and-answers/242639-awk-add-subtract-integer-each-entry-columns.html
# I used this resource to find the %c flag that lets all values in the first column be converted to
ASCII characters http://www.grymoire.com/Unix/Awk.html
echo "*** ENCODING MESSAGE TO ASCII. *** "
awk '{print $2+3}' tmp.txt > finalMessage.txt
awk '{printf "%c\n", $1}' finalMessage.txt > tmp.txt
echo "--- Done encoding message to tmp.txt --- "

# Here we use AWK to go through the encoded message file and compare it with the codebook file. We then
map the numeric value of the tmp.txt file with the ASCII value that corresponds to that number in
codebook.
# I used this resouce to learn about comparing files using awk https://github.com/tldr-
pages/tldr/blob/master/pages/common/awk.md
echo "*** MAPPING TO CODEBOOK. *** "
awk 'FNR==NR { inputFile[$1]=$2; next } ($1 in inputFile) { print inputFile[$1],$2 }' $codebookFile
tmp.txt > finalMessage.txt
echo "--- Done mapping file to codebook --- "
echo "****** MESSAGE DECODED \ (•◡•) / ****** "
printf "The secret message is: " & tr -d '\n' < finalMessage.txt
exit
```

Resources

https://linuxhint.com/bash_tr_command/
https://explainshell.com/explain?cmd=cut+-f1+-dhttps://github.com/tldr-pages/tldr/blob/master/pages/common/sort.md
https://www.unix.com/unix-for-dummies-questions-and-answers/242639-awk-add-subtract-integer-each-entry-columns.html
http://www.grymoire.com/Unix/Awk.html
https://github.com/tldr-pages/tldr/blob/master/pages/common/awk.md
https://stackoverflow.com/questions/32481877/what-is-nr-fnr-in-awk