**The Secret Message**

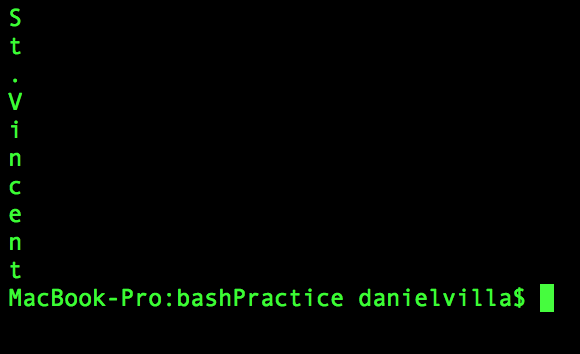


Figure Message is **St. Vincent**

**Bash Script**

#!/bin/bash

clear

#Removes the lines in logfile1.txt that contains the word server, memory, and syntax, store it

grep -v "server" logfile1.txt | grep -v "memory" | grep -v "syntax" > words\_gone.txt

#Removes the lines in logfile2.txt that contains the word server, memory, and syntax, store it

grep -v "server" logfile2.txt | grep -v "memory" | grep -v "syntax" >> words\_gone.txt

#Get the lines with error and the line after, then remove the lines with error

grep -A1 'error' words\_gone.txt | grep -v "error" > logfile\_code\_sentence.txt

#get only the first word in the line with a length of 2

awk 'length($1) == 2 { print $1 }' logfile\_code\_sentence.txt | grep -v "-" > codewords.txt

sort codewords.txt > codewordscompare.txt

#print out the lines of words in “mapping-sorted.txt” that match in “codewordscompare.txt”

awk 'NR==FNR{a[$1];next}$1 in a{print}' codewordscompare.txt mapping-sorted.txt > foundwords.txt

#sort the numbers in the third column by smallest to biggest

sort -n -k3 foundwords.txt | awk '{print $2}' > secretcode.txt

cat secretcode.txt

**Report**

Bash was difficult to understand. At a glance, the commands of bash look like gibberish. However, I eventually figured out how to use some of the functions of bash. ”Grep” was used to remove lines with the words “server”, “memory” and “syntax”. Then I used “grep” again to find the lines with error, get the line right after those lines then remove the lines with error. This gave me lines with the potential codes at the beginning. Next, I used “awk” to get the words at the beginning of each line with only a length of 2 letters and filtered out the line dashes. After sorting the potential code words in alphabetical order, I compared the code words I found to the ones found in “mapping-sorted.txt” to filter the valid code words. Finally, I sorted the code words by the numbers and got the secret message **St. Vincent**.