Assignment 1 - Computing Systems

Short introduction

Hello, my name is Dawid Szczesny (21300293), and I worked on this assignment with Brian Moyles.

BashBook version 1 is a terminal based, Facebook-like, client-run application where friends and family can add each other and post on each other's walls, all from just their shared computer.

Organizational Structure

Our organizational structure in this assignment was a partnership. We both got together and worked on the pseudo code, in order to plan out our approach of BashBook. We divided up the scripts needed to run the program amongst ourselves and helped each other whenever we got stuck.

In terms of the organizational structure of the program, we divided it into many little scripts, with one main script called server.sh which contained the while loop. Each script would then be run from the server.sh script according to what the user does.

Implementations

From the pseudo code, we came up with 5 scripts that were needed. We divided them equally and then whoever got their scripts done first could get a start on the 5th one.

I implemented the server.sh file first. I understood that it was the 'main method' of the program. I first added the while loop to loop over the commands that the user could enter, and then using a switch case syntax to determine what that command was.

```
while true do
  case $req in
    command1)
      # do something
      ;;
    command2)
      # do something else
      ;;
  esac
done
```

The main issue I faced when implementing this was trying to figure out how to intake a number of arguments. I eventually figured out the syntax for that in Bash, and realized that I needed a max of 4 arguments.

```
read req arg0 arg1 arg2 arg3
```

Another essential part of the program that I implemented later on was to check whether the user directory exists. This directory was essential for the program to work and hence should be checked on run, and create it if it doesn't exist.

```
if [[ ! -d "users" ]]; then
  mkdir "users"
fi
```

The second file I implemented was the creates.sh. The was a much less challenging script to write as all it entailed was to check if a directory of the user name existed and respond appropriately. I run this script directly from the server.sh using the source keyword

```
source ./creates.sh $arg0
```

It takes in one argument which is the name of the user.

I then decided to implement the display_wall.sh script, as it was another easy script to program. It takes in one argument which is the user name. We check if that user exists, and if it does we display their wall.txt.

Brian Moyles implemented the post_message.sh and add_friend.sh scripts. We figured they'd be quite similar files and would be best implemented by the one person.

After we both finished implementing our respective files, we came together and helped each other program the parts we done.

The post_message.sh script takes in 3 arguments. The first is the sender name, the second is the receiver name. The the third takes in all arguments after. This is because a message could involve spaces which would count as an extra argument.

```
msg=${@:3}
```

The add_friend.sh script is very similar to the post_message.sh script in terms of checking if the sender and receiver users both exist.

Adding friends is symmetrical.

Difficulties

- Bash syntax. Although it is similar enough to c and python syntax it is also very different and its syntax is more strict. (Has to be a space after the double square bracket in an if statement.)
- Passing arguments from console. I found it difficult at first to take in multiple variables from the users. I eventually figured it out with trial and error.

Conclusion

- Bash is a terminal based language that is useful to automate computer tasks. It takes some getting used to but after a while it isn’t too difficult to use.
- BashBook was an interesting assignment and definitely helped in learning the basics of bash. It also helped improve my knowledge of linux.
- BashBook was an easy to implement program, that helped improve my team-working abilities