Names->Vaibhav Saini (301386847) and Sehajvir Singh Pannu (301386534)

Assignment->2

Resources used->

https://www.geeksforgeeks.org/making-linux-shell-c/

https://www.geeksforgeeks.org/strdup-strdndup-functions-c/

https://stackoverflow.com/questions/2521927/initializing-a-global-struct-in-c

https://danishpraka.sh/2018/01/15/write-a-shell.html

https://solarianprogrammer.com/2019/04/03/c-programming-read-file-lines-fgets-getline-implement-portable-getline/

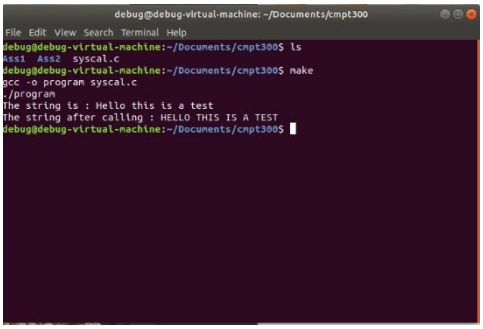
https://hackernoon.com/u/MIMA

Late Days claimed-> 1 day

Q2-> The assignment took us around 7 days to complete with constant debugging and troubleshooting. gdb really came to use with the debugging part and solving issues with memory management. All the knowledge from the lectures on pipe(), processes, etc. came to use with the assignment and we learned a lot about new things as well such as terminal colours, custom system calls. we honestly feel proud of what we were able to do with the assignment and looking forward to next assignment.

Bonus Question 1->

Our system call uppercase.c, takes in a str as a parameter and prints the string after it has been converted to uppercase. We chose not to use the string length as a parameter as it wasn't necessary for the execution of the system call (piazza @128). You can see an example of the implemented system call below

Bonus Question 2->

For our second system call, we decided to implement the theme build-in command as a system call, since the theme command is used and defined in this assignment, implement a system call to replace the whole function is a useful method and seems like unique compared to any other known system call that we have learnt of. To use this system call theme.c, we pass a lowercase string containing the colour we want the output to be changed to. The valid colour choices for this call are black, red, green, yellow, blue, purple, cyan, white. If we pass an invalid colour to the system call, it gives an error statement. An example using this system call can be seen below.