AMERICAN INTERNATIONAL UNIVERSTIY BANGLADESH (AIUB)

PROJECT REPORT

Project Title	:	BatWing
Project Type	:	Self-designed terminal using Bash shell script.
Author	:	IMTEAJ, MD. NABID
ID	:	11-19998-3
E-mail	:	nabid.imteaj@live.com
Course Name	:	Operating System
Section	:	[E]
Faculty	:	HOSSAIN, BAYZID ASHIK
Department	:	CSE

INDEX

✓ Objective

The goal of this project is to design a self-terminal and to create self-defined command for that terminal. Beside this, its aspect is also to get introduced with Linux surface as well as Bash scripting language.

✓ Reference

References from World Wide Web, other resources and several books.

✓ System Environment

Compiled and ran on Ubuntu 12.10 system with Intel i386 based system architecture.

✓ Introduction

Now a days, Linux is a stronger, densely used and interactive platform for both user and programmer. There was lot of people designed Linux. It is open source and based on Linux Kernel UNIX/Like operating system. Numerical number of Linux distros are available for different purposes.

Different people implemented the interpreter function of the shell different ways. This gave rise to various types of shells, the most prominent of which is the Bourne Shell. Bourne Again Shell, known as BASH, is the most popular and bundled with every UNIX/Linux system.

✓ BatWing

The BatWing is a self-designed terminal using Bash script designed for class project. It was very interesting implementing Bash command lines into the BatWing terminal.

Several methods and pre-defined functions were used inside the terminal. Choosing the name is very interesting. Because both the Bat and BatWing can echo(!) , it was named after the Bat.

✓ Why terminal project?

Among several choices self-designed terminal was very interesting to me. Due to time shortage Python and Java GUI were not included. Self-designed file comparer written in C/C++ was included. It was very challenging to work in a Linux environment first time.

✓ Project Descriptions

- The most important thing was to design a user interface. Using "tput" commands it became very easy. More about this command can be easily found on the terminal by "man tput" command. "tput clear" clears the screen. "tput cup row col" puts the interpreter to desired row by column. "tput setaf value" changes the font color as given value.

- Another interesting and challenging thing was work with the input/output interpreter. "< , >, <<, >>" are used for i/o interpreter. Like the "logname" was easily written in a file "tmp" using "logname > tmp" and then from "tmp" file the "logname" was read using "read usr < logname" commands.

- "if [[condition]]" was used instead of "if [condition]". This is because of if we use "if [condition]" for string comparing, if a string gets a null value then in the condition it vanishes like "if [string1 = string2]" becomes "if [= string2]" when string1 gets a null/return value from the interpreter while executing.

- The terminal takes commands as well as parameters. Like “history -v” gives view of history and “history -d” deletes history. “echo something” prints “something”.

- “help” option gives the command list used in this terminal.

✓ Future Plans

This project helped a lot working with Linux environment and using of Bash scripts as well as system interpretation. As a student of CS department it is very important to work with several platforms, in other way, Unix/Linux. This project played an important role to get a good relationship with these platforms. It has increased my hope to dare working with Kernels.

✓ Conclusion

The project came from the bottom of my heart. Though Java/Python was in my mind, but due to some cause it was not implemented. As it was my first project in Bash and Linux environment there may be some errors or bugs which should be excused. Thank You.

***