GROUP 44 DSA PROJECT REPORT

ASSIGNMENT SHELL

GITHUB REPO LINK : https://github.com/AnirudhGovil/DSA-project-group-44

This report is organised by contributors. Each contributor documents the details of their implementation.

Distribution of Work

* Aarush
    * setup- reads a text file from downloads and and creates a folder with this structure
    * test – run python code and store output in text file

* Vamsi
    * create- create a new folder and copy contents from download folder
    * update – delete old ones and add copy new ones

* Vaibhav
    * submit – zip file and keep it in the download folder.
    * compare – compare zip with assignment folder

* Anirudh
    * main.c, parse & execute - parse turns the raw input into tokens, execute forks a child process and runs the necessary function
    * switch - changes directories like cd, also directly accesses sibling directories by name
    * use - stores the location and name of a selected directory and uses it as a default argument

---

Anirudh Govil 2020101133

Parts implemented

File structure compilation
main.c
parse.h parse.c
execute.h execute.c
switch.h switch.c
use.h use.c
create.c create.h
update.c update.h
submit.c submit.h
compare.c compare.h

Algorithm

Take input from the user via main. Remove extra whitespaces and tokenise using parse. Call on the appropriate function using execute in a child process. Switch checks the parent directory for the argument if it isn't found using chdir() within the current directory. Use saves the name of the argument and displays the same in the prompt.

Data structures used: All of the above commands extensively use character arrays.

Parser takes O(n) time, where n is the length of the character input.

MD5 Hash runs in O(n) time.

The remaining commands run in O(1) time.

Vamsi Krishna Boyina 2020101071

Parts Implemented

create.h create.c
update.h update.c

Data structures used:

All of the above commands extensively use character arrays.

mkdir, **cp** and **rm** commands are used to implement create directory ,copy and remove the selected files/directories

Both create and update commands take O(1) time

Aarush Jain 2020101016

Parts Implemented

testfolder.h testfolder.c
setupfolder.h setupfolder.c

Data Structures used:

All of the above commands extensively use character arrays.
The `setup` command makes use of trees to represent and store the folder structure read from the text file.

The process of forming a tree out of the indented file has a time complexity of O(n).

To create a folder structure from a tree, we do pre-order traversal, creating a folder for each node. The time complexity for pre-order traversal is O(n).

Vaibhav Agarwal 2020101041

Parts Implemented

submitfolder.h submitfolder.c
comparefolder.h comparefolder.c

Data Structures used:

All of the above commands extensively use character arrays and diff function is used to find the difference in two files(which are compared). For the diff command the time complexity will be O(n) where n will be the number of files in the folder.