In Q2 of Assignment 1 of OS, we simulated a special Unix System and named it **‘BBOsh’**.

In BBOsh, there are a total of 4 commands:

1. **word:** Counts the number of words in a certain file. It has 2 further modes:
    a. -n : neglects ‘\n’ as a word
    b. -d : returns the absolute difference between the Number of Words in 2 files.

2. **dir:** Creates a new Directory and throws an Error if the Directory already Exists. It has 2 further modes:
    a. -r : Instead of throwing an Error if the directory does not exist, it removes the Current Directory and creates a new one.
    b. -v: Displays a message whenever another dir command is called in the future.

3. **date:** Displays the date and time when a particular file was Last Modified. It has 2 further modes:
    a. -R: Displays the output in RFC 5322 format

4. **exit:** Exits and Closes the Interactive Shell.

The Basic Working of the Program is as follows:

There is a **‘main.c’** file which resembles the Parent Process. It is continuously running in the background. Whenever the User enters a Command in our Shell, a **fork** happens and a child process is formed. The execution of the command takes place in the child process, and during this time, the Parent process is **waiting** for the child to complete. Then once the Child process Terminates, the Parent Process sends us back to the loop, where the User is asked for input again.

The **‘word’** command, being an internal command, also has its code in the **main.c** file and when the word command is called, appropriate functions are called which execute the word command.

The **‘dir’** and the **‘date’** commands are External; their codes are written in separate files, **‘dir.c’** and **‘date.c’**. When any of these two commands are entered by the User, the child performs **exec** to run the respective compiled file, which in turn executes the desired command.