COMP 304 : Operating Systems

Project 1 - Seashell

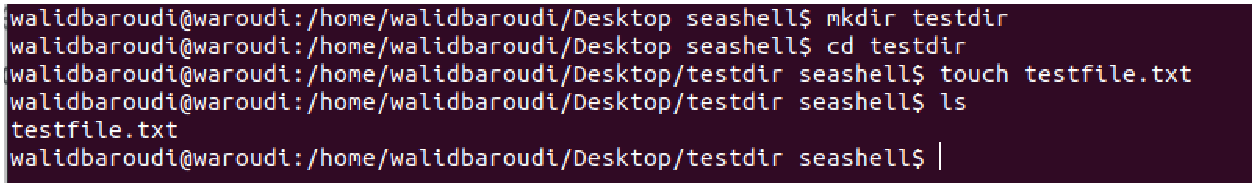
Abdul Rahman Hamadeh, Walid Baroudi.

Part I - seashell:

Brief Description:

Seashell is the shell that all the commands of this project will be run on.

Examples:



```
walidbaroudi@waroudi:/home/walidbaroudi/Desktop seashell$ mkdir testdir
walidbaroudi@waroudi:/home/walidbaroudi/Desktop seashell$ cd testdir
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/testdir seashell$ touch testfile.txt
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/testdir seashell$ ls
testfile.txt
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/testdir seashell$
```

Implementation:

First of all, we retrieved all the paths that are associated with the PATH environment variable using the **`getenv("PATH")`** method, and stored them in an array.

Then we appended the name of the command the user provides to all of the paths. After that, we ran **`execv()`** providing the stored paths as the path parameter for **`execv()`** function and the command the user provides as the command parameter.

If the command was not found within the stored paths, we try to find it in the path where the custom commands are located (the commands required for this project.)

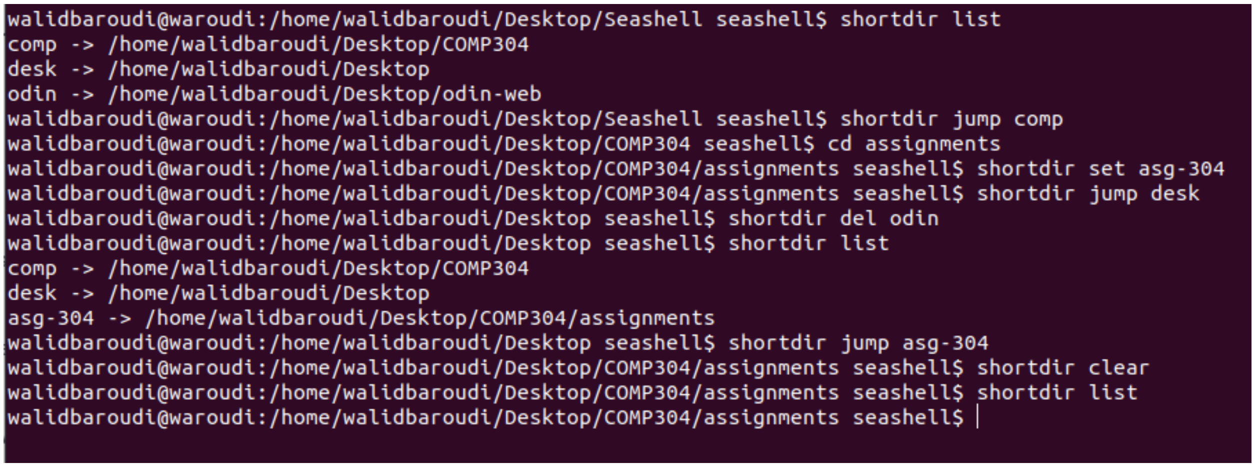
Part II - shortdir:

Brief Description:

A number of commands to create and control aliases for directories. Shortdir can be used to create an alias for the current directory, remove an alias from a given directory or all aliases at once, display saved aliases, and move to a directory associated with a given alias.

Examples:

A demo of all shortdir commands is shown below



```
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ shortdir list
comp -> /home/walidbaroudi/Desktop/COMP304
desk -> /home/walidbaroudi/Desktop
odin -> /home/walidbaroudi/Desktop/odin-web
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ shortdir jump comp
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/COMP304 seashell$ cd assignments
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/COMP304/assignments seashell$ shortdir set asg-304
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/COMP304/assignments seashell$ shortdir jump desk
walidbaroudi@waroudi:/home/walidbaroudi/Desktop seashell$ shortdir del odin
walidbaroudi@waroudi:/home/walidbaroudi/Desktop seashell$ shortdir list
comp -> /home/walidbaroudi/Desktop/COMP304
desk -> /home/walidbaroudi/Desktop
asg-304 -> /home/walidbaroudi/Desktop/COMP304/assignments
walidbaroudi@waroudi:/home/walidbaroudi/Desktop seashell$ shortdir jump asg-304
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/COMP304/assignments seashell$ shortdir clear
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/COMP304/assignments seashell$ shortdir list
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/COMP304/assignments seashell$ |
```

Implementation:

The aliases along with their associated directories are written to a text file in home/documents

Set: gets the current directory, writes the given name to a line, and the directory to the next line, and leaves a blank line after that. **Set** ensures there are no duplicates by calling **delete** on the given name first.

Del: writes the saved dirs to a new file, excluding the shortdir with the given name, if found. It then replaces the old file with the new file.

Jump: jump is the only command implemented inside **seashell.c** so that it can change the directory of seashell, rather than the shortdir executable itself. **Jump** searches for the shortdir associated with the given name and then moves to it.

List: list reads the contents of the file and displays them in a formatted way.

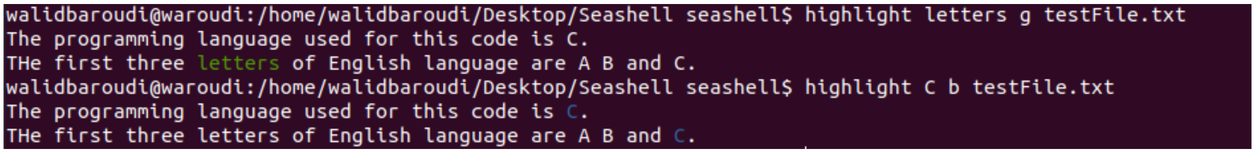
Clear: to clear the file, we open it in ‘w+’ mode, which, according to the docs, truncates the file size to 0, if found. We then simply close the file, since its contents were erased.

Part III - highlight:

Brief Description:

Given a text file and a word, the highlight command will highlight the provided word in the given text file with a color chosen by the user (red, green, blue).

Examples:



```
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ highlight letters g testFile.txt
The programming language used for this code is C.
THe first three letters of English language are A B and C.
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ highlight C b testFile.txt
The programming language used for this code is C.
THe first three letters of English language are A B and C.
```

Implementation:

Firstly, we created a file with a text matching the one provided in the project description pdf. Then we read the content of the file using **`fgets()`** method and tokenized the contents of the file.

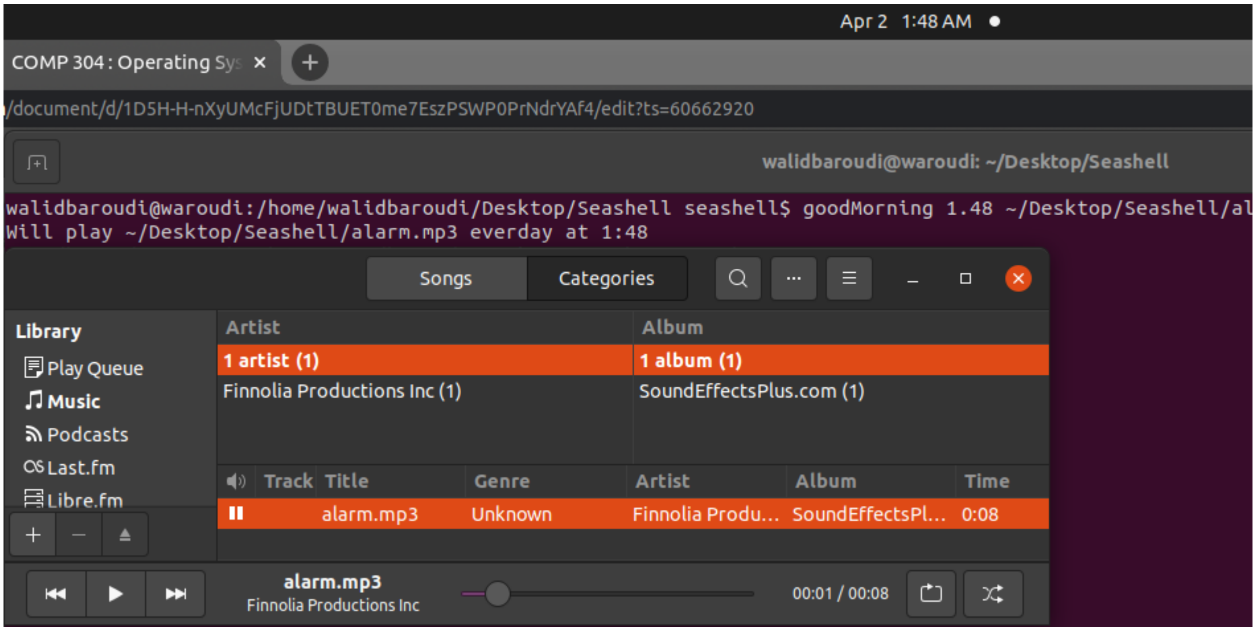
After tokenizing the text into words we compared them one by one with the word that is to be highlighted (provided by the user as a parameter). If one of the tokenized words matches the word given by the user, it will be colored and will replace its matching word (in the array of tokenized words). Finally, after the words’ comparison is done, the tokenized words are assembled as a string and are printed into the console.

Part IV - goodMorning:

Brief Description:

This command sets an alarm with the given time and sound file.

Examples:



```
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ goodMorning 1.48 ~/Desktop/Seashell/al
Will play ~/Desktop/Seashell/alarm.mp3 everday at 1:48
```

Artist	Album
1 artist (1)	1 album (1)
Finnolia Productions Inc (1)	SoundEffectsPlus.com (1)

Track	Title	Genre	Artist	Album	Time
	alarm.mp3	Unknown	Finnolia Produ...	SoundEffectsPl...	0:08

Implementation:

We used crontab and rhythmbox-client to achieve this command. First, the time entered by the user is processed to match the crontab format. Then some scripts are prepended to the rhythmbox command to define the environment variables, as crontab does not do that. The entire crontab script is then written into a file, and the file is loaded into crontab.

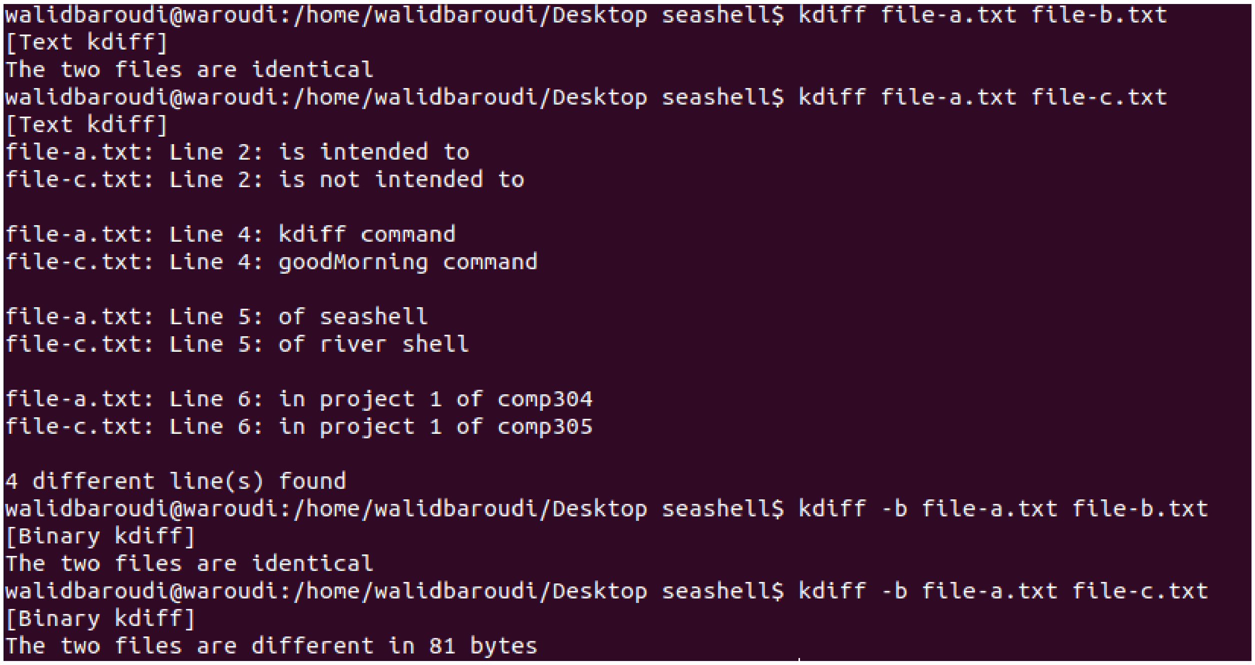
Part V - kdiff:

Brief Description:

A command that compares two files. kdiff compares two text files if flag ‘-a’, or none, is entered. If flag ‘-b’ is entered, kdiff compares two files of any type in their binary form.

Examples:

A demo of 4 cases of kdiff is shown below



```
walidbaroudi@waroudi:/home/walidbaroudi/Desktop seashell$ kdiff file-a.txt file-b.txt
[Text kdiff]
The two files are identical
walidbaroudi@waroudi:/home/walidbaroudi/Desktop seashell$ kdiff file-a.txt file-c.txt
[Text kdiff]
file-a.txt: Line 2: is intended to
file-c.txt: Line 2: is not intended to

file-a.txt: Line 4: kdiff command
file-c.txt: Line 4: goodMorning command

file-a.txt: Line 5: of seashell
file-c.txt: Line 5: of river shell

file-a.txt: Line 6: in project 1 of comp304
file-c.txt: Line 6: in project 1 of comp305

4 different line(s) found
walidbaroudi@waroudi:/home/walidbaroudi/Desktop seashell$ kdiff -b file-a.txt file-b.txt
[Binary kdiff]
The two files are identical
walidbaroudi@waroudi:/home/walidbaroudi/Desktop seashell$ kdiff -b file-a.txt file-c.txt
[Binary kdiff]
The two files are different in 81 bytes
```

Implementation:

In both cases, if one file has reached and end, the remaining lines (or bytes) are counted as differences.

Text: the two files are opened in read mode (‘`r`’), then a line is read from each file and the 2 lines are compared. Lines are read until <u>both</u> files are finished.

Binary: the two files are opened in binary read mode (‘`rb`’), then a byte is read from each file and the 2 bytes are compared. Lines are read until <u>both</u> files are finished.

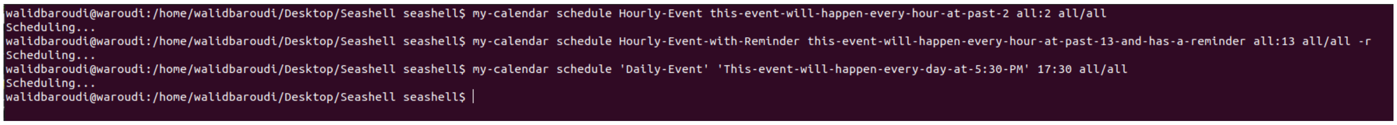
Part VI - my-calendar:

Brief Description:

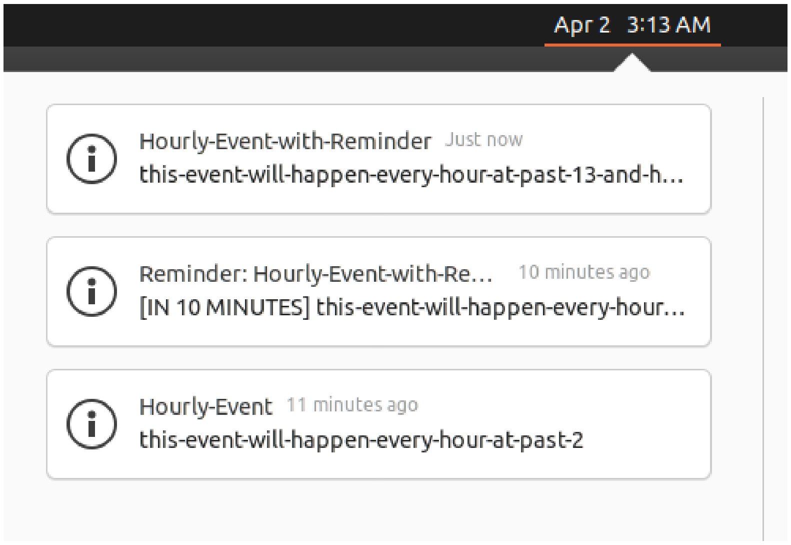
The calendar command is a command that allows the user to schedule events and view them. It also provides the user with the current weather conditions to see whether attending a certain event is a good idea or not.

Examples:

Scheduling:



```
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ my-calendar schedule Hourly-Event this-event-will-happen-every-hour-at-past-2 all:2 all/all
Scheduling...
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ my-calendar schedule Hourly-Event-with-Reminder this-event-will-happen-every-hour-at-past-13-and-has-a-reminder all:13 all/all -r
Scheduling...
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ my-calendar schedule 'Daily-Event' 'This-event-will-happen-every-day-at-5:30-PM' 17:30 all/all
Scheduling...
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ |
```

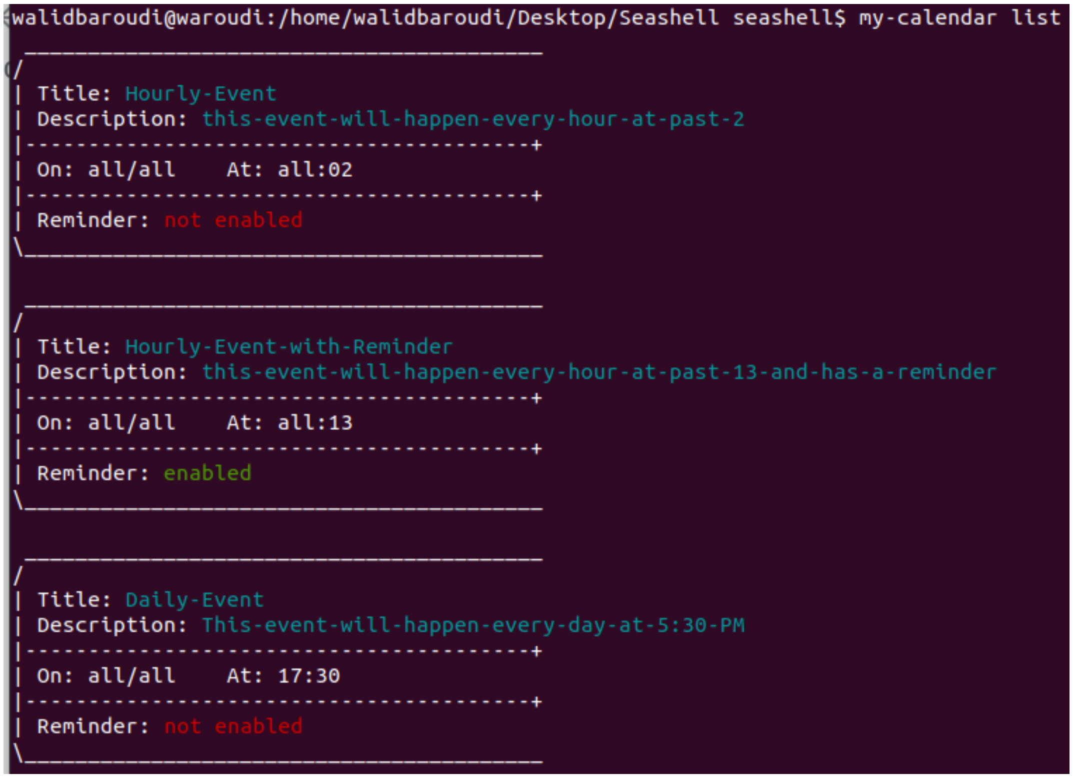


Hourly-Event-with-Reminder Just now
this-event-will-happen-every-hour-at-past-13-and-h…

Reminder: Hourly-Event-with-Re… 10 minutes ago
[IN 10 MINUTES] this-event-will-happen-every-hour…

Hourly-Event 11 minutes ago
this-event-will-happen-every-hour-at-past-2

Listing:

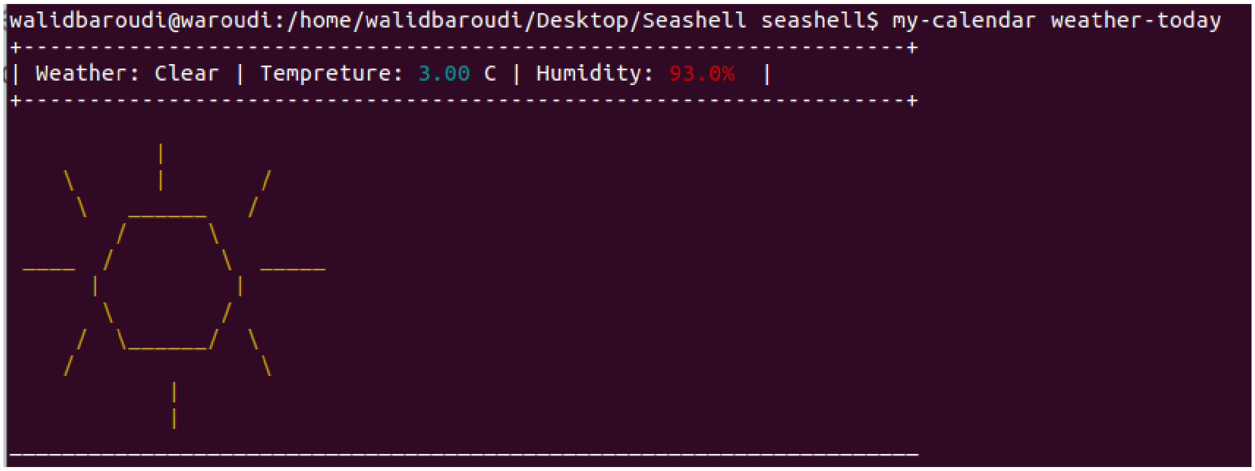


```
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ my-calendar list
 _________________________________________
/
| Title: Hourly-Event
| Description: this-event-will-happen-every-hour-at-past-2
|----------------------------------------+
| On: all/all    At: all:02
|----------------------------------------+
| Reminder: not enabled
\_________________________________________

 _________________________________________
/
| Title: Hourly-Event-with-Reminder
| Description: this-event-will-happen-every-hour-at-past-13-and-has-a-reminder
|----------------------------------------+
| On: all/all    At: all:13
|----------------------------------------+
| Reminder: enabled
\_________________________________________

 _________________________________________
/
| Title: Daily-Event
| Description: This-event-will-happen-every-day-at-5:30-PM
|----------------------------------------+
| On: all/all    At: 17:30
|----------------------------------------+
| Reminder: not enabled
\_________________________________________
```

Weather:



```
walidbaroudi@waroudi:/home/walidbaroudi/Desktop/Seashell seashell$ my-calendar weather-today
+-----------------------------------------------------------------------+
| Weather: Clear | Tempreture: 3.00 C | Humidity: 93.0%  |
+-----------------------------------------------------------------------+

            |
    \       |       /
     \    ______   /
        /        \
_____  /          \   _____
      |            |
       \          /
    /   \________/   \
   /                  \
             |
             |
________________________________________________________________________
```

Implementation:

The scheduling part: After the user provides the event information, the information is written into a file in a way that matches the `crontab` format, then the contents of the file are provided to the `crontab` to schedule.

The weather part: After typing the weather command, an HTTP GET request is sent to WeatherAPI.com and, in response, a JSON object is returned. Some information about the weather is retrieved from the JSON object and printed to the user along with a graphical print representing the current weather condition.

IMPORTANT NOTE 1 (weather-today)

In order for the **‘weather-today’** part of **‘my-calendar’** to work, the file **weatherApi.c** should be compiled. For that, libcurl must be installed on your machine. the file **weatherApi.c** should be compiled with **-lcurl** as follows:

```
gcc weatherApi.c -o weatherApi -lcurl
```

IMPORTANT NOTE 2 (schedule)

Title and description of events can be of any length when put in single quotes. However, since seashell gets args by tokenizing on whitespaces, multi-word titles and descriptions in seashell should have dashes ‘-’ between the words.