GCAL

Introduction

This assignment extends the functionality developed in the previous assignment (assignment 5).
Your task is to implement a C program named gcal that will function as a Linux command-line tool. The tool handles arguments to add, remove, and view reminders by passing arguments to the program. The tool has the following functionalities:

- A reminder can be added to a day using the following command:
  - gcal add <day> <reminder>
  - <day> is an integer between 1 and the number of days in the current month.
  - <reminder> is a string (may include spaces) representing the reminder to be added.
  - The output is the current month's calendar, and all the reminders added.
  - **An example is shown in the attached add.png**
- A reminder can be removed from a day using the following command:
  - gcal remove <day> <index>
  - <day> is an integer between 1 and the number of days in the current month.
  - <index> is an integer between 1 and the number of reminders added to that day.
  - The output is the current month's calendar and all the reminders added.
  - **An example is shown in the attached remove.png**
- The calendar and all the reminders can be viewed using the following command:
  - gcal view
  - The output is the current month's calendar and all the reminders added.
  - **An example is shown in the attached view.png**
- The reminders added to the current day can be viewed using the following command:
  - gcal today
  - The output is a list of all reminders added to the current day.
  - **An example is shown in the attached view.png**

Installation

Installing the program involves two steps:

1- Creating the executable:
   Create a makefile to build the program from the source files and create an executable file. **Refer to the example in install.png.**
2- Setting up the environment
   Place the executable file in a directory (e.g., $HOME/.local/bin) and update the $PATH environment variable to include this directory. This ensures that the Linux system recognizes the gcal command globally. **Refer to the example in which.png**

Write a shell script named install.h that will automate the previous installation steps.
The install.sh script will perform the following steps:

1- Run the make command to build the program using the makefile.
2- Create the installation directory (e.g. $HOME/.local/bin) if it does not already exist.
3- Copy the executable to the installation directory (e.g. $HOME/.local/bin)
4- Appends a command to the ~/.bashrc file to add the installation directory to the $PATH variable.

Submission

- Place the required files in a directory called gcal with the following structure:

|----- **src**
|---------- all .c and .h files are placed here
|---------- makefile
|------**install.sh**
|----- **screenshots**
|---------- add.png
|---------- remove.png
|---------- view.png
|---------- which.png
|---------- install.png

- The submitted screenshots must match the given screenshots
- Package the directory gcal into a tarball named gcal.tar
- Submit gcal.tar to OWL

Weight

This assignment is optional and not mandatory. It is a bonus assignment worth 5% of the course grade, ONLY applied to the final exam grade.
For example,

If you received 31% out of 35% in the final exam and 5% in this assignment, your final exam will be increased to 35%.

If you received 35% in the final exam and 5% in this assignment, no changes to your grade will be made.