Assignment 2: Working with Dates and Times

Task

This assignment builds on the `date` command and is specified by the following man-page-like description. Your job is to write a program that satisfies the requirements of this man page. The information regarding program submission is at the end of the assignment. I have written this man page to look like a "real one."

NAME

`datelist -- print future dates/times on a regular schedule`

SYNOPSIS

`datelist [-c <count>] <schedule>`

DESCRIPTION

Without options, `datelist` prints the dates and/or times obtained by adding the interval specified in the schedule to the current date/time, one per line, for ten occurrences in total. If the schedule contains units smaller than a day, then the dates and times are output. If the units are days or larger, only dates are output. In either case, the date/time is printed using the user's locale (`LC_TIME`) settings. The schedule string is of the form

```
<schedule>    =  <num> <time-unit> [<num> <time-unit> ... ]
<num>         =  [1-9][0-9]...
<time-unit>   =  year[s] month[s] week[s] day[s]  hour[s] minute[s]
```

`No time unit is allowed to repeat. There can be at most one occurrence of each unit.`

Examples

If the current date/time is Jan. 1, 2000 at 0:00:00 hours:

`date "10 days"` specifies the dates Jan 11, Jan 21, Jan 31, ... March 20 (leap year) *and no times would be displayed.* These would be printed using the user's locale format for dates, one per line.

`date "8 hours"` specifies the date/times Jan 1 8:00, Jan 1 16:00, Jan 2 0:00, Jan 2, 8:00, ... Jan 4 8:00. These would be printed using the user's locale settings for date and time, with dates preceding times. For example, for the U.S. locale *en_ US.utf8*, it would be of the form 01/01/2000 08:00:00 AM.

OPTIONS

The behavior of the command can be modified with the following option:

`-c <count>` If a count is given, then the number of occurrences that are printed is equal to the count.

EXIT STATUS

`0` If it succeeded.

`1` If it failed.

Error Handling

The program should detect missing command-line arguments, incorrect schedule syntax, negative amounts for any scheduling unit and/or the count option. It should report the error and print the correct usage.

Code Re-Use and Program Structure

You are free to use any of the code in the demos repository. All of it has GNU General Public Licenses. You are not free to use ChatGPT. If any code was not written by you, the documentation should indicate which parts are not written by you and credit its author. To make it easy to submit, the program should be a single source code file. All functions other than `main()` should precede `main()` in the file.

If the program uses any functions from the SPL library, libspl.a, you must instead copy the sources of those functions into your single source code file so that it can be compiled with a single command of the form

```
gcc -o output-file your-input-source-file.c
```

Guidance

- To write this program, you should thoroughly understand the implementation of the last version of the `spl_date` program from the book.
- The program will need to include all required header files. The man pages for every function and system call it may need show which header files to include.
- You need to make sure that you use the correct format for the user's dte and time locale.

Instructions for Submitting the Assignment

1. Use the `submithwk_cs49366` program to submit your program. It can be run on any `cslab` host, so login to a `cslab` host when you're ready to submit. To submit, if your file is named `myhwk.c`, you'd enter

   ```
   $ cd ~
   $ submithwk_cs49366  -t 2  myhwk.c
   ```

   The program will try to copy `myhwk.c` into the directory

   ```
   /data/biocs/b/student.accounts/cs493.66/hwks/hwk2/
   ```

   and if it is successful, it will display the message, "`File hwk2_`*`username`*`.c successfully submitted.`"

   where *`username`* is your username. You will not be able to read this file, nor will anyone else except for me. But you can double-check that the command succeeded by entering the command

   ```
   ls -l /data/biocs/b/student.accounts/cs493.66/hwks/hwk2
   ```

and making sure you see a non-empty file named `hwk2_`*`username`*`.c` where *`username`* is your user name and whose date of last modification is the time at which you ran the command.

2. ***You can do step 1 as many times as you want. Newer versions of the file will overwrite older ones.***

Deadline

You have one week to do this assignment. It must be submitted before its ***deadline***, which is ***Monday, March 4, at 7:00 PM. After that, you will not receive credit for completing it.***

Grading Rubric

This assignment is **16%** of your final grade. The output must be correct and the program must conform to the rules written in the ***Programming Rules*** document on the class's webpage. It must be thoroughly documented.