



## Assignment 1: Reporting User Statistics

This assignment is a good first assignment because, although it is not very challenging algorithmically, it requires attention to detail and it exercises several different knowledge areas. In particular, it requires command-line option parsing, accessing a system data structure (the `wtmp` file), and manipulating UNIX time values.

### Background

The `wtmp` file is identical to the `utmp` file in structure. The only difference as far as this assignment is concerned is that, whereas a logout causes an entry in the `utmp` file to be changed, a logout in the `wtmp` file adds an entry. When a user logs out on a terminal line, an entry with a NULL username field and a matching terminal line is added to the file, with the time at which the logout occurred. Thus, for each finished session there will be two entries – one, a login and a second, the logout, and what they have in common is the same pseudo-terminal name and number.

### The Task

You are to write a program that satisfies the requirements of the following man page. The information regarding program submission and content is at the end of the assignment.

#### NAME

```
logtime -- print various statistics of logtime of one or more users
```

#### SYNOPSIS

```
logtime [-as] [ -m min_time] [-u d|h|m|s ] [-f file] [username] [username] ...
```

#### DESCRIPTION

Without options, `logtime` prints one line for each username argument, containing the total time that the user has spent logged into the system since record keeping was last started. If no usernames are listed, it displays this time for the current user. If there is no `wtmp` file and the `-f` option is not present, it prints a message on the standard error stream that there is no source for login information. Otherwise, for each username argument, it displays the username followed by the total login time, accurate to the second, in the time format specified below, using either the `wtmp` file, or the file supplied with the `-f` option.

Only completed login sessions contribute to the accounting. Sessions that are in progress at the time the command is invoked are not counted in any calculations, so if there is a login without a subsequent logout, that login is ignored.

If a user is logged in on more than one terminal device simultaneously for a period of time of duration  $t$ , that interval of time is not counted more than once. Thus, for example, if a user is logged in on `pts/1` from 1:00 to 2:00 and on `pts/2` from 1:10 to 2:10, then the total time is from 1:00 to 2:10, or 70 minutes.



Total time is displayed in units of days, hours, minutes, and seconds unless the `-u` option is present. Without the `-u` option, if the total is less than a day, the days are omitted, and if less than an hour, the hours and days are omitted, and if less than a minute, only seconds are displayed. Without the `-u` option, if the value of a unit is 0, it is not displayed. (This rule implies the preceding ones, but also implies that a time such as “1 day 2 seconds” is possible.) If a username is given but there are no logins for the user, "0 seconds" is displayed for that username. Without the `-u` option, all times are non-negative integers. Usernames are not sorted alphabetically. For example, the output may look like

```
sweiss      1 day    9 hours  39 mins  51 secs
saad        14 hours  1 sec
dbreczin    22 mins  42 secs
```

In the second line, it is an implicit time of 0 days, 14 hours, 0 minutes and 1 second.

## OPTIONS

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

- `-a`  
Show the statistics for all users. If there are username arguments, they are ignored.
- `-f file`  
Use file instead of the `wtmp` file.
- `-m min_time`  
Omit from the calculation all login sessions whose total length is less than `min_time` seconds.
- `-s`  
Show the average login time for each user instead of the total. The average is the total divided by the number of sessions. A session is a continuous sequence of overlapping intervals. If a user is logged in on different terminals, a session starts at the start of the earliest login and ends at the end of the last logout.
- `-u h|d|h|m|s`  
Without exactly one of `d`, `h`, `m`, or `s` following the `-u` option, this is an error. The time is displayed as a fixed decimal number of days, hours, minutes, or seconds respectively with a precision of 2 decimal places.

These options are independent; for example

```
logtime -as
```

shows the average logtimes for all users, and

```
logtime -s -m 100
```

computes the average, omitting all sessions less than 100 seconds long.



---

## EXIT STATUS

- 0 If it succeeded.
- 1 If it failed.

## FILES

`/var/log/wtmp`

## SEE ALSO

`login (3)`, `logout (3)`, `utmp (5)`, `wtmp (5)`

## Program Requirements

The program must be written in C or C++. Each source code file must be well-documented according to established standards and must contain a preamble or prologue at the very top like the ones in my demo programs, which provides authorship, use, and other pertinent information. I have to be able to build and run the program without error on the server, `cs82010.gc.cuny.edu`.

## Submitting the Assignment

Create a directory in

`cs82010.gc.cuny.edu:/home/class_stuff/cs82010/projects/project1`

named `username_hwk1`. Give it permissions 700 so that only you have access to it. Within that directory, put all project-related files. For example, if you have a multi-file project with a Makefile, all files should be there. If you use a Makefile, make sure you name the executable `logtime` when it is built. If you write a multi-file program, you must create a Makefile, so that I can compile and build it with the single command "make." If it is a single file, name it `logtime.X`, where X is the GNU extension for the language (.c for C, .cpp or .C for C++, etc.)