CSCB09H3 - Assignment 1: C program to monitor different resources

Student Number: 1006906680

Libraries included:

- `stdio.h`
- `stdbool.h`
- `stdio.h`
- `stdlib.h`
- `utmp.h`
- `string.h`
- `unistd.h`
- `sys/utsname.h`
- `sys/sysinfo.h`
- `sys/resource.h`
- `getopt.h`

Functions:

- `cpu_info()`
  fetch and print the cpu information namely the number of cores and cpu utilization and print it to the terminal in non-graphical format.
- `cpu_info_graphic()`
  fetch and print the cpu information namely the number of cores and cpu utilization and print it to the terminal in graphical format.
- `memory_info(bool graphic, float *prev_mem , float *curr_mem, int i)`
  fetch the physical and virtual memory information regarding its utilization and availability and print it to the terminal in graphical or non-graphical format as described by the graphic variable.
- `self_usage()`
  fetches and prints the memory used by the current process to the terminal.
- `system_info()`
  fetches and prints the system information including system name, machine name, version, release, and architecture of the machine on to the terminal.
- `sessions_info()`
  fetches and prints the information regarding users currently active on the machine on to the terminal.
- `section_divide()`
  outputs a series of '–' to differentiate between different sections of output in the terminal

Flags:

- `--user` used to display only user information.
- `--tdelay=t` used to specify the timedelay to be introduced between sampling. To be specified in unit seconds.
- `--samples=N` used to specify the number of samples to be taken while sampling.
- `--graphics` used to specify format of output. By default the output is non-graphical.
- `--system` used to display only system information.

Note: Above is the recommended order of the flags to be described in command passed in the command line. A different ordering might cause unexpected behaviour. Also user and system are mutually exclusive flags and cannot be called together.

Report:
Q: How did you solve the problem?
A: I first identified the things that need to be presented in the output and then went over finding those things through the given documentations and man pages. After I found all the references and included required libraries, I implemented one part of the output per function and tested each function after implementation. After my basic layout for the program was done, I implemented the flags required as per the function description and lastly, I implemented the graphical output as desired. To implement simultaneous output rather to create that illusion the program uses ANSI ESCodes. All of this was done using the nano editor on the lab machine through ssh. After I had thoroughly tested my program, I copied the file to my local system using scp and wrote the documentation and the report.

Q: How to run your program?
To run the above program one can compile the attached file assignment_1.c using a C compiler like gcc. Then run the executable file generated by the compiler along with the desired flags for the desired output. The program will print all the information according to the specifications on to the terminal.