

CS570 Summer 2019 Assignment 3

This page last modified 16 May, 2019

Design and implement an event management program (emp).

You shall create a program that uses signals to coordinate work among sibling threads. You will build one thread to be the signal handle and worker threads to perform the following (note, you will also use semaphores as needed):

- Your program shall read the arguments provided by the user. If arguments are not provided use default values as stated below.
- Your program shall create worker threads from the main, initial thread which busy-waits until all worker threads have completed, then, perform a clean exit.
- One of the worker threads shall manage/monitor the countdown timer, upon reaching the specified number of seconds it shall signal all threads to terminate and then terminate itself. Note, the thread shall use `signal()` to do this.
- Another worker thread shall manage/monitor the “countdown timer” and generates/signals that this time has elapsed.
- Another process shall implement a “wall clock” which prints the hour, minute, and second of the time of day to the user terminal once every second (in human readable form, hint, use `localtime`). When the hour, minute, and second matches the user-provided countdown time, an Alarm would be printed in addition to the time let the user know the program is terminating.
- Once all child threads have terminated, the parent thread (and the process) shall print a friendly message to the user (using `stdout`), then perform a clean exit.

User provided inputs/defaults (e.g. `% emp: 45 1 15`) – note, 45 is 1st parameter, 1 is 2nd parameter, and 15 is 3rd parameter

- The first parameter will be time (in seconds) the program will run before it exits. If no value is provided, the default value of 25 seconds shall be used for this countdown timer.
- The second parameter will be 1 or 60, if it's 1 then print out time every second, if 60 then print out time every minute, all other values will be an error and your program shall gracefully exit
- The third parameter will be used as a countdown timer, the value is the time (in seconds) when one of the worker threads shall schedule a signal to occur (in seconds) that the countdown “reached zero” (i.e., that many seconds have just elapsed).
- If just `emp` is entered, use the following default values: 32, 1, 17

Your project shall include a README file using the same conventions/requirements specified in the course README instructions file.

Your program will be tested by compiling and executing on **edoras**. Your program shall be written such that it compiles and executes cleanly when using `gcc/g++`. **Note - you must use a Makefile**. You shall create a sub-directory named “**a3**” in your home directory. In it, you shall place all of your project files, including your Makefile. Your source files shall contain sufficient comments for making the source easy to read. Points will be taken off for poorly (or non) commented source. Name the executable “**emp**”. Also, create an archive file (tarball, zip) and upload to Blackboard (one student per project).

- Create `~/a3` by hand.

- Create all necessary project files. Put them into ~/a3.
- The Makefile shall create an executable named "emp" in this same directory (~/a3).
- The system call "system()" will NOT be allowed
- You must work individually or in pairs (individually or a team of 2 students)
- You may use gcc, or g++ compiler on this assignment

The assignment is due by 1800 on Sunday, 7 Jul 2019

TURNING IN YOUR WORK:

Follow the turn-in procedures on class Blackboard.