Homework 1. Alarms.

a) Propose an architecture (an understandable pseudocode) of a program which:

- Displays a running real-time clock with a resolution in seconds.
- Reads (without unnecessary delays) from stdin a (possibly very long) sequence of alarm times in the form of positive decimal floating point numbers separated by white spaces. The numbers are referring to times from the moment when the program has been started (in fractions of a second).
- Triggers the alarms, writing to stdout a sequence of the sum of (positive) differences between the desired and actual triggered alarm times measured so far (in fractions of a second). No alarm may be triggered sooner than its time specifies.
- Terminates when the standard input stream ends (reading an alarm with time 0), and after all pending alarms have been triggered.

The goal is to minimise the total triggering delay over all alarms, i.e. the final sum of the differences between the desired and actual triggered alarm times.

b) Explicitly state the number of threads used in your program.

c) Report essential (POSIX) system calls which are likely to appear in the implementation of your program, for each system call state in which thread you expect it to appear.

Hand out a PDF file.