

Laboratorio de Semana 4

Hora Marciana Coordinada

La Administración Nacional del Espacio y Aeronáutica (NASA por sus siglas en inglés) ha decidido que todas sus misiones relacionadas con el planeta Marte utilizarán relojes sincronizados a tiempo marciano: todos los operadores y científicos viven en tiempo marciano mientras están asignados a la misión. Un día marciano promedio se define como 88.775,244147 segundos. Debido a que esto es solo ligeramente más largo que el día terrestre promedio (86.400 segundos, ó 24 horas), la NASA ha decidido que el día marciano se subdividirá en 24 horas marcianas equitativas, cada una subdividida en 60 minutos marcianos, y cada uno, a su vez, subdividido en 60 segundos marcianos. Debido a que estas divisiones se basan en el día marciano, cada uno tiene una duración proporcional ligeramente mayor que su contraparte terrestre.



En este laboratorio, usted deberá diseñar una aplicación que imprima, cada vez que avance un segundo en la tierra, la hora actual; y cada vez que avance un segundo en marte, la hora marciana. El programa debe recibir tres parámetros, indicando la hora actual en cada planeta y el tiempo (en unidades terrestres) que se mantendrá el programa corriendo antes de cerrarse. El reloj no debe atrasarse aunque la impresión tome más de un segundo.

Puede consultar la hora marciana actual en <http://jtauber.github.io/mars-clock/>.

1 Requisitos del programa

- Debe usar `fork()` y `wait(...)`
- Debe usar `time(...)`, `gettimeofday(...)` ó `clock_gettime(...)`
- Debe usar alguna función de la familia `exec(...)`
- Debe usar solo una de las siguientes: `sleep(...)`, `usleep(...)` ó `nanosleep(...)`; pero puede usarla tantas veces como desee
- Debe tener un `makefile`

Puede consultar estas funciones en la [especificación del Estándar POSIX](#) o escribiendo `man` seguido del nombre de la función en la consola.

2 Extra Credit

Puede presentar el siguiente programa al final del curso si le faltan puntos:

- Investigue los comandos equivalentes del API de Windows e implemente una aplicación equivalente que pueda correr en su *Command Prompt* (Símbolo de Sistema)
- Para puntos adicionales, implemente la aplicación de tal manera que pueda ser compilada en Windows o en Linux usando compilación condicional (`#if defined(WINDOWS)`, etc)