

## Sistemes de Temps Real. Pràctica 2.

### Presentació.

Pràctica 2 de sistemes de temps real que consisteix en fer els següents exercicis guiats sobre sistemes de Temps Real i POSIX.

### Exercici 1. Creant Threads de temps real.

1.1.- Escriviu un programa que creei múltiples threads de temps real fent servir la funció `pthread_create()`.

Podeu trobar més informació al següent enllaç.

[pthread\\_create](#)

1.2.- Assigneu diferents “scheduling policies” (p.ex.: SCHED\_FIFO, SCHED\_RR) i diferents prioritats a cadascun dels threads.

1.3- Observeu el comportament dels threads documenteu-lo.

### Exercici 2. Fent servir POSIX Timers

2.1.- Escriviu un programa que faci servir un timer de POSIX fent servir la funció `timer_create()`

Podeu trobar més informació i exemples al següent enllaç:

[timer\\_create\(2\): create POSIX per-process timer - Linux man page](#)

2.2.- Configureu el timer per enviar una senyal (p.ex.: SIGALARM) en intervals regulars.

2.3.- Escriviu un “signal handler” per a que faci una tasca cada cop que el timer expiri.

### Exercici 3.- Cues de missatges.

3.1.- Escriviu un programa productor-consumidor fent servir les cues en POSIX (`mq_open()`,`mq_send`, `mq_receive()`)

[https://users.pja.edu.pl/~jms/qnx/help/watcom/clibref/mq\\_overview.html](https://users.pja.edu.pl/~jms/qnx/help/watcom/clibref/mq_overview.html)

**Nota:** Estigueu segurs que el productor i el consumidor funcionen en threads diferents.

3.2.- Experimenteu amb diferents prioritats dels missatges i observeu l'impacte en el

processament dels missatges per part del consumidor.

3.3.- Documenteu l'aplicació i el comportament observat.

**Exercici 4.- Memòria compartida.**

4.1.- Creeu un programa que faci servir la memòria compartida en POSIX (`shm_open()`, `nmap()`) per a compartir dades entre 2 processos.

Podeu trobar més informació al següent enllaç:

[https://pubs.opengroup.org/onlinepubs/009695099/functions/shm\\_open.html](https://pubs.opengroup.org/onlinepubs/009695099/functions/shm_open.html)

4.2.- Implementeu alguna mesura de protecció de la memòria compartida com ara semàfors.

4.3.- Documenteu el que heu fet.

### **Exercici 5.- Senyals de Temps Real.**

5.1.- Escriviu un programa que envii i manegui senyals de temps real (de *SIGRTMIN* a *SIGRTMAX*)

5.2.- Feu servir *sigqueue()* per a enviar senyals amb dades.

5.3.- Implementeu un manegador de senyals per a processar les senyals rebudes i les seves dades.

<https://man7.org/linux/man-pages/man3/sigqueue.3.html>

### **Exercici 6.- Inversió de prioritats.**

6.1.- Desenvolupeu un escenari on la inversió de prioritats pugui passar.

6.2.- Feu servir POSIX mutex amb herència de prioritats per a poder la inversió de prioritats. (*pthread\_mutexattr\_setprotocol()*).

Podeu trobar més informació al següent enllaç

[https://pubs.opengroup.org/onlinepubs/7908799/xsh/pthread\\_mutexattr\\_setprotocol.html](https://pubs.opengroup.org/onlinepubs/7908799/xsh/pthread_mutexattr_setprotocol.html)

6.3.- Proveu i documenteu el comportament abans i després de fer herència de prioritats

### **Lliurament**

Heu de lliurar un document amb les diferents observacions que l'exercici us va demanant en .pdf i un fitxer comprimit amb tots els exercicis de codi en c.

**Data màxima per a lliurar la tasca:** dilluns 12 de gener a les 23:55 h