Sistemas Operativos
Trabajo Práctico Nº 2

Threads y Concurrencia

Objetivos

- Implementar una aplicación “multithread” con el API POSIX.

Ejercicios

Implementar una aplicación multi-thread que permita realizar operaciones matemáticas (suma y multiplicación) sobre 2 matrices de enteros.

El programa deberá crear un thread por cada renglón de la matriz resultante. El tamaño de las matrices deberá ser de **<n x n>**, donde **<n>** será un parámetro estático del programa.

Utilizar las funciones provistas por pthread:

```
#include <pthread.h>
```

pthread_create()

Creación de un thread, devuelve 0 si tiene éxito:

```
int pthread_create(pthread_t *thread,
                   const pthread_attr_t *attr,
                   void *(*start)(void *),
                   void *arg);
```

pthread_exit()

Termina un thread, devuelve el valor *retval* que puede ser recuperado con thread_join():

```
void pthread_exit(void *retval);
```

pthread_self()

Devuelve el ID del thread que llama a la función:

```
pthread_t pthread_self(void);
```

pthread_join()

Espera a que un thread termine, retval es el valor devuelto por el thread:

```
int pthread_join(pthread_t thread, void **retval);
```