



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

UNIVERSIDAD DE BUENOS AIRES

MATERIA

FECHA DE CURSADA

Tema del trabajo práctico

Informe Individual

Darius Maitia - 95436

Contents

1 Concurrencia

1.1 Dos razones para la concurrencia

- Eficiencia: hay tareas que se hacen mejor de a varios
- I/O : en un mismo proceso un hilo se ocupa del input/output y los otros hilos se ocupan de otras cosas.

1.2 Concurrencia vs procesos

- Los procesos se ejecutan de forma aislada
- Los threads comparten recursos:
 - Comparten la memoria; entre threads no se necesita switchear page tables ni nada.
 - No comparten stack!** : cada hilo tiene su propio stack.

1.3 Crear y joinear threads en C

```
#include <pthread.h>;

void * mythread(void * arg){
    printf("Imprimo gilada con una letra al final: %c", arg);
}

int main(){
    int i;
    pthread_t p1, p2;
    pthread_create(&p1, NULL, mythread, "A");
    pthread_create(&p2, NULL, mythread, "B");

    pthread_join(&p1, NULL);
    pthread_join(&p2, NULL);

    return 0;
}
```

Pthread_create y pthread_join deberían devolver 0 en caso de éxito.

1.4 Critical sections

Definición : una critical section es una parte del código donde hay una variable compartida a la que pueden acceder múltiples threads.

Para encarar este problema hay que hacer lo siguiente:

1. Separar las incumbencias de los threads lo más que se pueda en el código.
2. En donde hayan variables de acceso compartido entre los threads, usar locks.

1.4.1 Locks

Los locks permiten acceder a una variable de una critical section con total seguridad, sabiendo que otro hilo no podrá acceder a esa variable mientras yo esté usándola.

La lógica es la siguiente:

```
lock_t mutex;
...
lock(&mutex);
balance = balance + 1;
unlock(&mutex);
```

En lenguaje de programación C la lógica es la siguiente:

```
pthread_mutex_t lock = PTHREAD_MUTEX_INITIALIZER;  
Pthread_mutex_lock(&lock);  
balance = balance + 1;  
Pthread_mutex_unlock(&lock);
```

2 Conclusión

References

- [1] Michael Larabel, *Mozilla's Servo Engine Is Crazy Fast Compared To Gecko*, 9 nov. 2014, https://www.phoronix.com/scan.php?page=news_item&px=MTgzNDA