

# Sistemas Concurrentes y Distribuidos - Portafolio de prácticas

28 de septiembre de 2016

## Parte I

# Seminario 1. Programación multihebra y sincronización con semáforos.

## Hebras: introducción.

Una hebra es un flujo de control en el texto (zona con la secuencia de instrucciones a ejecutar) del proceso al que pertenece (que puede ser común a varias hebras).

Los ejemplos utilizan hebras de POSIX, más concretamente, utilizan *hebras del kernel de linux*, esto significa que hebras distintas de un mismo proceso pueden ejecutarse en procesadores distintos (lo cual hace que sean una herramienta ideal para aprovechar el potencial de los sistemas multiprocesador).

### Creación de hebras.

Utilizamos la función *pthread\_create*:

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

La forma más simple de crear hebras:

```
#include <pthread.h>

void *proc1( void *arg){
    ....
}

void *proc2( void *arg){
    ...
}

int main(){

pthread_t hebra1, hebra2 ;
pthread_create(&hebra1, NULL, proc1, NULL) ;
pthread_create(&hebra2, NULL, proc2, NULL) ;
}

/* Si usamos memoria dinámica sería algo como:

pthread_t *hebra1 ;
pthread_t *hebra2 ;
hebra1 = new pthread_t ;
hebra2 = new pthread_t ;
pthread_create(hebra1, NULL, proc1, NULL) ;
pthread_create(hebra2, NULL, proc2, NULL) ; */
```