Larbi Ben M'hidi University – Oum El Bouaghi

Faculty of Exact Sciences and Nature and Life Sciences

Department of Mathematics and Computer Science

Module: Operating Systems II

Lecturer: Dr. Moustafa Sadek KAHIL

2025 – 2026

« pthread.h » Library in C

Function	Usage	Example
pthread_t thread;	Declare a thread variable	pthread_t thread;
pthread_create()	Creates a new thread and executes a function.	pthread_create(&thread, NULL, thread_function, &arg);
pthread_exit()	Terminates the execution of a thread.	pthread_exit(NULL);
pthread_join()	Waits for a thread to finish.	pthread_join(thread, NULL);
pthread_mutex_init()	Initializes a mutex for synchronization.	pthread_mutex_init(&mutex, NULL);
pthread_mutex_lock()	Locks a mutex to protect a critical section.	pthread_mutex_lock(&mutex);
pthread_mutex_unlock()	Unlocks a mutex.	pthread_mutex_unlock(&mutex);
pthread_cond_wait()	Waits for a condition (blocks until it is signaled).	pthread_cond_wait(&cond, &mutex);
pthread_cond_signal()	Wakes up a thread waiting on a condition variable.	pthread_cond_signal(&cond);
pthread_detach()	Detaches a thread so it releases its resources upon	pthread_detach(thread);

	completion without needing a join().	
pthread_self()	Retrieves the calling thread's identifier.	pthread_t id = pthread_self();

Example of thread creation and termination:

```
#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>



void*  thread_function(void*  arg)  {
int* num = (int*)arg;

  printf("Thread %d is running.\n", *num);   pthread_exit(NULL);

}



int       main()       {
pthread_t     thread;
int arg = 42;


  // Creating the thread

  if (pthread_create(&thread, NULL, thread_function, &arg) != 0)
{      perror("Error while creating the thread");       return -1;

  }



if (pthread_join(thread, NULL) != 0) {
perror("Error     at     pthread_join");
return EXIT_FAILURE;

  }



  printf("Thread terminated.\n");

return 0;

}
```