**Code: Thread scheduling**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

// Function prototypes

void \*thread\_function(void \*arg);

int main() {

pthread\_t tid1, tid2; // Thread IDs

int thread1\_arg = 1, thread2\_arg = 2; // Arguments to threads

// Create first thread

if (pthread\_create(&tid1, NULL, thread\_function, (void \*)&thread1\_arg) != 0) {

fprintf(stderr, "Error creating thread 1\n");

return 1;

}

// Create second thread

if (pthread\_create(&tid2, NULL, thread\_function, (void \*)&thread2\_arg) != 0) {

fprintf(stderr, "Error creating thread 2\n");

return 1;

}

// Wait for both threads to finish

if (pthread\_join(tid1, NULL) != 0) {

fprintf(stderr, "Error joining thread 1\n");

return 1;

}

if (pthread\_join(tid2, NULL) != 0) {

fprintf(stderr, "Error joining thread 2\n");

return 1;

}

printf("Both threads have finished execution\n");

return 0;

}

// Thread function

void \*thread\_function(void \*arg) {

int thread\_arg = \*((int \*)arg);

printf("Thread %d is running\n", thread\_arg);

// Do some work...

printf("Thread %d is finished\n", thread\_arg);

pthread\_exit(NULL);

}