

Wymagania:

1. Jeden wątek powinien wypisać słowo "hello ", drugi słowo "world", a funkcja main powinna dopisać symbol "\n".
2. Wątek "world" powinien zostać utworzony przed wątkiem "hello".
3. Oba wątki, "hello" i "world" powinny być wykonywane współbieżnie.

```
/*
gcc -o hello hello.c -lpthread
*/

#define _REENTRANT
#include <pthread.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>

pthread_mutex_t prt_lock = PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t prt_cv = PTHREAD_COND_INITIALIZER;
int prt = 0;

/* hello_thread - print the "hello" part. */
void *hello_thread( ) {
    pthread_mutex_lock( &prt_lock );
    printf( "hello " );
    prt = 1;
    pthread_cond_signal( &prt_cv );
    pthread_mutex_unlock( &prt_lock );
    return( 0 );
}

/* world_thread - print the "world\n" part. */
void *world_thread( ) {
    pthread_mutex_lock( &prt_lock );
    while ( prt == 0 )
        pthread_cond_wait( &prt_cv, &prt_lock );
    printf( "world" );
    pthread_mutex_unlock( &prt_lock );
    pthread_exit( 0 );
}
/* NOTREACHED */
}
```

```

/* main - start here.                                     */
main( ) {
    int      i;
    int      n;
    pthread_attr_t attr;
    pthread_t tid;

    pthread_attr_init( &attr );
    pthread_attr_setdetachstate( &attr, PTHREAD_CREATE_DETACHED );
    if ( n = pthread_create( &tid, &attr, world_thread, NULL ) ) {
        fprintf( stderr, "pthread_create: %s\n", strerror( n ) );
        exit( 1 );
    }
    pthread_attr_destroy( &attr );
    if ( n = pthread_create( &tid, NULL, hello_thread, NULL ) ) {
        fprintf( stderr, "pthread_create: %s\n", strerror( n ) );
        exit( 1 );
    }

    if ( n = pthread_join( tid, NULL ) ) {
        fprintf( stderr, "pthread_join: %s\n", strerror( n ) );
        exit( 1 );
    }
    printf( "\n" );
    return( 0 );
}

```