Кафедра прикладной математики и кибернетики

Выполнили:

Студенты 3 курса группы ИП-111  
Корнилов А.А.,  
Попов М.И.,

Толкач А.А.

Проверил:

Профессор кафедры ПМиК  
Малков Е.А.

ЛАБОРАТОРНАЯ РАБОТА №9

По дисциплине: «Операционные системы»

Сибирский государственный университет  
телекоммуникаций и информатики

Новосибирск, 2023

Министерство цифрового развития, связи  
и массовых коммуникаций Российской Федерации

**Задание:** протестируйте спин-блокировку используя фрагменты кода лекции 9.

**Цель:** протестируйте спин-блокировку используя фрагменты кода лекции 9

**Выполнение работы:**

Ниже представлена модифицированная программа из лекции 9.

В функции Thread: Поток захватывает мьютекс перед изменением содержимого массива sh с помощью pthread\_mutex\_lock(&mutex). Затем он проверяет, является ли counter четным или нечетным, и изменяет содержимое массива sh соответственно. После этого поток освобождает мьютекс с помощью pthread\_mutex\_unlock(&mutex);.

В функции main: Основная программа также использует мьютекс для захвата перед чтением содержимого массива sh с помощью pthread\_mutex\_lock(&mutex). Затем она выводит содержимое массива sh на экран. После этого основная программа освобождает мьютекс с помощью pthread\_mutex\_unlock(&mutex). Таким образом, использование мьютекса обеспечивает синхронизацию доступа к общему ресурсу (sh) из разных потоков, предотвращая возможные проблемы, такие как гонки данных.

|  |
| --- |
| #include <stdio.h>  #include <time.h>  #include <pthread.h>  #include <unistd.h>  char sh[6];  int n = 100;  pthread\_mutex\_t mutex = PTHREAD\_MUTEX\_INITIALIZER;  void\* Thread(void\* pParams){  int counter = 0;  while (1){  pthread\_mutex\_lock(&mutex); // Захватываем мьютекс перед записью в sh  if (counter % 2){  sh[0] = 'H';  sh[1] = 'e';  sh[2] = 'l';  sh[3] = 'l';  sh[4] = 'o';  sh[5] = '\0';  }  else{  sh[0] = 'B';  sh[1] = 'y';  sh[2] = 'e';  sh[3] = '\_';  sh[4] = 'u';  sh[5] = '\0';  }  counter++;  pthread\_mutex\_unlock(&mutex); // Освобождаем мьютекс после записи в sh  }  return NULL;  }  int main(void){  int t = 10;  pthread\_t thread\_id;  pthread\_create(&thread\_id, NULL, &Thread, NULL);  while (t--){  pthread\_mutex\_lock(&mutex); // Захватываем мьютекс перед чтением sh  printf("%s\n", sh);  pthread\_mutex\_unlock(&mutex); // Освобождаем мьютекс после чтения sh  }  } |

Листинг 1 – программа lab09\_2.c