



**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»**

**ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ  
Кафедра программных систем**

**А.В. Баландин**

## **Лабораторная работа**

### **Запуск и синхронизация нитей**

**Методические указания**

**Самара  
2022**

Оглавление	цель работы.....	Ошибка! Закладка не определена.
ЗАДАНИЕ 1 .....		3
ЗАДАНИЕ 2 .....		4
ЗАДАНИЕ 3 .....		5
ТРЕБОВАНИЯ К РАЗРАБОТКЕ ПРИЛОЖЕНИЯ .....		5
ОТЧЁТ ПО РЕЗУЛЬТАТАМ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЕ .....		6
ПРИЛОЖЕНИЕ.....		7

## ЦЕЛЬ РАБОТЫ

Освоение функций запуска и синхронизации нитей в при разработке многонитевых приложений в ОСРВ QNX .

### ЗАДАНИЕ 1.

Разработать приложение, состоящее из одного процесса с тремя запущенными нитями:

1. M(main).
2. T1(F1).
3. T2(F2).

В качестве нити M(main) выступает функция main(). Нити T1(F1) и T2(F2) запускаются нитью M(main) на базе соответственно функций F1() и F2(). Все три нити, работая параллельно, должны совместно динамически сформировать текст вида:

```
"Text0, Text1, Text2.\n"
```

#### Порядок работы приложения

Вначале нить M(main) запускает первой нить T1(F1), затем - T2(F2), передавая им в качестве параметра указатель совместно формируемого текста.

Далее нить M(main), записывая в текст букву за буквой, формирует свою часть текста:

```
"Text0, "
```

После формирования нитями всего текста нить M(main) выдаёт его на печать и завершает свою работу.

Запущенная нить T1(F1) должна тем же способом добавить в формируемый текст свою часть:

```
"Text1, "
```

После завершения записи своей части текста нить T1(F1) ожидает записи в буфер своей части текста нитью T2(F2), после чего завершает свою работу.

Запущенная нить T2(F2) должна так же добавить в формируемый текст свою часть:

```
"Text2.\n "
```

после чего завершает свою работу.

*Замечание.* Для имитации времени записи в текст одной буквы использовать пустой цикл в 1000 итераций.

Варианты:

1.	Блокировки чтения/записи. Мутексы. Присоединение.
2.	Блокировки чтения/записи. Мутексы. Барьеры.
3.	Блокировки чтения/записи. Ждущие блокировки. Присоединение.
4.	Блокировки чтения/записи. Ждущие блокировки. Барьеры.
5.	Блокировки чтения/записи. Условные переменные. Присоединение.
6.	Блокировки чтения/записи. Условные переменные. Барьеры.
7.	Неименованные семафоры. Условные переменные. Присоединение.
8.	Неименованные семафоры. Условные переменные. Барьеры.
9.	Именованные семафоры. Ждущие блокировки. Присоединение.

10.	Именованные семафоры. Ждущие блокировки. Барьеры.
11.	Мутексы. Ждущие блокировки. Присоединение.
12.	Мутексы. Ждущие блокировки. Барьеры.

### ЗАДАНИЕ 2.

Разработать приложение, состоящее из одного процесса с тремя запущенными нитями:

1. M(main).
2. T1(F1).
3. T2(F2).

В качестве нити M(main) выступает функция main(). Нити T1(F1) и T2(F2) запускаются нитью M(main) на базе соответственно функций F1() и F2(). Все три нити, работая параллельно, должны совместно динамически сформировать текст вида:

"Text0, Text1, Text2.\n"

#### Порядок работы приложения

Вначале нить M(main) запускает нить T1(F1), передавая ей в качестве параметра указатель совместно формируемого текста и функцию F2.

Далее нить M(main), записывая в текст букву за буквой, формирует свою часть текста:

"Text0, "

После формирования нитями всего текста нить M(main) выдаёт его на печать и завершает свою работу.

Запущенная нить T1(F1) запускает нить T2(F2), после чего тем же способом добавляет в формируемый текст свою часть: "Text1, "

После завершения записи своей части текста нить T1(F1) ожидает завершения выполнения нити T2(F2), после чего завершает свою работу.

Запущенная нить T2(F2) должна так же добавить в формируемый текст свою часть:

"Text2.\n "

после чего завершает свою работу.

*Замечание.* Для имитации времени записи в текст одной буквы использовать пустой цикл в 1000 итераций.

Варианты:

1.	Блокировки чтения/записи. Мутексы. Присоединение.
2.	Блокировки чтения/записи. Мутексы. Барьеры.
3.	Блокировки чтения/записи. Ждущие блокировки. Присоединение.
4.	Блокировки чтения/записи. Ждущие блокировки. Барьеры.
5.	Блокировки чтения/записи. Условные переменные. Присоединение.
6.	Блокировки чтения/записи. Условные переменные. Барьеры.
7.	Неименованные семафоры. Условные переменные. Присоединение.
8.	Неименованные семафоры. Условные переменные. Барьеры.
9.	Именованные семафоры. Ждущие блокировки. Присоединение.
10.	Именованные семафоры. Ждущие блокировки. Барьеры.
11.	Мутексы. Ждущие блокировки. Присоединение.

12.	Мутексы. Ждущие блокировки. Барьеры.
-----	--------------------------------------

### ЗАДАНИЕ 3.

Разработать приложение, состоящее из одного процесса с тремя запущенными нитями:

1. M(main).
2. T1(F1).
3. T2(F2).

В качестве нити M(main) выступает функция main(). Нити T1(F) и T2(F) запускаются нитью M(main) на базе общей функции F(). Все три нити, работая параллельно, должны совместно динамически сформировать текст вида:

```
"Text0, Text1, Text2.\n"
```

#### Порядок работы приложения

Вначале нить M(main) запускает первой нить T1(F), затем - T2(F), передавая им в качестве параметра указатель совместно формируемого текста и строку с номером нити - "1" или "2" соответственно.

Далее нить M(main), записывая в текст букву за буквой, формирует свою часть текста:

```
"Text0, "
```

После формирования нитями всего текста нить M(main) выдаёт его на печать и завершает свою работу.

Запущенная нить T1(F) должна тем же способом добавить в формируемый текст свою часть:

```
"Text1, "
```

После завершения записи своей части текста нить T1(F) ожидает завершения выполнения нити T2(F), после чего завершает свою работу.

Запущенная нить T2(F2) должна так же добавить в формируемый текст свою часть:

```
"Text2.\n "
```

после чего завершает свою работу.

*Замечание.* Для имитации времени записи в текст одной буквы использовать пустой цикл в 1000 итераций.

Варианты:

1.	Блокировки чтения/записи. Мутексы. Присоединение. Барьеры.
2.	Блокировки чтения/записи. Ждущие блокировки. Присоединение. Барьеры.
3.	Блокировки чтения/записи. Условные переменные. Присоединение. Барьеры.
4.	Неименованные семафоры. Условные переменные. Присоединение. Барьеры.
5.	Именованные семафоры. Ждущие блокировки. Присоединение. Барьеры.
6.	Мутексы. Ждущие блокировки. Присоединение. Барьеры.

### ТРЕБОВАНИЯ К РАЗРАБОТКЕ ПРИЛОЖЕНИЯ

При разработке приложения необходимо обязательно применить все указанные в варианте задания методы синхронизации, даже если их применение будет выглядеть несколько

искусственным. В дополнение к указанным в варианте задания можно обоснованно использовать и другие методы синхронизации.

## **ОТЧЁТ ПО РЕЗУЛЬТАТАМ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**

Отчёт по результатам выполнения лабораторной работы заключается в следующем:

1. Демонстрация работы приложения.
2. Проверка теоретических знаний по средствам запуска и синхронизации нитей.
3. Представление оформленного отчёта с результатами выполнения лабораторной работы.

По структуре отчет должен содержать:

- стандартно оформленный титульный лист (см. Приложение), □ текст задания с номером задания и номером варианта,
- исходный текст программного модуля с комментариями.

Отчет печатается на бумаге формата А4 через один интервал размер 12пт.

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»**

**ИНСТИТУТ ИНФОРМАТИКИ И КИБЕРНЕТИКИ**  
Кафедра программных систем

Дисциплина  
**Технологии промышленного программирования**

**ОТЧЁТ**  
по лабораторной работе

**Запуск и синхронизация нитей**

Вариант № \_\_\_\_

Студент: Фамилия И.О.

Группа: Номер

Преподаватель: Фамилия И.О.

Самара 20 \_\_\_\_