Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра ПМиК

Расчетно-графическое задание

по дисциплине

«Операционные системы реального времени»

Выполнил: студент 4 курса

ИВТ, гр. ИП-813

Стояк Ю.К.

Проверил: Преподователь кафедры ПМиК

Белевцова Екатерина Андреевна

.

Новосибирск 2021

Оглавление

[Задание 3](#_Toc59733303)

[Выполнение 4](#_Toc59733304)

[Вывод 14](#_Toc59733305)

# Задание

1. Сравните время запуска (создания) нити и время активизации с помощью семафора заранее созданной нити.

2. Функции типа fwrite(), работающие через структуру FILE, используют внутреннюю буферизацию данных. Анализируя время выполнения функций, определите размер буфера

# Выполнение

1. Для выполнения первого задания была реализована программа с использованием функций pthread\_create, sem\_init, sem\_wait, sem\_post, для отслеживания времени был выбран ClockCycles.

int pthread\_create( pthread\_t\* thread, const pthread\_attr\_t\* attr, void\* (\*start\_routine)(void\* ), void\* arg) –создаёт нить на которой выполняется указанная при создании функция ;

int ClockCycles() – функция из библиотеки sys/neutrino.h, возвращает текущее значение 64-битного счетчика циклов процессора. Чтобы вычислить количество прошедших секунд необходимо сначала вычислить количество циклов в секунду у системы с помощью SYSPAGE\_ENTRY(qtime)->cycles\_per\_sec и уже поделить наши циклы на это количество.

Замер был произведен на создание нити с началом замера времени, вызовом sched\_yield() и печатью строки после её вызова - void \*testThread(void \*args). Также были созданы нити с функцией void \* testThread2(void \*args), которая останавливает замер времени .

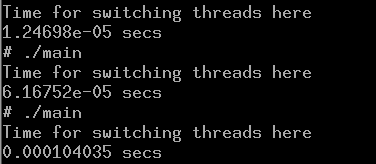
Как показали результаты замеров и сравнения времени функция прерывания сработала правильно, время замены нитей минимально (с учётом времени на генерацию второй нити.

1. Для выполнения второго задания в программе также использовалась функция mmap для создания страниц памяти.

В программе выделяются две страницы памяти. Разница в них только в теге PROT\_NOCACHE. Исходя из документации. флаг PROT\_NOCACHE следует использовать для предотвращения получения устаревшей, то есть закэшированной памяти при получения доступа к памяти, которая может быть изменена аппаратно.

# Результаты работы программ

**Main1**

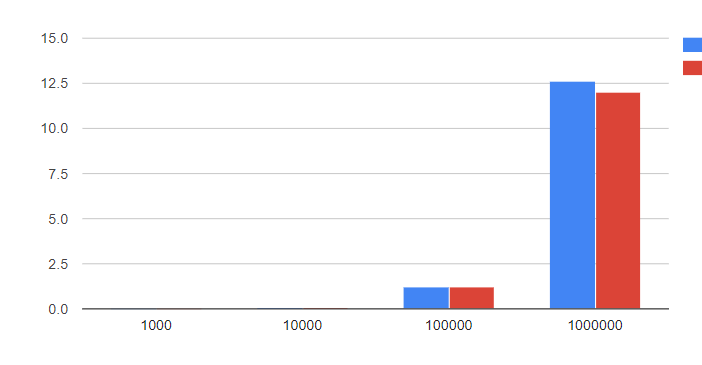


**Main2**



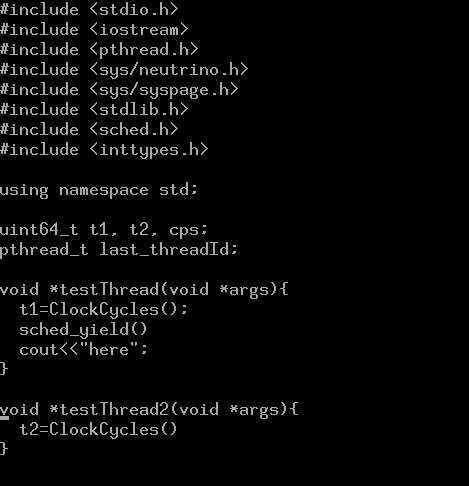
**Строки – время**

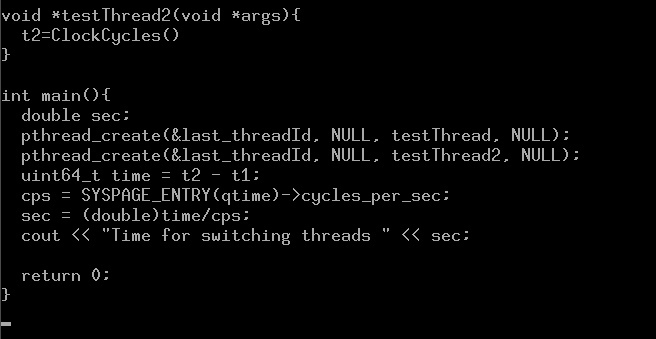
**Столбцы – кол-во итераций**



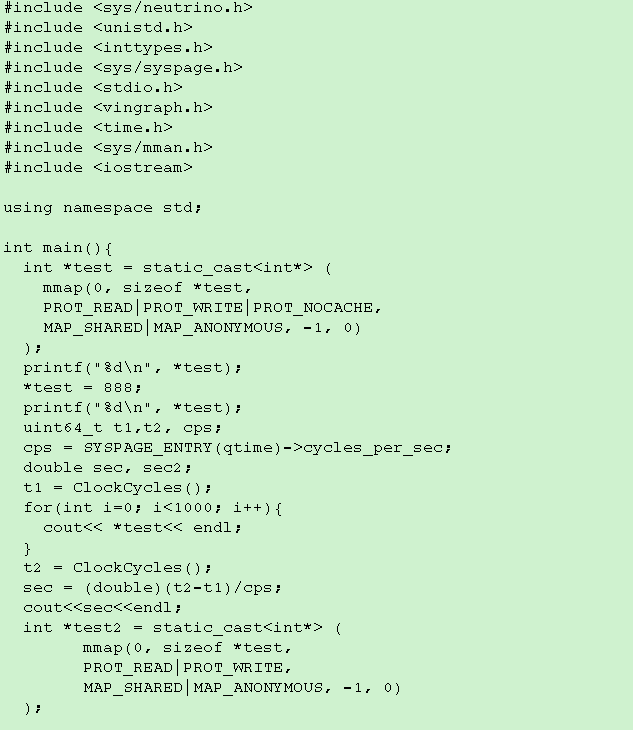
# Листинг

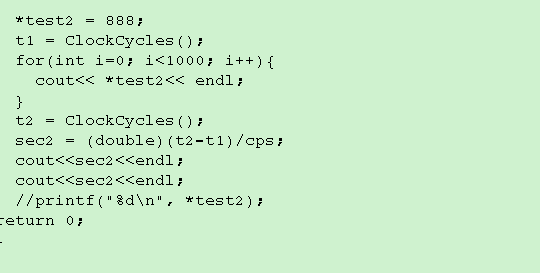
**main1.cpp**





**main2.cpp**





# Вывод

В рамках данного курса я узнал о операционных системах реального времени и научился работать в одной из них, QNX, узнал о принципе системы, чем она отличается от других систем.

В процессе выполнения расчетно-графического задания я углубил свои теоретические знания об ОСРВ, в частности о QNX. Больше узнал об алгоритме планирования и распределения ресурсов системы. Узнал о том, как измерить время работы части программы в QNX.