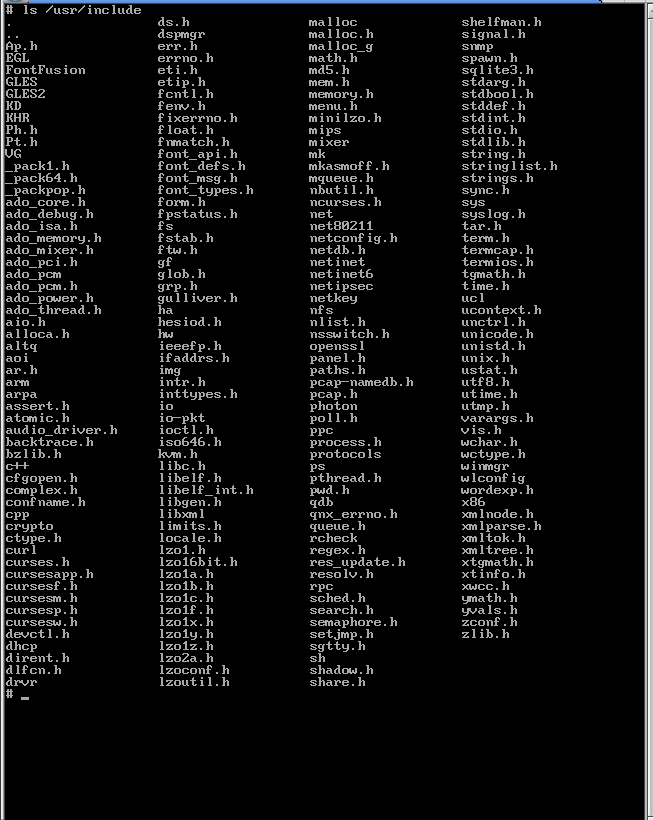
Лабораторная работа №4

1.1



1)pthread.h: Предназначен для работы с потоками, содержит объявления функций и типов для работы с потоками POSIX (мьютексы, условные переменные и т.д.).

2)semaphore.h: Предназначен для работы с семафорами, которые используются для управления доступом к общим ресурсам.

3)mqueue.h: Содержит функции для работы с очередями сообщений, которые используются для обмена сообщениями между процессами.

4)sched.h: Предназначен для управления планировкой потоков, содержит функции для изменения атрибутов планирования.

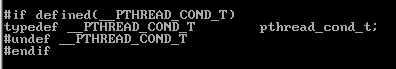
5)neutrino.h: Содержит объявления функций и структур, специфичных для QNX Neutrino, таких как системные вызовы и управление процессами.

1.2.

Мьютексы(pthread\_mutex\_t)

{C552AAF7-2035-429C-92C3-1377E8175932}

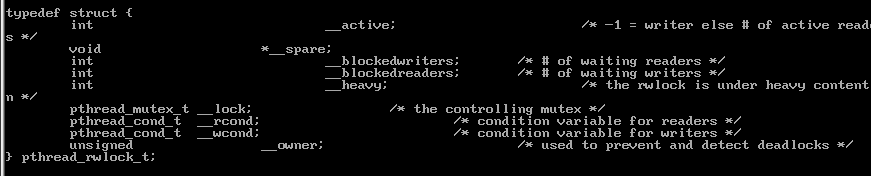
Условные переменные(pthread\_cond\_t)



Ждущие блокировки(pthread\_barrier\_t)



Блокировки чтения/записи(pthread\_rwlock\_t)



Семафоры

{41376347-FD21-4D51-8776-9395C98556C1}

FIFO

{9B176DE1-F4F5-4960-818C-53EDC91394A8}

Очереди сообщений

{FA7F999C-ED62-488C-BDA5-359A742B8F47}

2.1. Нажимаем в верхнем правом углу кнопку Help, нас перебрасывает в справочный центр

В поле для ввода текста вводим pthread\_create() и жмём enter

Пролистываем справку вниз и копируем код из Example

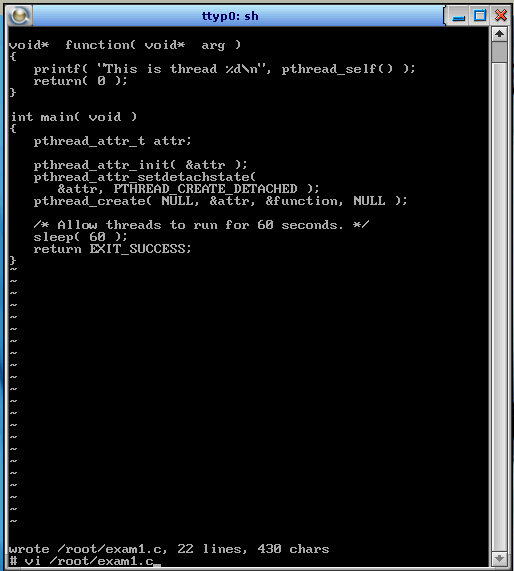
Открываем терминал

В терминале пишем vi /root/exam1.c

Нажимаем кнопку i для перехода в режим добавления текста

С помощью правой кнопки мыши вставляем скопированный код

Нажимаем Esc и пишем :wq



1 #include <stdio.h> // Подключение стандартной библиотеки для ввода-вывода

2 #include <stdlib.h> // Подключение стандартной библиотеки для работы с памятью и общими функциями

3 #include <pthread.h> // Подключение библиотеки для работы с потоками

4 void\* function(void\* arg) { // Определение функции, которая будет выполняться в потоке

5 printf("This is thread %d\n", pthread\_self()); // Выводит идентификатор текущего потока

6 return (0); // Завершение работы функции потока

7 }

8 int main(void) { // Главная функция программы

9 pthread\_attr\_t attr; // Объявление переменной для хранения атрибутов потока

10 pthread\_attr\_init(&attr); // Инициализация атрибутов потока значениями по умолчанию

11 pthread\_attr\_setdetachstate(&attr, PTHREAD\_CREATE\_DETACHED); // Установка атрибутов для отсоединенного состояния потока

12 pthread\_create(NULL, &attr, &function, NULL); // Создание потока с заданными атрибутами и функцией выполнения

14 sleep(60); // Задержка выполнения основной программы на 60 секунд для наблюдения за работой потока

15 return EXIT\_SUCCESS; // Завершение программы с кодом успешного выполнения

16 }

Сам код в примере немного неверен

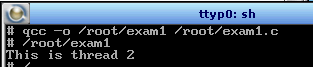
В шестой строке вместо return ( 0 ) нужно написать return NULL

А в пятнадцатой вместо return EXIT\_SUCCESS нужно написать return 0

Дальше компилируем файл с помощью qcc -o /root/exam1 /root/exam1.c

И запускаем с помощью

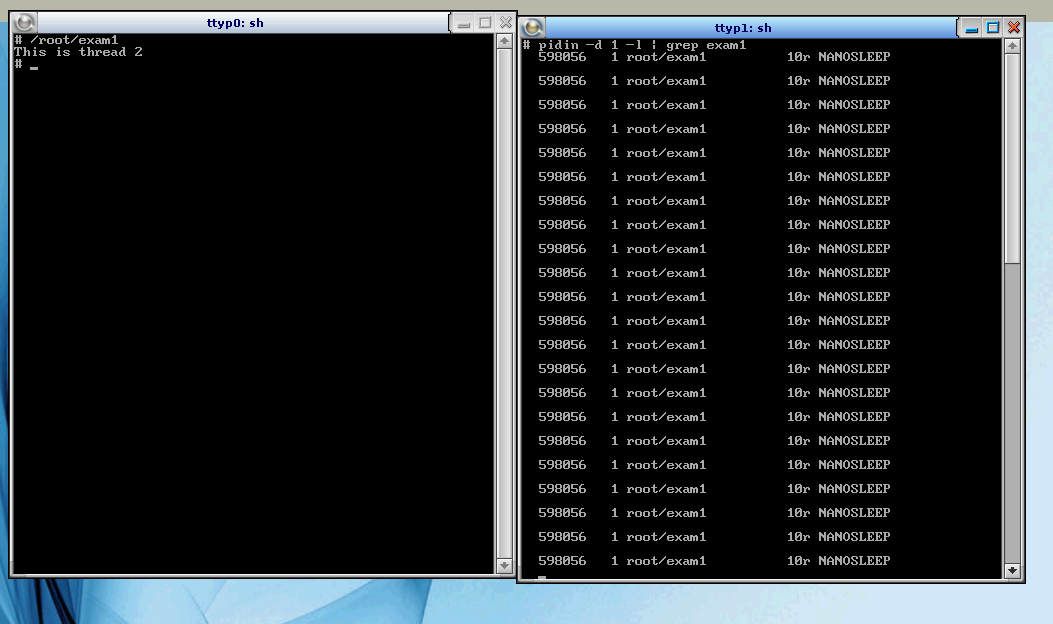
/root/exam1



3

Запускаем два терминала и вводим команды

Запускаем exam1 и утилиту pidin



pidin — утилита, используемая для отображения информации о процессах и потоках в ОС QNX.

-d 1 — задает задержку в 1 секунду между обновлениями данных о процессах. Это полезно для мониторинга в реальном времени.

-l — отображает подробную информацию о потоках, включая их состояние, идентификаторы, и другие параметры.

| grep exam1 — фильтрация вывода, чтобы показать только информацию, относящуюся к процессу exam1

3.4

Объяснение вывода команды pidin:

598056 — это идентификатор процесса (PID) exam1.

1 — это идентификатор потока (TID), указывающий, что это основной поток процесса exam1.

root/exam1 — это имя процесса и пользователя, под которым запущен процесс exam1.

10r — это приоритет потока. Число 10 показывает текущий приоритет, а суффикс r (roundrobin(карусель)) указывает на тип приоритета.

NANOSLEEP — это текущее состояние потока, показывающее, что поток находится в режиме ожидания или сна, вызванного функцией sleep().