**Операционные системы и системное программирование**

**Лабораторная работа № 2**

2. Класс "ThreadPool"

Разработать переносимый (portable) класс ThreadPool.

Объект этого класса содержит N потоков, находящихся в состоянии ожидания. К объекту можно обратиться с запросом на выполнение некоторого метода в потоке. Для выполнения процедуры объект класса ThreadPool подключает поток из пула ожидающих потоков. При создании объекта класса ThreadPool вводится максимальное число создаваемых потоков (K, K >= N).

Предусмотреть логгирование процесса работы ThreadPool. Файл логов должен включать в себя следующие сообщения:

1) создание N потоков при инициализации ThreadPool;

2) сообщение при добавлении новой задачи;

3) сообщения об ошибке выполнения задачи;

4) предупреждения, полученные при работе ПС.

В случае возникновения ошибок при работе пользовательской функции, сообщение об ошибке выводится в файл логов. Поток завершает обработку пользовательской функции и освобождается.

Задание на 8 баллов:

Часть 1.

Для управления процессом добавления задач, а также управления существующими потоками, используется один из следующих вариантов:

1) создается отдельный поток, управляющий потоками-рабочими;

2) основной поток пула, добавляющий новые задачи, контролирует процесс создания/удаления новых потоков;

3) потоки из пула сами занимаются планированием и выборкой задач, созданием и удалением потоков пула.

Часть 2.

При инициализации объекта ThreadPool указывается кол-во потоков N, которые создаются (доступны для использования изначально). Должно быть предусмотрено кол-во по-умолчанию.

При достижении максимального количества используемых потоков выполняется одно из следующих действий:

1) выводится сообщение об ошибке (превышение максимально допустимого количества работающих потоков). Пользовательская задача не добавляется.

2) выводится предупреждение о превышение максимально допустимого количества работающих потоков. Ожидается освобождение одного из работающих потоков, при освобождении выполняется пользовательская задача.

3) выводится предупреждение о превышение максимально допустимого количества работающих потоков. Увеличивается кол-во доступных для работы потоков (создаются новые).

Часть 3.

Задание на 10 баллов:

1) Разработанный класс ThreadPool должен работать одинаково на ОС Windows и Linux.

2) Предусмотреть возможность управления приоритетами задач и потоков.

Обязательно на любую оценку:

Создать приложение, демонстрирующее работу переносимого класса ThreadPool. В приложении должна быть продемонстрирована работа ThreadPool в случае превышения числа доступных потоков.

Проект должен разрабатываться с использование системы контроля версий git/svn (рекомендуется git).

Все функции/процедуры, большие блоки кода должны содержать комментарии. Язык комментариев - английский.

Варианты заданий для студентов выбираются с учетом выбора потока-планировщика (Часть 1), поведения при достижении максимального допустимого числа потоков (Часть 2) и дополнительного задания (Часть 3).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Номер варианта | Часть 1 | Часть 2 | Часть 3 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 1 | 1 | 1 | 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 2 | 1 | 2 | 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 3 | 1 | 3 | 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 4 | 1 | 1 | 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 5 | 1 | 2 | 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 6 | 1 | 3 | 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 7 | 2 | 1 | 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 8 | 2 | 2 | 1 |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 9 | 2 | 3 | 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 10 | 2 | 1 | 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 11 | 2 | 2 | 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 12 | 2 | 3 | 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 13 | 3 | 1 | 1 |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 14 | 3 | 2 | 1 |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 15 | 3 | 3 | 1 |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 16 | 3 | 1 | 2 |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 17 | 3 | 2 | 2 |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| 18 | 3 | 3 | 2 |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_