

Практические задания

Общие требования

Во всех заданиях необходимо разработать программу, которая моделирует работу одного из механизмов многозадачных ОС. Программы могут быть как оконными, так и консольными (с простейшим диалогом на основе меню). Язык реализации – любой по выбору: C/C++, C#, Java, Pascal/Delphi.

Тема 1. Моделирование работы планировщика потоков

Задание 1

Написать программу, моделирующую работу простейшего **бесприоритетного** циклического планировщика потоков в соответствии со следующими условиями:

- 1) каждый поток может находиться в одном из трех состояний – готовности к выполнению, выполнения, ожидания;
- 2) приоритеты потоков не используются;
- 3) все имеющиеся в системе ресурсы условно перенумерованы от 1 до 5;
- 4) основные события: создание нового потока, завершение активного потока, блокирование активного потока, разблокирование одного или нескольких потоков, завершение кванта времени для активного потока;
- 5) создание нового потока: вывод информационного сообщения и добавление идентификатора потока в конец списка готовых потоков;
- 6) завершение активного потока: вывод информационного сообщения и активизация первого готового к выполнению потока;
- 7) блокирование активного потока: имитация запроса занятого ресурса со случайным номером, изменение состояния потока и активизация нового потока;
- 8) разблокирование потоков: имитация освобождения ресурса со случайным номером и изменение состояния всех потоков, ждущих освобождения данного ресурса;

- 9) завершение кванта активного потока: включение его в конец списка и активизация нового потока;
- 10) события задаются пользователем с выводом текущего состояния системы;
- 11) если готовых к выполнению потоков нет, выводится сообщение о бездействии системы.

Задание 2

Написать программу, моделирующую работу простейшего **приоритетного** планировщика потоков в соответствии со следующими условиями:

- 1) потоки могут находиться **только в двух состояниях** – готовности к выполнению и выполнения;
- 2) каждый поток имеет приоритет от 1 (max) до 5 (min), генерируемый случайным образом при создании потока;
- 3) запросы ресурсов потоками не производятся;
- 4) все готовые к выполнению потоки находятся в своих приоритетных очередях;
- 5) для выполнения выбирается первый по порядку поток в наиболее приоритетной непустой очереди;
- 6) при появлении нового потока с более высоким приоритетом выполнение текущего потока приостанавливается;
- 7) при приостановке выполнения текущего потока по причине истечения кванта его приоритет **уменьшается** на 1 и он помещается в конец соответствующей очереди;
- 8) при приостановке выполнения текущего потока по причине появления более приоритетного потока его приоритет **увеличивается** на 1 и он помещается в конец соответствующей очереди;

9) все события (запуск нового потока, завершение кванта времени для текущего активного потока, полное завершение выполнения потока) задаются пользователем с выводом текущего состояния системы.

Алгоритм обработки события создания нового потока:

- сравнить приоритет нового потока с приоритетом выполняемого потока;
- при необходимости переключить потоки или поместить новый поток в конец соответствующей очереди.

Алгоритм обработки события завершения текущего активного потока:

- выбрать для выполнения очередной поток;
- если все очереди пусты, вывести сообщение о бездействии системы.

Алгоритм обработки события истечения кванта времени:

- поместить прерываемый поток в конец соответствующей очереди;
- выбрать для выполнения очередной поток.

Задание 3

Написать программу, объединяющую оба предыдущих задания и моделирующую работу приоритетного планировщика с тремя возможными состояниями потоков.