

Системное программирование

Л.р.6. Взаимодействие процессов (потокaв): взаимное исключение и синхронизация

Цель:

Подходы, системные объекты и функции для обеспечения синхронизации и передачи управления между взаимодействующими процессами.

Типичные проблемы, возникающие при организации взаимодействия, модели для их описания, пути их решения.

Теоретическая и методическая часть

Задачи взаимного исключения (предотвращения некорректного совместного доступа к критическим ресурсам) и синхронизации – в числе основных задач взаимодействия.

Критический ресурс и критическая секция.

Проблема атомарности доступа для проверки и модификации глобальных объектов, используемых для синхронизации.

Теоретические модели для описания ситуаций синхронизации.

Механизм CriticalSection; функции ожидания WaitFor***() и объекты ожидания: Mutex, Semaphore, Event, WaitableTimer; барьеры; спин-блокировки и т.д.

Практическая часть

Общая постановка задачи:

Приложение, демонстрирующее параллельную согласованную работу процессов (потокaв) и их взаимодействие. Анализ корректности (отсутствия коллизий). Оценка эффективности механизмов синхронизации (ISO).

Специальных требований к приложениям не предъявляется; в частности, во многих случаях они могут быть не обязательно оконными, но также и консольными.

Варианты заданий:

- Модель «обедающие философы»
- Модель «Писатели-читатели»
- Система массового обслуживания
- Параллельная обработка данных в разделяемой памяти
- ...

1 Модель «обедающие философы»

Реализация модели взаимодействия конкурирующих параллельных процессов (потокaв) «обедающие философы» с возможностью ее

Системное программирование: Л.р.6 – Взаимодействие процессов (поток): взаимное исключение и синхронизация

параметризации и наглядного (не обязательно графического) представления результатов.

Проверка и демонстрация различных подходов к решению задачи обеспечения корректного взаимодействия.

Изменяемые параметры модели: количество «философов»; выбор логики разрешения конфликта; интенсивность обращений к критическому ресурсу, длительность использования ресурса и т.п. (характеристики случайных величин при моделировании); величина тайм-аутов и др.

Результаты моделирования: соотношение времени активности/блокировки для отдельных «философов» и модели в целом; соотношение успешных/неуспешных обращений; общая эффективность (пропускная способность); фиксация возникновения (и разрешения, если это предусмотрено) тупиков и др.

2 Модель «Писатели-читатели»

Реализация модели взаимодействия процессов (поток) «писатели-читатели» с возможностью параметризации и наглядного (не обязательно графического) представления результатов.

Поскольку речь идет о модели, реальные данные не обязательны, можно ограничиться моделированием обращений к ним (обращение характеризуется параметрами запроса, моментом обращения, длительностью исполнения). Но можно реализовать и макет системы, использующей реальные (тестовые, «учебные») данные в соответствии с этой моделью. Таким образом, вместо моделирования обращений с заданными характеристиками могут быть сами обращения к «настоящим» данным (прикладные функции системы можно позаимствовать из предыдущих работ).

Обеспечение корректного функционирования, т.е. избегание как «грязного» считывания данных и одновременно минимизация блокировок.

Изменяемые параметры модели: количество «писателей» и «читателей»; интенсивность их обращений к ресурсам, длительность использования ресурса, размер блока данных и т.п. (характеристики случайных величин при моделировании); штрафы за простой участников; величина тайм-аутов и др.

Результаты моделирования: соотношение времени активности/блокировки участников; соотношение успешных/неуспешных обращений; общая эффективность (пропускная способность) по «записи» и «чтению» и др.

3 Система массового обслуживания

Реализация модели системы массового обслуживания (СМО) из 3-4 (по желанию можно больше) ступеней, в которой «каналы» (устройства) обработки данных в каждой ступени представлены отдельными процессами (поток), с возможностью параметризации и наглядного (не обязательно графического) представления результатов.

Точное следование определенному типу СМО не обязательно. В ступенях предполагается наличие нескольких «каналов», конкурирующих за поступающие заявки. В ступнях возможна логика «барьерной» синхронизации: ожидание освобождения всех каналов и накопление достаточного количества заявок, чтобы загрузить сразу все каналы.

Необходимо обеспечить синхронизацию функционирования «каналов» и ступеней, разрешение коллизий при доступе к буферам (очередям) заявок и к общим функциям модели (например, сбор статистики о работе элементов модели).

Изменяемые параметры модели: количество ступеней и каналов в них; емкость буферов между ступенями; логика отбрасывания заявок или блокировки «каналов»; интенсивность потока заявок и количественные характеристики их обработки (характеристики случайных величин при моделировании); штрафы за необработанные заявки; величина таймаутов и др.

Результаты моделирования: пропускная способность системы; частота потерь заявок; загрузка элементов системы (доля времени простоя каналов и заполнение накопителей) и др.

4 Параллельная обработка данных в разделяемой памяти

Организация доступа к содержимому разделяемой памяти для параллельно выполняющихся процессов (потокaв) с предотвращением коллизий. Близость к модели «писатели-читатели».

Эксперименты (тесты) с целью определения количественных характеристик – прогоны с разными параметрами системы.

Примеры практического использования: файл (банк/база данных), разделяемый несколькими процессами (близко к модели «писатели-читатели»); высокопроизводительный буфер «контейнеров» с данными для использования множеством потребителей без перераспределения памяти для «контейнеров и без копирования их содержимого.

Пример подхода к реализации: фрагментация памяти на блоки фиксированного (или переменного) размера, предоставляемые потребителям (процессам, потокам); занятый блок временно блокируется для доступа со стороны других потребителей. Массив (список) блоков, массив (список) семафоров (мьютексов), управляющих доступом к блокам. API для запроса, освобождения и др. действий с блоками.

Изменяемые параметры модели: общий объем разделяемой памяти; размер (количество) фрагментов; количество потребителей; характеристика запросов к памяти и др.

5 ...