

Курсовой проект по курсу «Операционные системы»

Цель курсового проекта

1. Приобретение практических навыков в использовании знаний, полученных в течении курса
2. Проведение исследования в выбранной предметной области

Задание

Необходимо спроектировать и реализовать программный прототип в соответствии с выбранным вариантом. Произвести анализ и сделать вывод на основании данных, полученных при работе программного прототипа.

Возможные варианты курсового проекта:

Аллокатеры памяти

Исследование 2 аллокатеров памяти: необходимо реализовать два алгоритма аллокации памяти и сравнить их по следующим характеристикам:

- Фактор использования
- Скорость выделения блоков
- Скорость освобождения блоков
- Простота использования аллокатора

Каждый аллокатор памяти должен иметь функции аналогичные стандартным функциям free и malloc (realloc, опционально). Перед работой каждый аллокатор инициализируется свободными страницами памяти, выделенными стандартными средствами ядра. Необходимо самостоятельно разработать стратегию тестирования для определения ключевых характеристик аллокатеров памяти. При тестировании нужно свести к минимуму потери точности из-за накладных расходов при измерении ключевых характеристик, описанных выше.

В отчете необходимо отобразить следующее:

- Подробное описание каждого из исследуемых алгоритмов
- Процесс тестирования
- Обоснование подхода тестирования
- Результаты тестирования
- Заключение по проведенной работе

Виртуальная память

Необходимо создать рабочую модель виртуальной памяти. Требуется взять алгоритм аллокации памяти и использовать его совместно с алгоритмом замещения страниц. При разработке предусмотреть следующий интерфейс использования:

- CreateMemory
- Malloc
- Free

Параметры функций могут отличаться в зависимости от выбранных алгоритмов. Исследовать полученную модель виртуальной памяти – ее плюсы и минусы. Возможно сделать данные алгоритмы, переопределив стандартные механизмы аллокации языка C++. Также проработать модель тестирования полученного программного решения.

Сервер сообщений

Необходимо создать собственный сервер сообщений. В качестве механизма передачи сообщений возможно использовать следующие:

1. Pipes
2. Sockets
3. Files/Shared memory

При работе с сервером сообщений необходимо предусмотреть следующие механизмы (в зависимости от варианта):

- Долговременное хранение сообщений
- Транзактивность
- Система имен очередей
- Возможность настройки переадресации сообщений по фильтрам
- Приоритеты сообщений
- И другие возможные, в зависимости от варианта

Основные операции, которые должен поддерживать сервер сообщений и библиотека по работе с ним:

- CreateQueue
- DeleteQueue
- ConnectToQueue
- Push (Send)
- Pop/Top (Receive)

Текстовый препроцессор

Проработать полноценный текстовый препроцессор. За основу можно взять такие текстовые препроцессоры, как `ed`, `sed` и прочие. Основные особенности текстового препроцессора, которые могут быть реализованы (в зависимости от варианта):

- Транзактивность операций
- Кеширование при поиске
- Поддержка регулярных выражений
- Использование технологии MemoryMap
- И др.

Планирование процессов и потоков

Необходимо создать и протестировать библиотеку, реализующую функционал по работе с потоками. Библиотека должна быть надстройкой над существующими системными вызовами. Возможные особенности библиотеки (в зависимости от варианта):

- Приоритезация потоков

- Решение взаимоблокировок
- Упорядочивание ресурсов
- Удобная обертка при работе с несколькими аргументами при создании потоков
- Расписание потоков
- И другие

Создание клиента для передачи мгновенных личных сообщений

На основе любой из выбранных технологий:

- Pipes
- Sockets
- Сервера очередей
- И другие

Создать собственный клиент быстрых сообщений (возможно, и сервер — зависит от выбранной архитектуры), который бы работал в рамках сети. Также есть возможность создания собственного клиента быстрых сообщений для уже существующих систем обмена сообщениями (например, telegram). Сервер или клиент передачи мгновенных сообщений должны обладать дополнительным функционалом — шифрование, эффективное хранение истории, подписки и др.

Проектирование интерактивной (консольной | графической) клиент-серверной игры

На основе любой из выбранных технологий:

- Pipes
- Sockets
- Сервера очередей
- И другие

Создать собственную игру для нескольких пользователей. Игра может быть устроена по принципу: клиент-клиент, сервер-клиент.

Возможные варианты игр:

1. Аналог «Bomberman»
2. Аналог «Распан»
3. И др.

Вариант на «удовлетворительно»

Необходимо написать 3-и программы. Далее будем обозначать эти программы А, В, С.

Программа А принимает из стандартного потока ввода строки, а далее их отправляет программе С. Отправка строк должна производиться построчно. Программа С печатает в стандартный вывод, полученную строку от программы А. После получения программа С отправляет программе А сообщение о том, что строка получена. До тех пор пока программа А не примет «сообщение о получении строки» от программы С, она не может отправлять следующую строку программе С.

Программа В пишет в стандартный вывод количество отправленных символов программой А и количество принятых символов программой С. Данную информацию программа В получает от программ А и С соответственно.

Способ организация межпроцессорного взаимодействия выбирает студент.