

Лекция 13. Подсистема ввода-вывода. Сетевые операционные системы.

Планирование запросов

Алгоритмы разрешения ситуации, при которых доступ к устройству ввода-вывода хотят получить сразу несколько устройств.

Целью планирования запросов является минимизация полного времени удовлетворения всех существующих запросов. Время позиционирования разделяется на время, для поворота головки на нужный цилиндр и время, необходимое для того, чтобы пакет пластин повернулся так, чтобы под головками оказался требуемый сектор. Время передачи информации не зависит от сектора.

Единственный параметр для планирования -- время поиска цилиндра, величина -- количество пройденных цилиндров в результате удовлетворения всех запросов.

FCFS

удовлетворение запросов в порядке их поступления. Однако это очень неэффективный алгоритм.

SSTF

Следующий запрос -- запрос, для которого нужно прокрутить минимальное число цилиндров. Недостатки: если мы находимся в середине диска и у нас есть несколько запросов, то после попадания в конец одной половины нужно пройти очень много цилиндров чтобы вернуться к другой половине.

Осмысленное сканирование

Двигаемся в одном направлении и удовлетворяем все запросы. Если понятно, что в данном направлении уже не будет запросов, то мы меняем направление.

Циклическое сканирование

Модификация этого алгоритма. Головка не меняет направление, а перемещается из начала диска в его конец (или наоборот) и продолжает движение.

Сетевые операционные системы

Автономно функционирующие вычислительные системы могут объединяться в сеть.

Причины:

- Совместное использование ресурсов.
- Ускорения вычислений.
- Повышение надежности.
- Необходимость пользователей в общении между собой.

Два подхода к построению ОС:

- Построение сетевых операционных систем. ОС, в которую добавлена небольшая часть, обеспечивающая взаимодействие с сетевым интерфейсом.
- Распределенные операционные системы.

Отличия взаимодействия локальных и удаленных процессов:

- Локальные процессы всегда взаимодействуют посредством разделяемой памяти
- Информация от отправителя к получателю в удаленных процессах проходит через промежуточные узлы
- Передача информации в рамках одного компьютера обладает почти 100% надежностью. (Взаимодействие удаленных процессов нужно рассматривать как взаимодействие, основанное на ненадежной связи)
- Адреса процессов должны быть уникальными в рамках всей сети
- Может возникать race condition

Понятие протокола

Сетевой протокол -- перечень правил, который определяет формат сообщений для обмена. Связывает один уровень.

Интерфейсы — это протоколы, которые связывают разные уровни

OSI-модель:

1. Физический
2. Канальный
3. Сетевой
4. Транспортный
5. Сеансовый
6. Представления
7. Приложения

Самая популярная реализация -- TCP/IP

Модели построения сетевых систем

Решение проблемы адресации, чтобы процессы имели уникальные адреса:

1. Одноуровневый подход -- процесс посылает запрос своей ОС с адресом, который он хочет занять. ОС посылает широковещательный запрос, для выяснения занят ли адрес.
2. Двухуровневая адресация -- каждый узел сети имеет свое уникальное имя, а каждый процесс на этом вычислительном комплексе имеет свой уникальный адрес
 - a. Централизованный подход -- есть некоторый выделенный компьютер, который при получении запроса от другого компьютера подбирает ему свободное имя, сохраняет в файл и регулярно рассылает всем другим компьютерам. Однако такой подход не подходит для больших сетей.
 - b. Распределенный подход -- компьютеры разбиваются, создавая древовидную структуру. Каждой области, состоящей более, чем из одного компьютера присваивается свой DNS-сервер.

Маршрутизация сети

Виды:

- От источника --полный путь от источника до получателя задается непосредственно отправителем
- Пошаговый -- решение о том куда переслать дальше пакет принимается с помощью таблицы маршрутизации

Алгоритмы пошаговой маршрутизации:

- Фиксированная маршрутизация -- таблицы задаются администратором
- Простая маршрутизация -- без таблиц
 - Случайная -- след. путь выбирается случайно.
 - Лавинная -- отправляется во все линии связи
 - По прецедентам -- запоминаются все прецеденты отправок сообщений, при повторной отправке используется запомненный маршрут
- Динамическая маршрутизация
 - Дистанционно векторный -- запоминается информация, кому нужно передать информацию. Соседи обмениваются этой информацией и накапливают полную таблицу с кратчайшими маршрутами
 - Граф связей -- на каждом узле строится база данных, содержащая граф всей сети

Интерфейсы для взаимодействия удаленных процессов:

- С помощью датаграмм
- С помощью установления логического соединения.