

# Лекция 10. Управление памятью. Управление файловыми системами.

## Пробуксовка. Trashing.

это высокая частота возникновения страничных нарушений, когда на подкачку страниц тратится больше времени, чем на выполнение команд в процессе.

Решение:

1. Метод границ -- вводятся отметки для максимальной и минимальной частоты страничных нарушений. При пересечении нижней границы частота начинает снижаться, при пересечении верхней границы процесс отбирает страницы у тех процессов, у которых частота меньше минимума. При трешинге всех страниц один из них временно выталкивается из оперативной памяти на диск, чтобы получить свободные кадры для других процессов.
2. Принцип пространственной локальности -- программа в один момент времени работает с ограниченным числом адресов. Переходный процесс из одной процедуры в другую потребует большого количества подгрузок страниц из памяти.

Имеется дело с множествами страниц -- процесс в какой-то момент времени либо работает в одном рабочем множестве, переходит между ними.

Разделение памяти на страницы позволяет реализовать разделяемую память.

## Общие сведения о файлах

Существуют 2 типа файлов:

- последовательного доступа -- в таком файле мы не можем получить произвольный кусок. Доступные операции для такого файла: get, put, rewind, getattribute и setattribute для доступа к метainформации
- прямого доступа -- в таком файле мы можем переместить указатель на любое место. Доступные операции: seek, write, read, getattribute и setattribute для доступа к метainформации

В файлах можно ввести внутреннюю структуру. Можно также ввести ключи, соответствующие блокам данных, и организовать индексно-последовательную организацию файла.

Структуру можно строить над файлами: должны существовать базовые операции для работы с несколькими файлами: new, free.

Если структуры файлов нет, то мы имеем дело с кучей, когда у файлов есть уникальное имя, но они не упорядочены.

Организация файловой системы — это способ отображения структуры файлов на внешний носитель вместе со способом логической организации файлов.

## Организация доступа к файлам

ОС должна уметь:

- Идентифицировать файлы -- определять по имени местонахождение
- Организовывать внешнюю память -- где/каким образом размещен файл
- Обеспечивать надежность хранения информации и отказоустойчивость
- Защита информации
- Обеспечение совместного доступа для некоторых процессов

Варианты размещения файлов:

1. Каждому файлу выделяется непрерывная последовательность блоков. Преимущества: для работы нужно знать только где начинается файл и его размер, чтобы работать с ним. Недостатки: фрагментация диска, при которой между файлами остается свободное пространство.
2. Индексный список -- файлы размещаются в непоследовательных блоках. В каждом блоке указывается адрес следующего. Недостатки: дополнительная память для хранения адресов и невозможность прямого доступа к памяти. Кроме того, при повреждении одного блока ведет к потере всего файла.
3. FAT (File Allocation Table) -- файлы размещаются в непоследовательных блоках при этом все адреса хранящиеся в специальной таблице. Преимущества: эту таблицу сразу можно загрузить в оперативную память. Недостатки: при повреждении этой таблицы теряется вся информация на диске. Решение: создают несколько копий такой таблицы.
4. Индексный узел (inode) -- каждому файлу присваивается структура с его атрибутами, а также списком адресов его блоков. все inodes выносятся в отдельный массив и этот массив также можно загрузить в оперативную память для быстрого доступа.