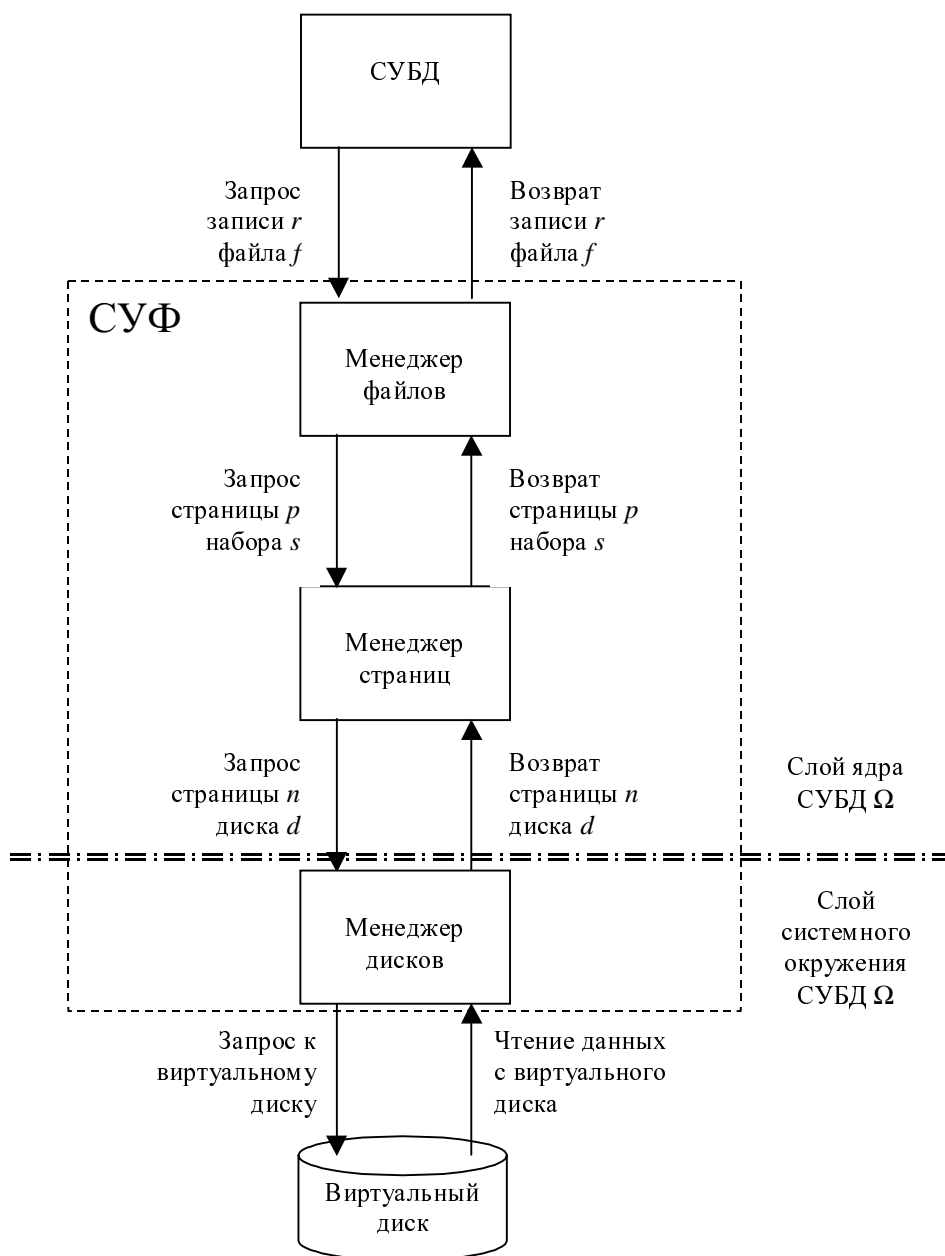
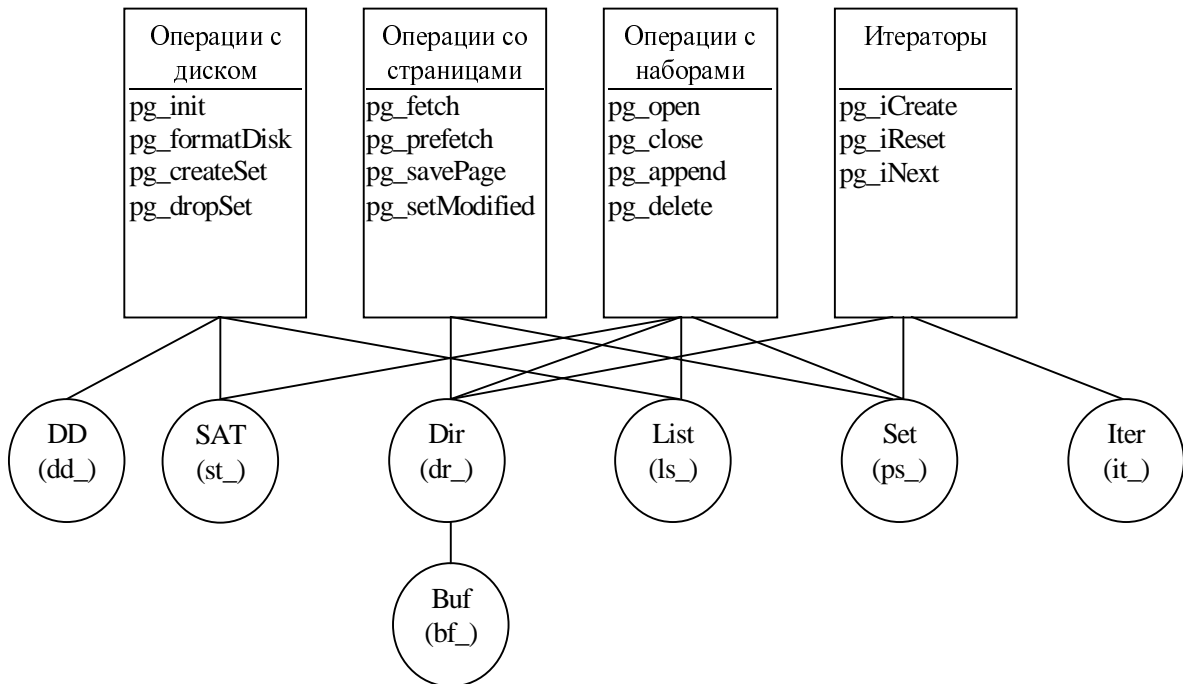


Общая структура СУФ



Иерархия модулей и объектов



- **DD** (Disk Directory) - заголовок диска;
- **SAT** (Set Allocation Table) - таблица размещения наборов;
- **Dir** (Buffer Directory)- избыточный индекс буферизованных страниц (индекс для **Buf**);
- **Buf** (Buffer) - буфер для буферизации страниц;
- **List** (Space List) - список свободного пространства (ССП);
- **Set** (Open Page Set Descriptor Table) - таблица дескрипторов открытых наборов;
- **Iter** (Iterator Table) - таблица дескрипторов итераторов.

Интерфейс Менеджера дисков

- /* Записать страницу на диск в асинхронном режиме */
int /* No. канала*/ **ds_write**(int /*номер страницы */, void * /* указатель на буфер */);
- /* Считать страницу с диска в асинхронном режиме */
int /* No. канала*/ **ds_read**(int /*номер страницы */, void * /* указатель на буфер */);
- /* Проверить завершение write/read операции */
int /* 1-да, 0-нет */ **ds_done**(int /* No. канала*/);
- /* Сброс содержимого виртуального диска в файл на host-машине */
void **ds_dump**(char * /* имя файла */);
- /* Загрузка содержимого виртуального диска из файла на host-машине */
void **ds_reset**(char * /* имя файла */);

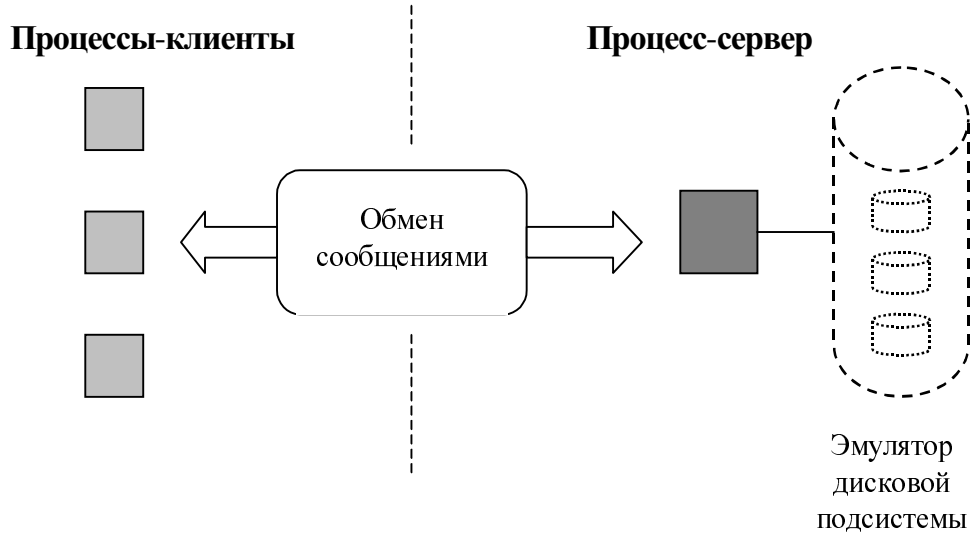
Интерфейс Менеджера файлов

- /* Создать пустой файл */
int /* >=0 - ОК, иначе - ошибка */ **fl_createFile**(int /* идентификатор файла */, int /* длина info */);
- /* Удалить файл */
int /* >=0 - ОК, иначе - ошибка */ **fl_dropFile**(int /* идентификатор файла */);
- /* Открыть файл */
int /* >=0 - ОК, иначе - ошибка */
fl_open(int /* идентификатор файла */,
char /* режим 0 (FL_READ) - только чтение, 1 (FL_WRITE) - запись и чтение */);
- /* Закрыть файл */
int /* >=0 - ОК, иначе - ошибка */ **fl_close**(int /* идентификатор файла */);
- /* Добавить запись в конец файла */
int /* >=0 - УИД первой добавленной записи, иначе - ошибка */ **fl_append**(int /* идентификатор файла */);
- /* Пометить запись на удаление */
int /* >=0 - ОК, иначе - ошибка */ **fl_delete**(int /* идентификатор файла */, int /* УИД записи */);
- /* Удалить все записи с пометкой на удаление */
int /* >=0 - ОК, иначе - ошибка */ **fl_pack**(int /* идентификатор файла */, int /* УИД записи */);
- /* Выборка записи */
void* /* <>NULL - указатель на info записи, иначе - ошибка */
fl_fetch(int /* идентификатор файла */, int /* УИД записи */);

Интерфейс Менеджера страниц

- */* Создать пустой набор страниц */*
int / >=0 - ОК, иначе - ошибка */* **pg_createSet**(*int /* идентификатор набора */*);
- */* Удалить набор страниц */*
int / >=0 - ОК, иначе - ошибка */* **pg_dropSet**(*int /* идентификатор набора */*);
- */* Открыть набор страниц */*
int / >=0 - ОК, иначе - ошибка */* **pg_open**(*int /* идентификатор набора */*);
- */* Закрыть набор страниц */*
int / >=0 - ОК, иначе - ошибка */* **pg_close**(*int /* идентификатор набора */*);
- */* Добавить n страниц в конец набора */*
int / >=0 - идентификатор первой добавленной страницы, иначе - ошибка */*
pg_append(*int /* идентификатор набора */*, *int /* количество страниц */*);
- */* Удалить страницу из набора */*
int / >=0 - ОК, иначе - ошибка */*
pg_delete(*int /* идентификатор набора */*, *int /* идентификатор страницы */*);
- */* Выборка страницы */*
*void * /* указатель на образ страницы в буфере */*
pg_fetch(*int /* идентификатор набора */*, *int /* идентификатор страницы, или NIL */*,
int / статический рейтинг */*);
- */* Выборка страницы с упреждением */*
int / >=0 - ОК, иначе - ошибка */*
pg_prefetch(*int /* идентификатор набора */*, *int /* идентификатор страницы */*,
int / статический рейтинг */*);
- */* Установка атрибута образа страницы modified */*
int / >=0 - ОК, иначе - ошибка */*
pg_setModified(*int /* идентификатор страницы */*, *char /* новое значение атрибута */*);

Архитектура реализации Менеджера дисков



Формат сообщения

ЗАГОЛОВОК				ИНФОРМАЦИОННАЯ ЧАСТЬ
ИД операции	тип операции	номер узла	...	

Модель реализации

- Такт - атомарная единица выполнения операции
- Между тактами может быть более одной диспетчеризации
- счетчик_такта = 0 : операция еще не выполнялась
счетчик_такта = -1 : операция уже завершена

Событие

Пользователь инициировал новую операцию

Завершение обменов по линкам или добавление записи в таблицу операций

Реакция

1-я фаза: Добавление записи в таблицу операций

- счетчик_такта := 0

2-я фаза: Обработка таблицы операций

для всех записей, где такт ≥ 0

- счетчик_такта++;
- выполнить действие в соответствии с тактом;

Алгоритм записи страницы на диск

Такт	Клиент.Записать_страницу_на_диск	Такт	Сервер.Записать_страницу_на_диск
1	если линк заблокирован по записи то счетчик_такта-- иначе заблокировать линк по записи; запустить запись заголовка сообщения;	1	если линк заблокирован по чтению то счетчик_такта-- иначе заблокировать линк по чтению; запустить чтение информационной части;
2	запустить запись информационной части	2	если завершилось чтение информационной части то счетчик_такта-- иначе заблокировать диск по записи; запустить запись информационной части на диск;
3	разблокировать линк по записи	3	если запись на диск завершилась то если линк свободен по записи то запустить запись заголовка с кодом "операция завершена" клиенту; счетчик_такта := -1; иначе счетчик_такта--
4	если пришел заголовок с сервера с кодом "операция завершена" то счетчик_такта := -1 иначе счетчик_такта--;		