

# Основные функции DC-API

## Серверная часть

**int DC\_initMaster (const char \*configFile)** инициализирует библиотеки в соответствии с конфигурационным файлом. Возвращает 0 при успешном завершении, 1 — при неуспешном). Данную функцию необходимо вызвать прежде вызова прочих dc-api функций. Конфигурационный файл содержит путь к рабочей папке приложения, идентификационный номер приложения, путь к папке проекта.

**void DC\_setMasterCb (DC\_ResultCallback resultcb, DC\_SubresultCallback subresultcb, DC\_MessageCallback msgcb)** устанавливает функции обработки сообщений о событиях происходящих на сервере, resultcb - функция обработки результата, subresultcb - подрезультата, msgcb - сообщений). В примере integral используется только функция обработки результата process\_result (её задача скопировать пришедшие на сервер файлы с результатами в рабочую папку проекта).

**DC\_log(LOG\_NOTICE, "Master: Creating work units"):** печать лога.

**DC\_Workunit\* DC\_createWU (const char \*clientName, const char \*arguments[], int subresults, const char \*tag)** – создает расчетный блок, client-name - название приложения-клиента, arguments - аргументы с которыми запускается приложение-клиент, subresults - подрезультаты, tag - строка маркировка WU (наиболее удобно под tag ставить символьное изображение номера WU). Tag используется для идентификации WU внутри самой распределенной программы, boinc-сервер и boinc-клиент значение tag не используют.

**int DC\_addWUInput(DC\_Workunit \*wu, const char \*logicalFileName, const char \*URL, DC\_FileMode fileMode)** – присоединяет входной файл к WU. LogicalFileName - логическое имя файла, URL - расположение самого входного файла, fileMode - режим файла(необходимый для оптимизации дискового пространства).

**int DC\_addWUOutput (DC\_Workunit \*wu, const char \*logicalFileName)** – присоединяет выходной файл к WU. LogicalFileName - логическое имя файла.

**int DC\_submitWU (DC\_Workunit \*wu)** - отправляет WU в grid-инфраструктуру.

**char\* DC\_getWUtag(const DC\_Workunit \*wu)** – возвращает маркировку полученного WU. Каждому WU при создании присписывается некоторая символьная строка. После возвращения WU из вычислительной инфраструктуры по данной маркировке распределённое приложение идентифицирует данный WU.

**char\* DC\_getResultOutput (const DC\_Result \*result, const char \*logicalFileName)** — возвращает реальный путь к физическому файлу конкретного результата по его логическому наименованию.

**void DC\_destroyWU (DC\_Workunit \*wu)** — уничтожает, уже созданный WU, т.е. освобождает ресурсы необходимые для его существования. Если WU пребывает в процессе обчёта, его выполнение прекращается.

### **Клиентская часть**

**int DC\_initClient (void)** - инициализирует клиентский API. Функция должна быть вызвана ранее всех других функций DC-API.

**void DC\_finishClient (int exitcode)** — завершает работу клиентской стороны приложения. Exitcode=0 — успех, exitcode=1 — неудача.

**char\* DC\_resolveFileName (DC\_FileType type, const char \*logicalFileName)** – по логическому имени файла функция возвращает его физическое имя, т.е. наименование файла внутри файловой системы клиентского компьютера. Type — тип файла (входящий/выходящий).

**DC\_ClientEvent\* DC\_checkClientEvent(void)** — возвращает клиентское событие.

**void DC\_destroyClientEvent (DC\_ClientEvent \*event)** — уничтожает клиентское событие.