Формат обмена между сервером и клиентом СУПиК

Условности:

Сокращённое наименование БД – принятое в СУПиКе сокращение баз данных SQL:

alt – altium, con – constructives, dev – devices, ent – enterprise, sch – schemagee, sol – solidworks, sup – supik, tb – tb, sadm – sysadm

Общий формат команды от клиента к серверу:

**<cmd> <numbytes> <subcmd> <arg\_0> <arg\_1> … <arg\_n>\n**

Где

Cmd – команда из списка табл. 1

Numbytes – количество байт в посылке, начиная с поля subcmd или с arg\_1, если поля subcmd нет

Subcmd – подкоманда в команде согласно списка табл. 1

Arg\_x – аргументы команды

\n – символ перевода строки

Ответ от сервера представляет собой набор строк, каждая из которых заканчивается символом перевода строки. После каждой строки от клиента ожидается подтверждение командой RDY (табл. 1), после чего сервер передаёт следующую строку. Окончание передачи определяется по приходу заданного количества байт.

Общий формат ответа:

<numrecords> <arg\_01> <arg\_02> … <arg\_0n>\n

<arg\_11> <arg\_12> … <arg\_1n>\n

…

<arg\_m1> <arg\_m2> … <arg\_mn>\n

Для простых и коротких ответов разрешается не использовать параметр numrecords (см. табл. 1)

Длина строки ограничена 16384 символами. При превышении данного лимита сообщения передаются в несколько приёмов аналогично передаче файлов.

Исключением из общего формата ответа являются команды LOGIN и PSW, передаваемые при установлении соединения, сервером клиенту, т.к. они не являются ответом на запрос. Порядок установления соединения СУПиК (после установки физического соединения по протоколу TCP, порт 9687, защищённое соединение SSL AES256):

C<-S – от клиента к серверу, S->C – наоборот.

[c<-s] LOGIN\n

[c->s] <login>\n

[c<-s] PSW\n

[c->s] <password>\n

[c<-s] GROUP <group>\n

Таблица 1. Команды клиента и ответы сервера СУПиК

|  |  |  |
| --- | --- | --- |
| № п/п | Описание | Команда |
| 1 | Завершение сеанса связи | QUIT |
| Подтверждение | BYE |
| 2 | Простой запрос SQL: SELECT … FROM … WHERE … ORDERBY … [ASC|DESC] | GVSBFS |
| Запрос: **GVSBFS n m db table headers\_0 headers\_1 ... headers\_n cond\_header\_0 value\_0 cond\_header\_1 value\_1 ... cond\_header\_m value\_m [order\_header] [ASC|DESC]**  **n** - количество запрашиваемых полей n;  **m** - количество пар сравнения m;  **db** – сокращённое имя БД  **table** - имя таблицы  **headers** - наименования полей запроса  **cond\_header, value** - пары сравнения, сначала поле сравнения, затем значение  **order\_header** – необязательное поле, по которому проводить сортировку  **ASC|DESC** - в каком порядке проводить сортировку  Пример: **GVSBFS 1 2 sup mainmenu idmainmenu mainmenu Выход idtable 35 idmainmenu ASC**  SELECT `idmainmenu` FROM `supik`.`mainmenu` WHERE `mainmenu`=”Выход” AND `idtable`=”35” ORDERBY idmainmenu ASC; | | |
| Ответ: **<number\_of\_records>**  Клиент при получении количества записей посылает: **RDY**  Сервер в ответ выдаёт: **value[0][0] value[1][0] ... value[n][0] value[0][1] value[1][1] ... value[n][1] ... value[0][k] value[1][k] ... value[n][k]**  Пример:  **<-GVSBFS 2\n**  **->RDY\n**  **<-2 5\n**  2 результата: «2» и «5». | | |
| 3 | Запрос состояния, посылается раз в минуту для контроля работы сервера | STATUS |
| 4 | Ответ на запрос состояния: <msgnum> <persact1> <persact2> … <persatcn>  **msgnum** – количество новых сообщений для данного пользователя  **persact** – учётные записи подключённых на данный момент пользователей | - |
| 5 | Запрос состояния сервера | STATS |
| 6 | Ответ на запрос состояния сервера: <uptime> <connnum> <cpuload> <freememory> <freehdd> |  |
| 7 | Отправка результатов экзаменов  Формат: EXAMRES <section> <type> <idpers> <result> <file>  Примечание: состав полей соответствует п. 1.6.4 файла «СУПиК РПр» | EXAMRES |
| 8 | Отправка файла  Формат: FILEPUT <путь> <имя\_файла> <длина\_в\_байтах>  Пример обмена:  ***-> FILEPUT tb/prot “Валуйских А.О. 2016-09-10 16.09.05.pdf” 99615***  ***<- OK***  ***-> …. (16384 байт)***  ***<- OK***  ***-> …. (<16384 байт)***  ***<- IDLE*** | FILEPUT |
| 9 | Создание компонента Altium  Формат: CRALT <class> <Library Ref> <Library Path> <Footprint Ref> <Footprint Path> <Sim Description> <Sim File> <Sim Model Name> <Sim Parameters> <Manufacturer> <PartNumber> <Package> <Marking> <NominalValue> <NominalVoltage> <Tolerance> <OpTemperaturen> <OpTemperaturem> <Pmax> <TC> <Comment> <HelpURL> <RevNotes> <Discontinued> <Description> <Notes> <Modify Date> <Creator> <prefix> <isSMD> <Nominal> <Unit> <par4> <par5>  Примечания:  - параметры Library Path, Footprint Path, Sim File и HelpURL передаются относительно соответствующих папок СУПиК, считающихся корневыми;  - все параметры, кроме class, являются соответствующими полями таблиц БД altium, поле class определяет класс и имя таблицы компонента (см. п. 1.1.1 файла «СУПиК РПр»).  - ответ сервера один из следующих:  ERROR – ошибка при разборе и/или добавлении компонента;  IDLE – всё в порядке, запись прошла хорошо;  ALTEX <Manufacturer> <PartNumber> – компонент уже есть в БД, надо выдать запрос подтверждения перезаписи;  NKEX <Manufacturer> <PartNumber> - компонент с таким именем и таким производителем есть в БД номенклатуры, надо выдать запрос подтверждения перезаписи.  В двух последних случаях при подтверждении перезаписи клиент выдаёт «REWRITE», при отказе – «CANCEL».  Пример обмена:  **-> CRALT capasitors “Capasitor” “capasitors.SchLib” …..**  **<- ALTEX НКП “Чип тант. Конд. 220 мкФ,6,3 В,10%,тип D”**  **-> REWRITE**  **<- IDLE**  На сервере в БД создаётся запись в БД altium, а также соответствующая запись в БД номенклатуры. | CRALT |
| 10 | Создание устройства  Формат: CRDEV <devices> <decimal> <idmanuf> <description> <notes> <first> <creator>  Примечания:  - состав полей соответствует п. 1.5.1 файла «СУПиК РПр». Оставшиеся поля – filehw, filesw, file3d, idsp создаются командой DEVFILEPUT (11)  - ответ сервера – ИД нового элемента в таблице devices или ERROR при ошибке  Пример обмена:  ***-> CRDEV “Устройство присоединения» «АВМР.426435.047-01» «ООО \»АВМ-Энерго\»» «Устройство присоединения УПО АВМР.426435.047-01» АВМР АВМР.426435.047 2001***  ***<- 114*** | CRDEV |
| 11 | Отправка файла к изделию  Формат: DEVFILEPUT <iddevices> <type> <file> <filesize>  Примечания:  - file – имя файла для передачи, filesize – размер файла, передача файла ведётся аналогично п. 8;  - type принимает одно из значений:  0 – архив с файлами САПР устройства (altium, pcad, …)  1 – архив с 3D-моделью устройства  2 – спецификация устройства в формате csv (файл, разделённый запятыми)  3 – документ на изделие  4 – архив с программным обеспечением устройства  - сервер, приняв файл типа 2, разбирает его и формирует электронную спецификацию изделия в таблице dev.spec, а также перечень элементов в таблице dev.lists (см. пп. 1.5.4 и пп. 1.5.2 файла «СУПиК РПр»);  - для типов 0, 1, 2, 4, если в БД уже имеются ссылки на файлы (т.е. уже такие файлы передавались для данного изделия), старые файлы перемещаются в каталог устаревших файлов (см. файл «Файлы сервер»), для типа 3 для перемещения файла в каталог устаревших необходимо дать команду FILEDEL.  Пример обмена:  -> DEVFILEPUT 114 2 “АВМР.426435.047-01 BOM.csv” 19475  ***<- OK***  ***-> …. (16384 байт)***  ***<- OK***  ***-> …. (3091 байт)***  ***<- IDLE*** | DEVFILEPUT |
| 12 | Создать пароль сотрудника  Формат: CRPSW <idpersonel> <password>  Примечания:  - idpersonel – ИД сотрудника по таблице ent.personel  - password – задаваемый пароль, если уже был установлен пароль, то он заменяется на переданный, пароль шифруется по заданному алгоритму и записывается в таблицу ent.perspsw | CRPSW |