Redis

Marvin

struct sdshdr struct sdshdr

```
redisContext *redisContextInit(void)函数初始化：
static redisContext *redisContextInit(void) {
    redisContext *c;

    c = calloc(1,sizeof(redisContext));
    if (c == NULL)
        return NULL;

    c->err = 0;
    c->errstr[0] = '\0';
    c->obuf = sdsempty();
    c->reader = redisReaderCreate();
    return c;
}

typedef struct redisContext {
    int err; /* Error flags, 0 when there is no error */
    char errstr[128]; /* String representation of error when applicable */
    int fd;         //文件描述符，socket
    int flags;
    char *obuf; /* Write buffer */
    redisReader *reader; /* Protocol reader */
} redisContext;


redisReader *redisReaderCreate(void) {       初始化
    redisReader *r;
    r = calloc(sizeof(redisReader),1);
    if (r == NULL)
        return NULL;
    r->err = 0;
    r->errstr[0] = '\0';
    r->fn = &defaultFunctions;
    r->buf = sdsempty();
    r->maxbuf = REDIS_READER_MAX_BUF;
    if (r->buf == NULL) {
        free(r);
        return NULL;
    }
    r->ridx = -1;
    return r;
}

typedef struct redisReader {
    int err; /* Error flags, 0 when there is no error */
    char errstr[128]; /* String representation of error when applicable */
    char *buf; /* Read buffer */
    size_t pos; /* Buffer cursor */
    size_t len; /* Buffer length */
    size_t maxbuf; /* Max length of unused buffer */

    redisReadTask rstack[9];
    int ridx; /* Index of current read task */
    void *reply; /* Temporary reply pointer */

    redisReplyObjectFunctions *fn;
    void *privdata;
}redisReader;


对应的是fn,在hiredis.c中有具体实现
static redisReplyObjectFunctions defaultFunctions = {
    createStringObject,
    createArrayObject,
    createIntegerObject,
    createNilObject,
    freeReplyObject
};

typedef struct redisReplyObjectFunctions {
    void *(*createString)(const redisReadTask*, char*, size_t);
    void *(*createArray)(const redisReadTask*, int);
    void *(*createInteger)(const redisReadTask*, long long);
    void *(*createNil)(const redisReadTask*);
    void (*freeObject)(void*);
} redisReplyObjectFunctions;

typedef struct redisReadTask {
    int type;
    int elements; /* number of elements in multibulk container */
    int idx; /* index in parent (array) object */
    void *obj; /* holds user-generated value for a read task */
    struct redisReadTask *parent; /* parent task */
    void *privdata; /* user-settable arbitrary field */
} redisReadTask;
```