

Os-SET 底软平台和 Open-SET 开放运行平台 设计和使用说明

星 级：

版本/修订：V1.2

编 制 _____

审 核 _____

2021-XX-XX 文件发布日期

2021-XX-XX 文件生效日期

文档修订历史

[illegible]

目 录

1	引言	1
1.1	编写目的	1
1.2	背景	1
2	OS-SET 通用平台介绍	1
3	OS-SET 底软平台	2
3.1	平台结构	2
3.2	编译	2
3.2.1	编译方法	2
3.2.2	动态库说明	3
3.3	调用方法	3
3.4	环境要求	4
3.5	安装包版本说明	4
3.5.1	安装和卸载	4
3.6	单元测试框架	5
4	OPEN-SET 开放平台	9
4.1	平台结构	9
4.2	部署方式	10
4.3	对接 SSET 平台	11
4.4	通用开发接口	11
4.5	ESL 远程调用库	12
4.6	OM APP 范例	12
4.6.1	在线模块查看	14
4.6.2	OM 模块加载	15
4.6.3	OM 模块卸载	15
4.6.4	OM 模块重载	15

1 引言

1.1 编写目的

本文档的目的是从 OSET 平台需求出发，设计 V1.1 版本的整体方案，可分解出进行概要设计和详细设计的模块和子模块，定义各模块和子模块的功能，模块和子模块的关系，满足 OSET V1.1 版本的系统需求和软件设计需求。

本文档用于指导 OSET V1.1 版本的概要设计和详细设计文档，以及后续的综合测试。

1.2 背景

软件系统名称：OSET 通用平台

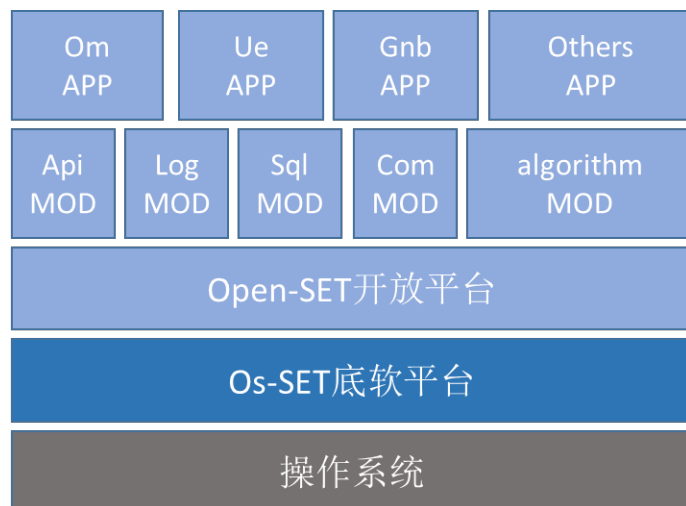
软件开发者：SSET 协议平台组

开发语言：C/C++、Python 等

硬件平台：X86 服务器

软件平台：Linux

2 OSET 通用平台介绍



为了实现各种不同体制协议的仿真和模拟，实现协议栈的模块化加载和迭代，减少操作系统对上层业务的影响，提高软件开发的统一性和可重用性，开发了 OSET 通用平台。

整个通用平台由 Os-SET 底软平台和 Open-SET 开放运行平台两部分构成。Os-SET 底层平台提供各种系统底层接口，实现底层系统和上层业务的隔离。Open-SET 平台是依托于 Os-

SET 底软平台开发的一款开放运行平台，整个平台全部采用模块化架构，支持模块的实时加载卸载，支持运行业务的实时弹缩，提供 esl 远程调用库，支持 python、java 的跨语言直接调用，尽量避免使用中间件。此外整个平台都围绕开放原则开发，定义了简单通用的函数接口和模板，方便第三方开发和迭代。

3 Os-SET 底软平台

3.1 平台结构

注册	卸载	重载	CLI交互	同步/异步		同步/异步		增/删/改/查	增/删/改/查	增/删/改/查	注册	配置				
				生产者	消费者	生产者	消费者										
模块管理				rocketMQ		Kafka		MYSQL接口	Redis接口	mongo	Nacos						
二次封装API																	
VOS兼容	内存管理	线程管理	OCKET管	文件管理	符串操作管	定时器管理	日志管理	信号量管理	uuid管理	errno管理	epoll	select	FIFO/Ring	Hash	链表	树结构
											I/O管理		数据结构管理				
C语言																	

Core 目录为核心目录，核心功能均在此目录开发，全部采用动态库提供给第三方。

平台目录结构如下



3.2 编译

3.2.1 编译方法

整个 OSET 工程可以独立安装，也可以作为外部工程被第三方工程 cmake 引用，视情况而定。独立安装时，编译使用 Cmake 外部编译，保持目录清洁。

1) cmake .. && make && make install

说明: cmake .. -DCMAKE_INSTALL_PREFIX=路径地址

//若不指定路径，编译的文件会默认放在/usr/local/install/oset，建议使用默认目录。

```
drwxr-xr-x 4 root root 36 Nov 24 00:18 core1
drwxr-xr-x 3 root root 21 Nov 24 00:18 core2
drwxr-xr-x 2 root root 4096 Nov 30 03:35 lib
[root@localhost oset]# pwd
/usr/local/install/oset
```

3.2.2 动态库说明

```
total 7368
-rwxr-xr-x 1 root root 118760 Jun 13 12:39 liboset-redis.so
-rwxr-xr-x 1 root root 519984 Jun 13 12:39 liboset-crypt.so
-rwxr-xr-x 1 root root 306048 Jun 13 12:39 liboset-esl.so
-rwxr-xr-x 1 root root 326488 Jun 13 12:39 liboset-hiredis.so
-rwxr-xr-x 1 root root 578568 Jun 13 12:39 liboset-ipfw.so
-rwxr-xr-x 1 root root 233448 Jun 13 12:39 liboset-mongo.so
-rwxr-xr-x 1 root root 311368 Jun 13 12:39 liboset-nacos.so
-rwxr-xr-x 1 root root 914328 Jun 13 12:39 liboset-nas5g.so
-rwxr-xr-x 1 root root 266736 Jun 13 12:39 liboset-rocketmq.so
-rwxr-xr-x 1 root root 3948792 Jun 13 12:39 liboset.so
```

Liboset.so *oset base* 动态库（日志、IO、线程、进程、内存管理、strings 等）

Liboset-xxx.so *oset module* 接口库

3.3 调用方法

默认不指定安装路径，安装在系统盘/usr/local/install/oset 目录下。

- cmake 工程：

- ✓ Base 库引用：

FindOSET.cmake

include_directories(\${OSET_INCLUDE_DIRS})

link_directories(\${OSET_LIBRARIES})

- ✓ module 库引用：

FindOSET-XXX.cmake

include_directories(\${OSET_XXX_INCLUDE_DIRS})

link_directories(\${OSET_XXX_LIBRARIES})

- 源码调用：

- 头文件引用+接口函数调用

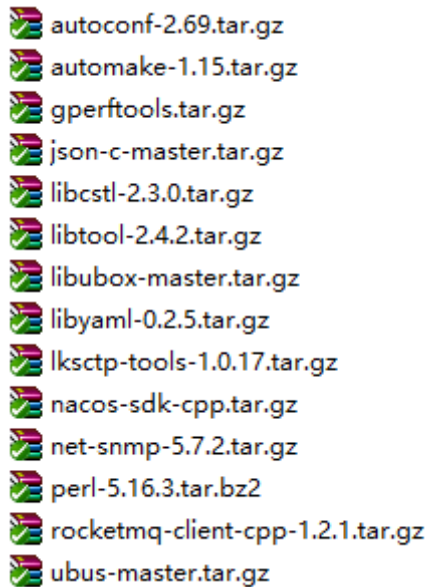
3.4 环境要求

1) 支持系统

Ubuntu20.04 64 位（推荐）

CentOS 7.4 64 位

2) 底软依赖库



- autoconf-2.69.tar.gz
- automake-1.15.tar.gz
- gperf tools.tar.gz
- json-c-master.tar.gz
- libcstl-2.3.0.tar.gz
- libtool-2.4.2.tar.gz
- libubox-master.tar.gz
- libyaml-0.2.5.tar.gz
- lksctp-tools-1.0.17.tar.gz
- nacos-sdk-cpp.tar.gz
- net-snmp-5.7.2.tar.gz
- perl-5.16.3.tar.bz2
- rocketmq-client-cpp-1.2.1.tar.gz
- ubus-master.tar.gz

执行 SSET/tools/ sset_os_oset_depends.sh 脚本，自动安装依赖。

3.5 安装包版本说明

● Ubuntu20.04


```
root@zky-PowerEdge-R740:/usr/local/install/oset/lib# dpkg -l|grep oset
ii  oset                    1.0.1-1
```

● Centos7.4

```
[root@localhost lib]# rpm -qa | grep oset
oset-1.0.1-1.x86_64
```

软件版本号 (major.minor.release) - 辅助版本号 (.api)

3.5.1 安装和卸载

 oset_1.0.1-1_amd64.deb	2022/1/12 21:15	DEB 文件	708 KB
 oset-1.0.1-1.x86_64.rpm	2022/1/12 21:15	RPM 文件	1,178 KB

安装新版本之前必须卸载旧版本!!!

3.5.1.1 Ubuntu20.04

查看安装包信息	<code>dpkg -s</code>	<code>oset</code>
安装	<code>dpkg -i</code>	<code>oset_ X.X.X-X _amd64.deb</code>
卸载	<code>dpkg --purge</code>	<code>oset</code>
查看安装情况	<code>dpkg -l grep</code>	<code>oset</code>

帮助命令使用 `dpkg --help`

3.5.1.2 Centos7.6

查看安装包信息	<code>rpm -qpi</code>	<code>oset- X.X.X-X.x86_64.rpm</code>
安装	<code>rpm -ivh --nodeps</code>	<code>oset- X.X.X-X.x86_64.rpm</code> （忽略依赖问题）
卸载	<code>rpm -evh</code>	<code>oset</code>
查看安装情况	<code>rpm -qa grep</code>	<code>oset</code>

帮助命令使用 `rpm --help`

3.6 单元测试框架

蓝色部分为底软初始化和相关配置；红色部分为 `abts` 框架。

```
int main(int argc, const char *const argv[])
{
    int rv, i, opt;
    oset_getopt_t options;
    struct {
        char *log_level;
        char *domain_mask;
    } optarg;
    const char *argv_out[argc+2]; /* '-e error' is always added */

    abts_suite *suite = NULL;
    oset_pkbuf_config_t config;

    rv = abts_main(argc, argv, argv_out); //UT 框架入参初始化
    if (rv != OSET_OK) return rv;

    memset(&optarg, 0, sizeof(optarg));
    oset_getopt_init(&options, (char**)argv_out);
```

```

while ((opt = oset_getopt(&options, "e:m:")) != -1) {
    switch (opt) {
        case 'e':
            optarg.log_level = options.optarg;
            break;
        case 'm':
            optarg.domain_mask = options.optarg;
            break;
        case '?':
        default:
            fprintf(stderr, "%s: should not be reached\n", OSET_FUNC);
            return OSET_ERROR;
    }
}

oset_core_initialize();
oset_pkbuf_default_init(&config);
oset_pkbuf_default_create(&config);
atexit(terminate);

rv = oset_log_config_domain(optarg.domain_mask, optarg.log_level);
if (rv != OSET_OK) return rv;

for (i = 0; alltests[i].func; i++) //加载所有测试用例函数
    suite = alltests[i].func(suite);

return abts_report(suite); //生成单元测试报告
}

const struct testlist {
    abts_suite *(*func)(abts_suite *suite);
} alltests[] = { //加载测试用例函数

    {test_list},
    {test_pool},
    {test_strings},
    {test_time},
    {test_conv},
    {test_log},
    {test_pkbuf},
    {test_memory},
    {test_rbtree},

```

```

    {test_timer},
    {test_thread},
    {test_socket},
    {test_queue},
    {test_poll},
    {test_tlv},
    {test_hash},
    {test_uuid},
    {NULL},
};

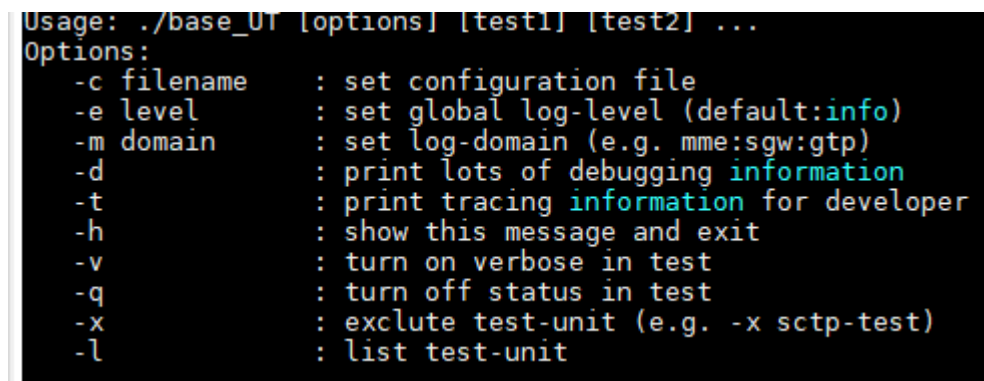
abts_suite *test_list(abts_suite *suite)// 加载测试用例 case
{
    suite = ADD_SUITE(suite)

    abts_run_test(suite, list_test1, NULL);
    abts_run_test(suite, list_test2, NULL);
    abts_run_test(suite, list_test3, NULL);
    abts_run_test(suite, list_test4, NULL);
    abts_run_test(suite, list_test5, NULL);
    abts_run_test(suite, list_test6, NULL);

    return suite;
}

```

1、使用说明：



```

Usage: ./base_01 [options] [test1] [test2] ...
Options:
  -c filename      : set configuration file
  -e level         : set global log-level (default:info)
  -m domain        : set log-domain (e.g. mme:sgw:gtp)
  -d               : print lots of debugging information
  -t               : print tracing information for developer
  -h               : show this message and exit
  -v               : turn on verbose in test
  -q               : turn off status in test
  -x               : exclute test-unit (e.g. -x sctp-test)
  -l               : list test-unit

```

- l 显示测试用例函数
- x 测试指定文件之外的其余用例
- v 打出错 case 详细信息（默认打开）
- q 关闭进度显示
- t 默认打开 trace 级别日志
- d 默认打开 debug 级别日志

-m log 模块 -e debug （设置部分模块打印日志级别，-m -e 配合使用）
整个单元测试框架默认使用 **error** 级别日志打印。

2、运行结果

```
[root@localhost base_UT]# ./base_UT
list-test      : SUCCESS
pool-test     : SUCCESS
strings-test   : SUCCESS
time-test     : SUCCESS
conv-test     : SUCCESS
log-test      : SUCCESS
pkbuf-test    : SUCCESS
memory-test   : SUCCESS
rbtree-test   : SUCCESS
timer-test    : SUCCESS
thread-test   : SUCCESS
socket-test   : SUCCESS
queue-test    : SUCCESS
poll-test     : SUCCESS
tlv-test      : SUCCESS
hash-test     : SUCCESS
uuid-test     : SUCCESS
UT Report      Total      Result
=====
UT Lists      90      All Pass
```

```
[root@localhost base_UT]# ./base_UT
list-test      : SUCCESS
pool-test     : SUCCESS
strings-test   : SUCCESS
time-test     : SUCCESS
conv-test     : -Line 120: Condition is false, but expected true
FAILED 1 of 7
log-test      : SUCCESS
pkbuf-test    : SUCCESS
memory-test   : SUCCESS
rbtree-test   : SUCCESS
timer-test    : SUCCESS
thread-test   : SUCCESS
socket-test   : SUCCESS
queue-test    : SUCCESS
poll-test     : SUCCESS
tlv-test      : SUCCESS
hash-test     : SUCCESS
uuid-test     : SUCCESS
Failed Tests   Total      Fail      Failed %
=====
conv-test     7          1          14.29%
```

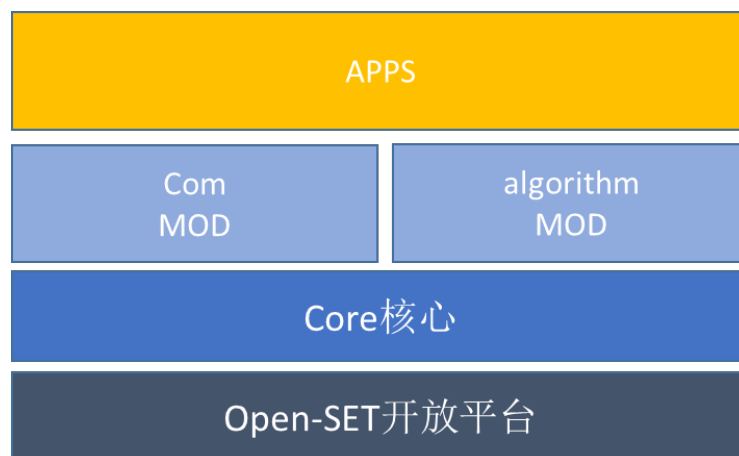
2、断言集说明

- 判断整形相等或不等
ABTS_INT_EQUAL / ABTS_INT_NEQUAL

-
- 判断 size 相等（size_t 类型）
ABTS_SIZE_EQUAL
 - 判断字符串相等或不等
ABTS_STR_EQUAL / ABTS_STR_NEQUAL
 - 判断指针相等或不为空
ABTS_PTR_EQUAL / ABTS_PTR_NOTNULL
 - 判断条件为真
ABTS_TRUE
 - 出错及断言
ABTS_FAIL / ABTS_ASSERT

4 Open-SET 开放平台

4.1 平台结构



Open-SET 平台由三大部分构成：

- **Core 核心：**core 核心负责整个平台的 running，是整个平台最重要的运行框架，负责 Mod 和 App 的管理和调度。
- **Mod：**Module 是平台的中间层，分为 common mod 和 algorithm mod 两大类，common 模块提供各种常用的 API 接口，供给上层业务调用，是 Os-SET 的实例化，如日志模块、CLI 模块、Redis 模块等；algorithm 模块是 SSET 常用的算法模块，提供各种算法 API 接口。
- **App：**APP 是平台最上层的业务，由业务逻辑和 Mod API 组合而成。

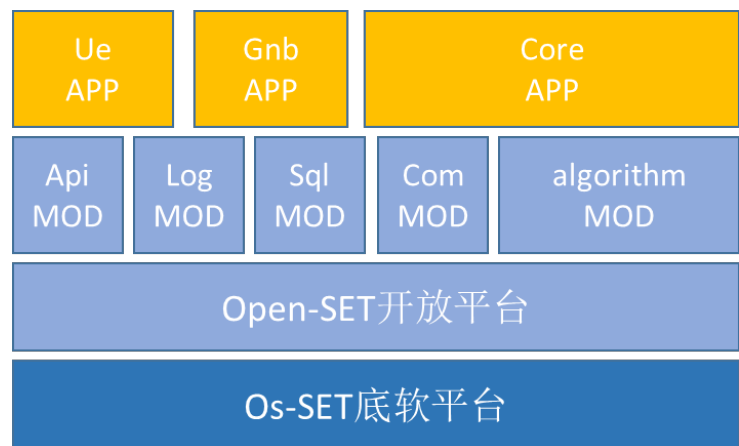
整个平台都是模块化设计，MOD 和 APP 两大核心可以任意加载、卸载、重载，非常灵活。

平台的运行依托于 Os-SET 底软平台，否则无法运行。

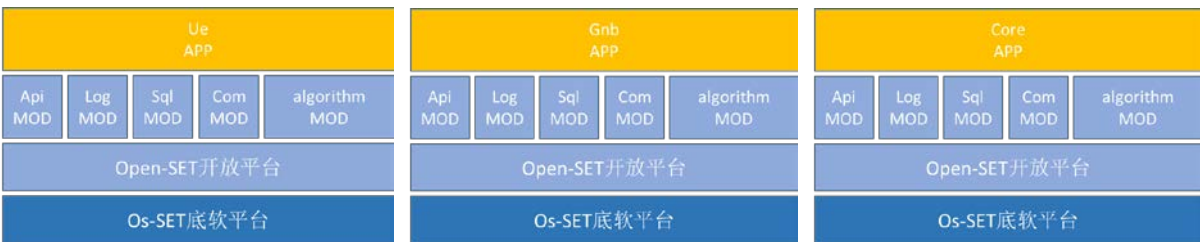
4.2 部署方式

部署的方式取决于业务，而不取决于平台。整个框架非常灵活，取决于应用场景。

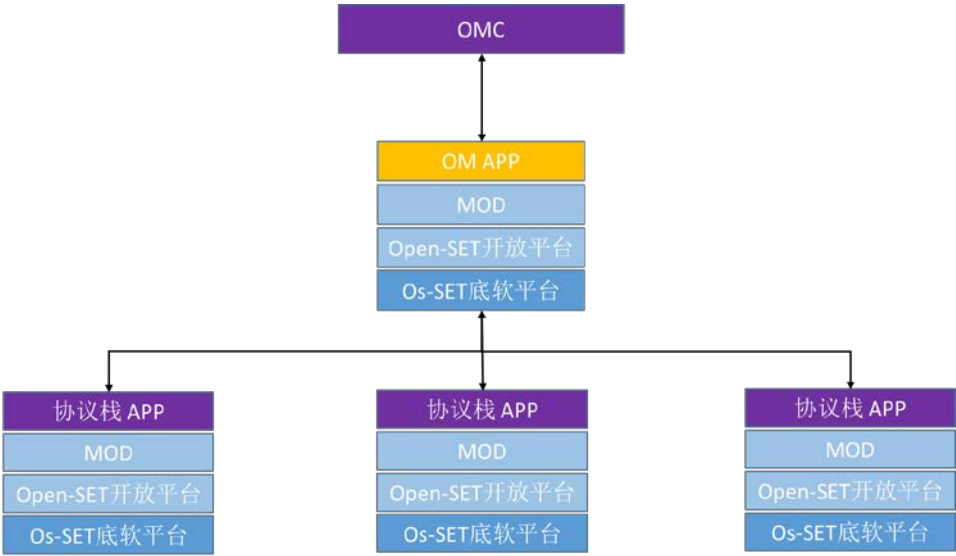
- 单机部署



- 分布式部署



4.3 对接 SSET 平台



SSET 平台作为分布式系统，协议栈业务的接入通过 OM 模块实现。

4.4 通用开发接口

通用开发接口是 MOD 和 APP 模块化的关键，也是第三方开发需要遵守的规范，整个接口非常简单和规范。

```
OSET_MODULE_LOAD_FUNCTION(mod_xxx_load)
{
    加载 xxx 模块
}

OSET_MODULE_SHUTDOWN_FUNCTION(mod_xxx_shutdown)
{
    卸载 xxx 模块
}

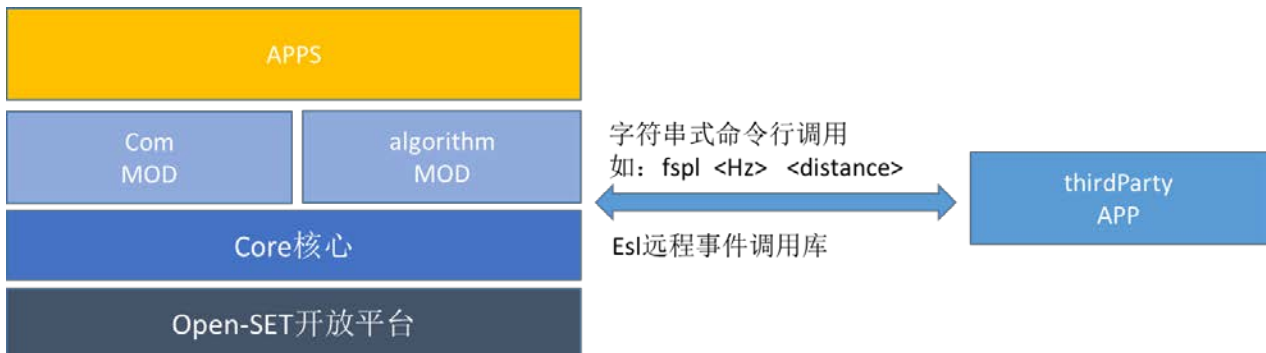
OSET_MODULE_RUNTIME_FUNCTION(mod_xxx_runtime)
{
    实时运行 xxx 模块
}

OSET_MODULE_DEFINITION(libmod_xxx,
```

```
mod_xxx_load,  
mod_xxx_shutdown,  
mod_xxx_runtime); //注册宏
```

接入 Open-SET 平台只需要以上四个函数，非常简单。

4.5 Esl 远程调用库



远程调用库是为了方便分布式场景下的远程调用，可以远程调用 Open-SET 平台上加载的任意模块 API。使用方法和本地方法一致，都是通过命令行方的字符串命令调用，至于底层实现方法，用户不需要关心。

4.6 OM APP 范例

整个协议栈的 OM 网管模块就是基于 Open-SET 平台开发,新模块的编码可以参考 OM。平台提供本地和远程两种 CLI 交互方式，通过 show help 命令可以查看使用方法。


```

2022-07-04 17:52:41.932173 100.100.1 [6E00] mod_event_socket.c:2003 socket up listening on 127.0.0.1:6021
OSET@node2> show help

...,Shutdown,libmod_api
acl,<ip>,Compare an ip to an acl list,libmod_api
alias,[add|stickyadd] <alias> <command> | del [<alias>|*],Alias,libmod_api
banner,,Return the system banner,libmod_api
bg_spawn,<command>,Execute a spawn command in the background,libmod_api
bg_system,<command>,Execute a system command in the background,libmod_api
bgapi,<command>[ <arg>],Execute an api command in a thread,libmod_api
complete,add <word>|del [<word>|*],Complete,libmod_api
console,loglevel [level]|colorize [on|toggle|off]|uuid [on|toggle|off]|json [on|toggle|off],Console,libmod_console
console_complete,<line>,,libmod_api
console_complete_xml,<line>,,libmod_api
db_cache,status,Manage db cache,libmod_api
echo,<data>,Echo,libmod_api
escape,<data>,Escape a string,libmod_api
event_sink,<web data>,event sink,libmod_event_socket
fspl,<f(Hz)> <distance(m)>,run FSPL modle,libmod_FSPL
getenv,<name>,getenv,libmod_api
gethost,,gethostbyname,libmod_api
help,,Show help for all the api commands,libmod_api
hiredis_api,,hiredis_api,libmod_hiredis
hostname,,Return the system hostname,libmod_api
json,JSON,JSON API,libmod_api
load,<mod name>,Load Module,libmod_api
log,<level> <message>,Log,libmod_api
memory,memory,Memory usage statistics,libmod_api
module_exists,<module>,Check if module exists,libmod_api
msleep,<milliseconds>,Sleep N milliseconds,libmod_api
mte,<distance(m)> <f(Hz)=20e9> <EIRP=34> <G_T=10> <encoding='16QAM'> <bandwidth=200e6>,run MTE modle,libmod_MTE
om,,Show om status,libapp_om
om_ring,,Show om ring queue status,libapp_om
osetname,,Return the oset name,libmod_api
pkbuf,,Show pkbuf status,libmod_api
reload,[ -f] <mod_name>,Reload module,libmod_api
reloadxml,,Reload XML,libmod_api
sched_api,[+@]<time> <group_name> <command_string>[&],Schedule an api command,libmod_api
sched_del,<task_id>|<group_id>,Delete a scheduled task,libmod_api
show,application|api|timer|aliases|complete|modules|tasks|limits|status,Show various reports,libmod_api
shutdown,,Shutdown,libmod_api
spawn,<command>,Execute a spawn command without capturing it's output,libmod_api
spawn_stream,<command>,Execute a spawn command and capture it's output,libmod_api
sql_escape,<string>,Escape a string to prevent sql injection,libmod_api
status,,Show current status,libmod_api
strftime_tz,<timezone name> [<epoch>|][<format string>],Display formatted time of timezone,libmod_api
system,<command>,Execute a system command,libmod_api
unload,[ -f] <mod_name>,Unload module,libmod_api
unsched_api,<task_id>,Unschedule an api command,libmod_api
uptime,[us|ms|s|m|h|d|microseconds|milliseconds|seconds|minutes|hours|days],Show uptime,libmod_api
uuid_loglevel,<uuid> <level>,Set loglevel on session,libmod_api
uuid_system,<thread id> <uuid>,Show uuid system status,libapp_om
version,,Version,libmod_api

```

4.6.1 在线模块查看

```
OSET@node2> show modules

type,name,iskey,filename
api,...,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,acl,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,alias,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,banner,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,bg_spawn,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,bg_system,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,bgapi,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,complete,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,console,libmod_console,/usr/local/install/sset/mod/libmod_console.so
api,console_complete,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,console_complete_xml,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,db_cache,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,echo,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,escape,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,event_sink,libmod_event_socket,/usr/local/install/sset/mod/libmod_event_socket.so
api,fspl,libmod_FSPL,/usr/local/install/sset/mod/libmod_FSPL.so
api,getenv,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,gethost,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,help,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,hiredis_api,libmod_hiredis,/usr/local/install/sset/mod/libmod_hiredis.so
api,hostname,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,json,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,load,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,log,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,memory,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,module_exists,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,msleep,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,mte,libmod_MTE,/usr/local/install/sset/mod/libmod_MTE.so
api,om,libapp_om,/usr/local/install/sset/mod/libapp_om.so
api,om_ring,libapp_om,/usr/local/install/sset/mod/libapp_om.so
api,osetname,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,pkbuf,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,reload,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,reloadxml,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,sched_api,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,sched_del,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,show,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,shutdown,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,spawn,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,spawn_stream,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,sql_escape,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,status,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,strftime_tz,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,system,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,unload,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,unsched_api,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,uptime,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,uuid_loglevel,libmod_api,/usr/local/install/sset/mod/libmod_api.so
api,uuid_system,libapp_om,/usr/local/install/sset/mod/libapp_om.so
api,version,libmod_api,/usr/local/install/sset/mod/libmod_api.so
application,bgssystem,libmod_app,/usr/local/install/sset/mod/libmod_app.so
application,hiredis_raw,libmod_hiredis,/usr/local/install/sset/mod/libmod_hiredis.so
application,log,libmod_app,/usr/local/install/sset/mod/libmod_app.so
application,socket,libmod_event_socket,/usr/local/install/sset/mod/libmod_event_socket.so
application,system,libmod_app,/usr/local/install/sset/mod/libmod_app.so
generic,libmod_logfile,libmod_logfile,/usr/local/install/sset/mod/libmod_logfile.so
generic,libmod_syslog,libmod_syslog,/usr/local/install/sset/mod/libmod_syslog.so
```

4.6.2 OM 模块加载

```
OSET@node2> load libapp_om
2022-07-04 17:56:47.792453 99.47% [INFO] oset-time2.c:1405 Timezone reloaded 0 definitions
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:204 pod thread que size[200], omc thread que size[100].
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:205 worker thread num [5], worker thread que size[200].
2022-07-04 17:56:47.792453 99.47% [INFO] asyncProducer.c:62 Producer start....
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:213 om create rtmq procuder success.IP[10.37.0.207],PORT[9876],GROUP_ID[Group_producer]
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-sockaddr.c:131 addr:10.37.6.28, port:20086
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-sockaddr.c:131 addr:10.37.6.28, port:20087
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:303 [Added] Number of om udp server is now 2
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-socket.c:82 socket create(2:2:17)
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-udp.c:21 udp_socket() family:2
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-socket.c:107 socket bind 10.37.6.28:20086
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-udp.c:40 udp_server() [10.37.6.28]:20086
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:278 om_udp_server() success [10.37.6.28]:20086
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-socket.c:82 socket create(2:2:17)
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-udp.c:21 udp_socket() family:2
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-socket.c:107 socket bind 10.37.6.28:20087
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-udp.c:40 udp_server() [10.37.6.28]:20087
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:278 om_udp_server() success [10.37.6.28]:20087
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:549 add_task [priority:3][push/top:1][owner:0x7f0e88875fb2] init!
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:447 task_new [priority:3][time:0][owner:0x7f0e88875fb2][task:0x7f0e8aad4b18] success!
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:510 om_worker_deal_thread[0x7f0e8a630e18][id = 0] running
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:549 add_task [priority:3][push/top:1][owner:0x7f0e88875fb2] init!
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:447 task_new [priority:3][time:0][owner:0x7f0e88875fb2][task:0x7f0e8aad4a98] success!
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:510 om_worker_deal_thread[0x7f0e8a630f18][id = 1] running
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:549 add_task [priority:3][push/top:1][owner:0x7f0e88875fb2] init!
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:447 task_new [priority:3][time:0][owner:0x7f0e88875fb2][task:0x7f0e8aad4998] success!
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:510 om_worker_deal_thread[0x7f0e8a630f18][id = 2] running
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:549 add_task [priority:3][push/top:1][owner:0x7f0e88875fb2] init!
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:447 task_new [priority:3][time:0][owner:0x7f0e88875fb2][task:0x7f0e8aad4818] success!
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:510 om_worker_deal_thread[0x7f0e8a630f18][id = 3] running
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:549 add_task [priority:3][push/top:1][owner:0x7f0e88875fb2] init!
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:447 task_new [priority:3][time:0][owner:0x7f0e88875fb2][task:0x7f0e8aad4898] success!
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:510 om_worker_deal_thread[0x7f0e8a631018][id = 4] running
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:549 add_task [priority:2][push/top:1][owner:0x7f0e88875fcd] init!
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:447 task_new [priority:2][time:0][owner:0x7f0e88875fcd][task:0x7f0e8aad4718] success!
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:732 om_pod_handle_thread[0x7f0e8a630418] running
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:549 add_task [priority:2][push/top:1][owner:0x7f0e88875fcd] init!
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:447 task_new [priority:2][time:0][owner:0x7f0e88875fcd][task:0x7f0e8aad4918] success!
2022-07-04 17:56:47.792453 99.47% [INFO] om-session.c:652 om_omc_handle_thread[0x7f0e8a631118] running
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:549 add_task [priority:1][push/top:1][owner:0x7f0e88875fcd] init!
2022-07-04 17:56:47.792453 99.47% [DEBUG] oset-threadpool.c:447 task_new [priority:1][time:0][owner:0x7f0e88875fcd][task:0x7f0e8aad4a18] success!
2022-07-04 17:56:47.792453 99.47% [CONSOLE] oset-loadable_module.c:680 Successfully Loaded [libapp_om]
+OK
+OK
2022-07-04 17:56:47.872375 99.47% [INFO] om-session.c:819 om_main_listen_thread[0x7f0e8a630918] running
2022-07-04 17:56:47.872375 99.47% [NOTICE] oset-loadable_module.c:184 Adding API Function 'om_ring'
2022-07-04 17:56:47.872375 99.47% [NOTICE] oset-loadable_module.c:184 Adding API Function 'om'
2022-07-04 17:56:47.872375 99.47% [NOTICE] oset-loadable_module.c:184 Adding API Function 'uuid_system'
```

4.6.3 OM 模块卸载

```
OSET@node2> unload libapp_om
2022-07-04 17:55:58.112434 98.30% [NOTICE] oset-loadable_module.c:397 Deleting API Function 'om_ring'
2022-07-04 17:55:58.112434 98.30% [DEBUG] oset-loadable_module.c:399 Write lock interface 'om_ring' to wait for existing references.
2022-07-04 17:55:58.112434 98.30% [NOTICE] oset-loadable_module.c:397 Deleting API Function 'om'
2022-07-04 17:55:58.112434 98.30% [DEBUG] oset-loadable_module.c:399 Write lock interface 'om' to wait for existing references.
2022-07-04 17:55:58.112434 98.30% [NOTICE] oset-loadable_module.c:397 Deleting API Function 'uuid_system'
2022-07-04 17:55:58.112434 98.30% [DEBUG] oset-loadable_module.c:399 Write lock interface 'uuid_system' to wait for existing references.
2022-07-04 17:55:58.112434 98.30% [CONSOLE] oset-loadable_module.c:1228 Stopping: libapp_om
2022-07-04 17:55:58.112434 98.30% [WARNING] om-session.c:424 om_task_termination waiting 3s!!!
2022-07-04 17:56:01.132384 98.17% [INFO] asyncProducer.c:70 Producer Shutdown!
2022-07-04 17:56:01.132384 98.17% [CONSOLE] oset-loadable_module.c:1248 libapp_om unloaded.
+OK
```

4.6.4 OM 模块重载

重载命令就是卸载+加载命令的合集，reload + 模块名字。