Main.c

主要函数：

ConfigSystem ：获取系统配置

InitSystem：系统初始化

InitPackSystem 控制结构初始化

Init\_MsgBuff 消息缓存区初始化

从数组来看，接受缓存有3维数组，[最大消息处理线程数][最大可缓存消息个数][最大可缓存消息长度]

消息分成：控制消息/告警消息

控制消息接收层 3维 消息发送 2维 告警消息 2维

Init\_LPCB\_Queue

MbufPoolInit内存池初始化

Init\_TimerQueue 定时器初始化

InitThreads线程初始化

控制消息发送线程启动（单个）

控制消息接收线程启动（多个）

TCPICP握手消息控制线程启动（单个）

UDPICP线程启动（单个）

告警消息多个接收控制线程启动（单个）

告警消息发送线程启动（单个）

InitMsipNfvSystem nfv MISP系统模块配置初始化

InitMpcpFuncSystem mpcp功能块初始化

MsipMfpsAllocResource mfps进程资源申请

MsipMfrpAllocResource mfrp进程资源申请

MsipMediaAllocResource media进程资源申请

MsipUdpAllocResource udp进程资源申请

这里涉及到资源集的链表队列操作（链表操作）

MsipCpuAffinityResourceInit （在多核系统上运行多线程程序的时候，实现绑核操作，LINUX的内核硬亲和性）

InitMfcpFuncSystem 初始化MFCP任务

InitMfrpFuncSystem 初始化MFRP任务

nfsClientInit 初始化NFS任务

InitRtpMediaSystem 初始化RTPMEDIA任务

InitMsipNfvBrdMan 初始化单板NFV管理任务

ListenPorts 开始监听端口包括telnet,tcpicp端口

有多个接受消息处理进程？一个发送进程的意思？都发往一个进程里处理吗？为什么发送的没有3维呢？

消息发送线程只有一个

告警消息处理线程只有一个

消息处理线程有多个，每个规定了最大处理能力的消息个数，以及每条消息的最大长度

一套系统

是这样子的

TCP 对应的是可靠性要求高的应用 从我这边来理解就是消息需要一来一回交互的

TCP更多是应用与面向连接的，端到端，可靠的数据包

所以我们系统控制消息的ICP应该是基于TCP的应用

UDP 对应的是可靠性要求不高的应用 从我这边来理解就是消息不需要一来一回交互的

多媒体数据包不需要的话，那就基于UDP

L2 Mac

L3 IP

L4 UDP ICMP ARP

那么此app消息处理就分为了

细致研究

控制消息接收发送线程是 具体哪些消息和那些应用连接的

TCPICP又是做什么的线程

UDPICP又是做什么的线程

线程具体内容；

MsgSender

发送的消息 进行一次拷贝 来自SystemData.SendMessageBuffer.MessageBuffer

根据模块号，进行消息发送

本模块发送直接写入接收SystemData.ReceiveMessageBuffer[i]. MessageBuffer

非本模块走的是SOCKET通信，

SOCKET通信：网络上的两个程序通过一个双向的通信连接实现数据的交换，这个连接的一端称为一个socket。 所以像我们运行在服务上跟其他APP连接的消息交互，走的是SOCKET通信

socket通信又分支持多种协议，常用的是TCP对应的是SOCK\_STREAM

UDP SOCK\_DGRAM

所以在MAIN读取配置的时候，就需要进行配置APP的通信格式

然而不管是何种消息都是来自于一个消息待发送数据区？

这里msg的消息数组使用static

1. 程序起来后这块数据就固定申请好在静态数据区，后续不用反复读取申请

2 静态数据区默认字节区都是0

握手消息，所有业务控制消息，即自定义的通信交互消息

MsgProcessor

多线程处理 消息来源为SystemData.ReceiveMessageBuffer[i]. MessageBuffer

为握手消息，所有业务控制消息

ListenIcpUdpSock

StartUdpServer

SOCKET通信配置启动

创建SOCKET，配置SOCKET套接口的缓存大小，绑定SOCKET

ConnectServer

建立SOCKET通信连接

RecvIcpUdpData

接收SOCKET –UDP报文

将报文写入缓存

SystemData.ReceiveMessageBuffer[iThreadId].MessageBuffer

ConnectTcpIcp

创建SOCKET，绑定SOCKET，连接SOCKET

AppIcpRecvMsg

接收消息 从消息池中接收 消息池为消息队列

最终写到

SystemData.ReceiveMessageBuffer[iThreadId].MessageBuffer

整套系统离不开

数据的接收发送处理

数据分成业务控制数据，媒体数据

业务控制数据分成可靠数据/不可靠数据

媒体数据在UDP协议上

分成

发送线程（单个）

接收处理线程（多个）

SOCKET通信（分成可靠通信，不可靠通信，分成可配置固定监听开启通信，业务过程开启的通信）