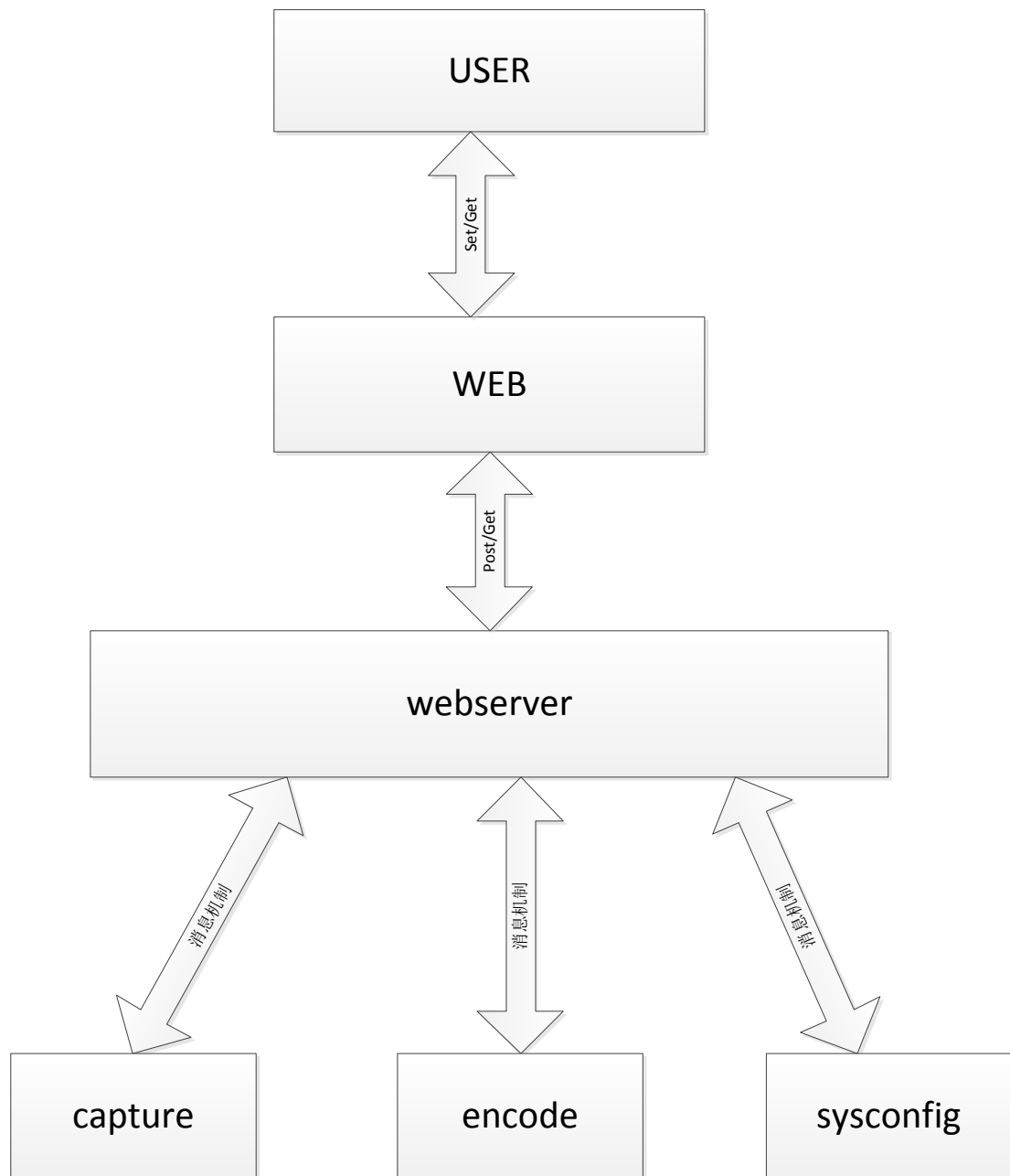


# Webserver 架构说明书

版本	日期	作者	说明
1.0	2016-7-13		1、webserver

## 1. 架构



图一 WEB 架构

## 2. 主要功能

模块主要功能：解析 XML 数据，把数据传输给相应模块

## 3. 限制条件

- 1、读取 webserver.ini 中的模块名、消息队列名（发送和接收），需要和具体模块一一对应
- 2、由于 80 端口可能被转码 webserver 进程占用，故此进程使用的端口是 81，访问可能需要输入 ip 地址：81 端口的形式（如：120.120.120.120:81）
- 3、不能超过系统允许的最大消息队列数量，可以用 `ipcs -l` 命令查看，不够需要更改系统支持的最大消息队列数量

```
[root@gospell-centos ~]# ipcs -l
----- 消息限制 -----
系统最大队列数量 = 31576
最大消息尺寸（字节）= 8192
默认的队列最大尺寸（字节）= 16384
```

图 2 消息队列限制

## 4. 输入

- 1、XML 文件

## 5. 输出

- 1、消息队列

## 6. 目录结构

Objs 目录：存放编译时产生的.o 等文件，是一个临时文件

.c/.h 文件：模块的源文件

Makefile 文件：编译所需文件

Build.sh 文件：一键编译文件，产生的可执行文件 `webserver_x64` 将被放到上级目录的 `build` 文件中

依赖库文件：`libmxml_x64.a`（XML 库文件）、`libmsg_x64.a`（消息机制库文件）、`libwebs_x64.a`（web 库文件）、`libuser_x64.a`（user 库文件）

## 7. 流程

- 一、代码流程图

```
//网页服务器
memset(&lWebInitParam,0, sizeof(WEB_InitParam));
lWebInitParam.m_pUserParam = s_pHandle;
lWebInitParam.m_WEBDataCB = Mult_WebDataCB;
lWebInitParam.m_PortNum = 80;
lWebInitParam.m_SocketTimeOut = 1000;
getcwd(dir, sizeof(dir));

strcpy(lWebInitParam.m_pDefaultHomePage, MULT_DEFAULT_HOME_PAGE);

sprintf(webdir, "%s/%s", dir, MULT_WEB_BASE_DIR);
strcpy(lWebInitParam.m_pDefaultWebDir, webdir);
sprintf(lWebInitParam.m_pDefaultUserFileName, "%s", MULT_WEB_USER_MANAGE_FILE_PATHNAME);
strcpy(lWebInitParam.m_pRealmName, "Gospel1");
lWebInitParam.m_NameLimit = 16;
lWebInitParam.m_PasswordLimit = 16;
lWebInitParam.m_UserLimit = 8;
```

图 3 代码流程图

## 二、流程框图

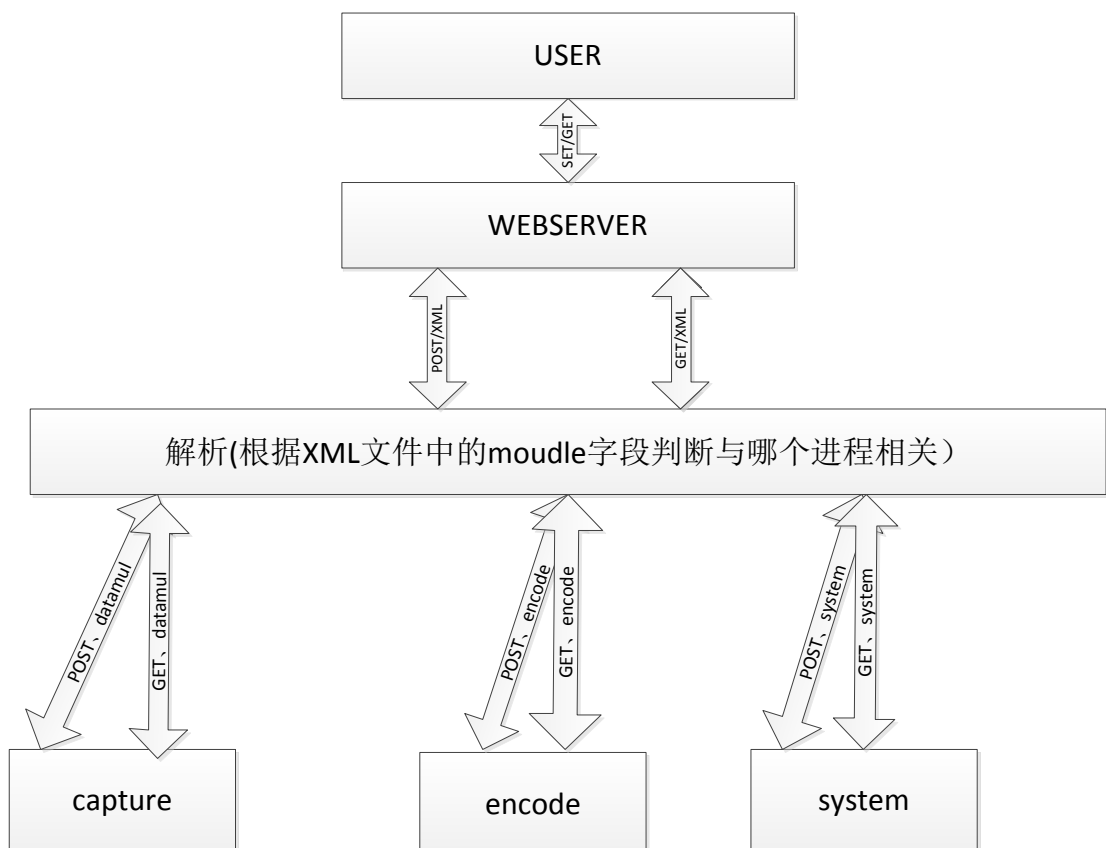


图 4 流程框图

## 三、代码详解（webserver 目录下 main.c，按执行顺序）

### 1、首先为进程的全局 handle 分配空间

```
pHandle = (MULT_Handle *)calloc(1,sizeof(MULT_Handle));
```

### 2、读取配置文件、初始化模块

```
cfg_init(pHandle);
```

- 3、分别创建 datamul、encode、system 三个进程的消息队列  
Msg\_Recv\_Create(pModule->m\_recv);  
Msg\_Send\_Create(pModule->m\_send);
- 4、分别创建 datamul、encode、system 三个进程的消息队列接收线程  
Mult\_Module\_Recv\_Main ();
- 5、回调函数赋值  
Mult\_WebDataCB ();
- 6、启动网页服务器  
WEB\_InitiateServer(&lWebInitParam);  
WEB\_StartServer();

## 8. 鸣谢

感谢之前对这个模块有过贡献的同事们。