ai目录

[工作必备软件：启动时执行。 2](#_Toc432167956)

[Lwork服务器启动需要执行的程序： 2](#_Toc432167957)

[Lwork业务 2](#_Toc432167958)

[Ltalk app业务 3](#_Toc432167959)

[工作总结 3](#_Toc432167960)

[关于Supervisor启动的问题， 5](#_Toc432167961)

[关于javascript的Jquery $的使用， 5](#_Toc432167962)

[egrep关于在linux中查找指定的字符串 5](#_Toc432167963)

[关于sse： 5](#_Toc432167964)

[关于yaws： 6](#_Toc432167965)

[关于用erlang进行json编解码： 7](#_Toc432167966)

[sdp说明 7](#_Toc432167967)

[Webrtc浏览器： 7](#_Toc432167968)

[网页界面布局分析： 8](#_Toc432167969)

[关于CSS z-index 属性的使用方法和层级树的概念 10](#_Toc432167970)

[关于vi的使用说明： 10](#_Toc432167971)

[关于erlang和python的集成，用erlport来进行通信：（windows） 10](#_Toc432167972)

[Erlport for Linux ubuntu 10](#_Toc432167973)

[Python 命令行下打印原始数据，可用 12](#_Toc432167974)

[linux下kill进程的方法 12](#_Toc432167975)

[Ubuntu启动erlang长节点的域名问题 12](#_Toc432167976)

[Python调用父类方法 13](#_Toc432167977)

[1 python webserver 13](#_Toc432167978)

[2 imaplib 获取邮件及解析 13](#_Toc432167979)

[特别是附件的处理,用到body和body[text] 13](#_Toc432167980)

[3 前端 13](#_Toc432167981)

[上传文件 13](#_Toc432167982)

[如何自动启动python文件 13](#_Toc432167983)

[Python编码错误问题 13](#_Toc432167984)

[邮箱开发总结： 13](#_Toc432167985)

[邮箱开发总结： 14](#_Toc432167986)

[命令行发起rest请求，可用curl 15](#_Toc432167987)

[关于screen 15](#_Toc432167988)

[Inbox 245号的邮件打不开。 15](#_Toc432167989)

[草稿箱 15](#_Toc432167990)

[Sublime的使用： 16](#_Toc432167991)

[安装erlang： 19](#_Toc432167992)

[安装yaws： 19](#_Toc432167993)

[查看某一端口的占用情况： lsof -i:端口号 27](#_Toc432167994)

[Create a new repository on the command line 28](#_Toc432167995)

[Push an existing repository from the command line 29](#_Toc432167996)

[ 30](#_Toc432167997)

[Linux下修改时间 30](#_Toc432167998)

[Ubuntu代理设置(版本1204) 32](#_Toc432167999)

[What file types work with PeerCDN? 33](#_Toc432168000)

[Who is behind PeerCDN? 33](#_Toc432168001)

[关于说吧 37](#_Toc432168002)

[Linux Shell远程执行命令（命令行与脚本方式） 42](#_Toc432168003)

[[Erlang 0033] 接入Erlang控制台的几种方法 46](#_Toc432168004)

[Linux dns设置 53](#_Toc432168005)

[Erlang call java 53](#_Toc432168006)

[快捷方式 53](#_Toc432168007)

[LTALK相关 53](#_Toc432168008)

## 工作必备软件：启动时执行。

1 putty

2 winscp

3 erlang应用程序环境

4 sourceinsight/sublime

5 erlang.chm

6 工作总结

Ninja

Clean: ninja –C out/Release –t clean WebRTCDemo

## Lwork服务器启动需要执行的程序：

## Lwork业务

启动脚本/home/ubuntu/lwork/batch即可

1 /home/ubuntu/lwork/www目录，

ulimit -n 409600

sudo yaws --name www@10.32.3.52 --conf yaws.conf --erlarg "-setcookie yxwabc +K true" --daemon

2 /home/ubuntu/lwork/server/user\_agent

erl -name lws@10.32.3.52 -pz ./ebin -setcookie yxwabc -s lw\_server -mnesia auto\_repair true –detached

3 /home/ubuntu/lwork/server/voice/mvoip

erl -name wrtc@10.32.3.52 -setcookie yxwabc -s voip -pa ./ebin –detached

4 /home/ubuntu/lwork/server/voice/code

yaws --conf yaws.conf --name www\_voice@10.32.3.52 --erlarg "-setcookie yxwabc" --daemon

5 /home/ubuntu/lwork/server/voice/code

erl -name voice@10.32.3.52 -setcookie yxwabc -pa ./ebin -pa ./yxa/ebin -s xengine –detached

6 /home/ubuntu/lwork/server/fate\_server

erl -name fate\_service@10.32.3.52 -setcookie yxwabc -pa ebin -s fate\_service –detached

7 /home/ubuntu/lwork/server/doc\_share

ulimit -n 409600

erl -name wftp@10.32.3.52 -setcookie yxwabc -pa ebin -s fid -detached

8 视频服务进程2个

/home/ubuntu/wconf

sudo yaws --conf yaws.conf --sname www\_conf --erlarg "-setcookie wrtc\_conference" --daemon

erl -sname manager -setcookie wrtc\_conference -pa ./ebin -s rooms -detached

9 用户管理进程

10 公司管理进程

11 工资管理进程 /home/ubuntu/salary\_login

yaws --name wwwsalary@10.32.3.52 --conf yaws.conf --erlarg "-setcookie yxwabc" –daemon

## Ltalk app业务

1 www节点 /home/ubuntu/wcg/www

ulimit -n 409600

yaws --name www\_dth@10.32.3.52 --conf yaws.conf --erlarg "-setcookie mlxyz +K true -s mnesia" --runmod wcg\_manager --daemon

2 gw节点 /home/ubuntu/wcg/gw\_git

## newlwork业务

1 /home/newlwk/www

Ulimit –n 65535

sudo yaws --sname 'newwww@localhost' --conf yaws.conf --runmod im\_router --erlarg "-setcookie hkxyz -boot start\_sasl -config elog" --daemon

2 agent /home/newlwk/server/user\_agent

erl -sname 'agent@localhost' -setcookie hkxyz -pa ebin -s agent

3 用户管理进程

/home/newlwk/management/www

sudo yaws --sname admin --conf yaws.conf --erlarg "-setcookie hkxyz" –daemon

4 文档管理进程 /home/newlwk/server/doc\_share

erl -sname wftp -setcookie hkxyz -pa ebin -s fsguard -detached

## 工作总结

2012-11-30:

内容：修改会议字段，和前端联调会议和计费。

预计：1-2天

时长：周3-周5

感觉：效率较低，原因：1 对于json编码不熟，细节纠缠；2 没有写测试用例，细节性问题解决时间比较长。

解决：测试用例编写，特别是对于arg的构造考虑一下

结果：to do

2012/12/3

Erlang函数的参数的原字符串，如何获取？类似c的##para

Voip的工作流程看看

2012/12/4

今天目标：

1 完成voip呼叫及话单功能

2 整理callback、meeting、voip及计费代码

3 完善测试平台功能。

Cookie：

2012/12/6

1上月总结

调整lwork的部分架构、部署策略和接口，可将lwork部署节点前移，将CT相关部分（ltalk、电话会议、voip、短信）几个部分聚合在一起，对外提供标准的ct服务接口，提高部署的灵活性，并提供ct服务的可扩展性。完成了代码的整合、测试、入库。

2 下月相关计划：

lwork新架构部署到香港在线运行，中兴Lwork需求满足

2012/1/5

1上月总结

lwork新架构部署到香港在线运行,新架构的好处是CT（ltalk、电话会议、voip、短信）业务作为一个独立的控制节点，对外提供标准的ct服务接口，提高部署的灵活性，并提供ct服务的可扩展性。

2 打通了企业间外部联系人功能，可以和外部联系人进行交流。此需求为满足中兴提出来，但不仅仅用于中兴。

3 企业社区开发完毕，便于公共交流或者信息发布。此需求为满足中兴提出来，但不仅仅用于中兴。

。

2 下月相关计划：

目前暂定Lwork外部定制化的需求满足以及lwork功能的一些完善工作。

2012/1

1 在运营支撑部门的协助下，通过调整会议语音编码方式和路由，提高了LWORK会议电话语音质量。

2 完成中国天翼短信单发业务功能，实现从smpp到中国天翼http转换的短信网关

3实现了和中国天翼的电话会议功能对接，使得回拨业务国内合法化。

4 短信信令跟踪功能，按照羊总要求，不过这个涉及到隐私问题，需要考虑。

5 邮箱收发功能接口实验完成，计划在lwork中集成邮箱服务，可以通过在lwork中设置，来进行邮件收发功能。

下月计划：

1 将短信网关部署到香港服务器。

2 实现中国天翼短信群发业务、短信回复业务

3 将中国天翼会议业务集成到lwork中。

4 邮箱业务集成到LWORK中。

备忘

2013/1/25

## 关于Supervisor启动的问题，

调用supervisor:start\_link({local, ?MODULE}, ?MODULE, []),的进程不能退出，否则，整个supervisor树都会退出。

## 关于javascript的Jquery $的使用，

只需要包含文件jquery-1.7.2.min.js。即在html的head标签中中包含: <script type="text/javascript" src="script/jquery-1.7.2.min.js"></script>

**关于弹出信息提示对话框**：

包含文件jquery\_plugins.js， 调用LWORK.msgbox.show('请输入跟踪号码', 5, 2000); 样式在common.css的zeng\_msgbox\_layer中，样式中包括图片images/gb\_tip\_layer.png

## egrep关于在linux中查找指定的字符串

cat q\_w2p.log | egrep '3187077825|2661303215'

**关于避免复杂的界面逻辑问题**

只需要将各个逻辑对应的界面各自封起来，统一show或者hide，修改forum中的界面逻辑问题，太复杂了。

**关于ajax请求：**

客户端需在html文件中包含restchannel.js文件，RestChannel.post (req.url, req.data, ok\_handler(data), err\_handler(err));



Erlang yaws端

## 关于sse：

客户端javascript代码：var es = new EventSource('/sse');

es.onmessage = function(event) {

var h2 = document.querySelector('#signal\_result');

h2.innerHTML += event.data;

h2.innerHTML += "<br/>";

}

Yaws侧代码： 在yaws.conf中增加 appmods = <"/sse", signal\_trace> ，在signal\_trace.erl中，用yaws\_sse来发送消息给客户端：如下代码：

out(A) ->

case (A#arg.req)#http\_request.method of

'GET' ->

case yaws\_api:get\_header(A#arg.headers, accept) of

undefined ->

{status, 406};

Accept ->

case string:str(Accept, "text/event-stream") of

0 ->

{status, 406};

\_ ->

{ok, Pid} = gen\_server:start(?MODULE, [A], []),

yaws\_sse:headers(Pid)

end

end;

\_ ->

[{status, 405},

{header, {"Allow", "GET"}}]

end.

handle\_info({traced\_msg, Msg, \_From}, State=#state{sock=Socket}) ->

%% nothing to do

io:format("trace:~p~n", [Msg]),

Data = yaws\_sse:data(Msg),

yaws\_sse:send\_events(Socket, Data),

{noreply, State};

## 关于yaws：

安装yaws，之后修改yaws.conf,如下：

include\_dir = "./app\_mod/src"

cache\_refresh\_secs = 0

id = lwork\_www

<server localhost>

port = 80

listen = 0.0.0.0

docroot = "./docroot"

appmods = <cgi-bin, yaws\_appmod\_cgi>

appmods = <lwork, lwork\_app>

appmods = <lmeeting, meeting>

partial\_post\_size=65536

deflate = true

</server>

在appmods对应的\*.erl中，增加out函数处理：例如



解析内容：对于url中的参数，是填充在arg记录的querydata字段中，对于post的参数是填充在clidata字段中。因此解析的方法是不同的。

如果是get方法，则clidata中是没有数据的，查询querydata字段，用yaws\_api:query\_var(Arg, Key),或者用utility:query\_xxxx(Arg, Key).

如果是post方法，查询querydata字段，方法和get是一样的。

如果clidata中是json格式，则用utility的decode方法解出，比如utility:decode(Arg, [{'SessionID',s},{'Key',s}])； 如果clidata中是<<"Status=0&SessionID=20”>>这样的http请求数据，则用yaws\_api:get\_var(Arg, Key)或者postvar(Arg,Key).

Out函数的返回值，为返回给客户端的response。

Daemon方式启动yaws，yaws –daemon

## 关于用erlang进行json编解码：

在out函数中用encode\_to\_json(JsonObj)返回。JsonObj用utility文件中的

utility:pl2jso或者a2jso等。

encode\_to\_json(JsonObj) ->

{content, "application/json", rfc4627:encode(JsonObj)}.

比如：

4> A=yaws\_utility:pl2jso([{a,1},{b,2}]).

{obj,[{a,1},{b,2}]}

5> www\_xengine:encode\_to\_json(A).

{content,"application/json","{\"a\":1,\"b\":2}"}

JsonObj的格式：

**一个实用的erlang序列化函数：**iolist\_to\_binary(Iolist).

## sdp说明



## Webrtc浏览器：

几个重要接口：

navigator.webkitGetUserMedia(mediaAttr, cb, fb)(获取媒体之后，回调cb)

webkitURL.CreateObjectURL

pc=new WebkitRTCPeerConnection

设置pc的OnIceCandidate(回调9次之后，createOffer)/OnAddStream/pc.addstream(localstream)/CreateOffer(pc\_OnGotLocalDescription)/pc.setRemoteDescription(new RTCSessionDescription({type:'answer',sdp:sdp}))

## 网页界面布局分析：

关于html文件编码的指定：

在html的head域中，增加一行：

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

关于lwork css的一些分析：

1 body： 了解clear:both的作用，配合前面float的使用。2 margin的使用(上、右、下、左) 3 list-style: none;将列表符号隐藏。4 font-size: 12px;

body{ margin: 0; padding:0; color:#404040(字体颜色); font-size:12px; background: white; }

body缺省大小：1407\*2615px， margin:0 之后为1423\*2615px

接下来是pagefix

.pagefix{position:fixed; width:100%; z-index:1002; top:0;}

z-index:1002;为垂直的叠放顺序。

position:fixed;的解释

|  |  |
| --- | --- |
| **值** | **描述** |
| static | 默认。位置设置为 static 的元素，它始终会处于页面流给予的位置（static 元素会忽略任何 top、bottom、left 或 right 声明）。 |
| relative | 位置被设置为 relative 的元素，可将其移至相对于其正常位置的地方，因此 “left:20″ 会将元素移至元素正常位置左边 20 个像素的位置。 |
| absolute | 位置设置为 absolute 的元素，可定位于相对于包含它的元素的指定坐标。此元素的位置可通过 “left”、”top”、”right” 以及”bottom” 属性来规定。 |
| fixed | 位置被设置为 fixed 的元素，可定位于相对于浏览器窗口的指定坐标。此元素的位置可通过 “left”、”top”、”right” 以及”bottom” 属性来规定。不论窗口滚动与否，元素都会留在那个位置。工作于 IE7（strict 模式）。 |

#container{ background:url(/images/body\_bg.png) repeat-y; 在垂直方向重复背景图片

clear:both;

margin:102px auto 0; 上面102px的间隔、两边自动（猜测应该是两边一样间隔）

position:relative; 相对于正常位置，调整其实际位置（根据margin）

min-width:1095px;} 宽度最少为：1095px

display:inline

|  |  |
| --- | --- |
| **值** | **描述** |
| none | 此元素不会被显示。 |
| block | 此元素将显示为块级元素，此元素前后会带有换行符。 |
| inline | 默认。此元素会被显示为内联元素，元素前后没有换行符。 |
| inline-block | 行内块元素。（CSS2.1 新增的值） |
| list-item | 此元素会作为列表显示。 |
| run-in | 此元素会根据上下文作为块级元素或内联元素显示。 |
| compact | CSS 中有值 compact，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。 |
| marker | CSS 中有值 marker，不过由于缺乏广泛支持，已经从 CSS2.1 中删除。 |
| table | 此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。 |
| inline-table | 此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。 |
| table-row-group | 此元素会作为一个或多个行的分组来显示（类似 <tbody>）。 |
| table-header-group | 此元素会作为一个或多个行的分组来显示（类似 <thead>）。 |
| table-footer-group | 此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。 |
| table-row | 此元素会作为一个表格行显示（类似 <tr>）。 |
| table-column-group | 此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。 |
| table-column | 此元素会作为一个单元格列显示（类似 <col>） |
| table-cell | 此元素会作为一个表格单元格显示（类似 <td> 和 <th>） |
| table-caption | 此元素会作为一个表格标题显示（类似 <caption>） |
| inherit | 规定应该从父元素继承 display 属性的值。 |

:after的用法

:after 伪元素在元素之后添加内容。

这个伪元素允许创作人员在元素内容的最后面插入生成内容。默认地，这个伪元素是行内元素，不过可以使用属性 display 改变这一点。

实例

此样式会在每个 h1 元素之后播放一段声音：

h1:after

{

content:url(beep.wav);

}

div： <div> 是一个块级元素。这意味着它的内容自动地开始一个新行。实际上，换行是 <div> 固有的唯一格式表现。可以通过 <div> 的 class 或 id 应用额外的样式。

Span：<span> 标签被用来组合文档中的行内元素。

# 关于CSS z-index 属性的使用方法和层级树的概念

<http://www.neoease.com/css-z-index-property-and-layering-tree/>

<input>框，缺省文本变灰可用插件jquery\_plugins.js，用法：

在html文件中包含

<script type="text/javascript" src="script/jquery\_plugins.js?t=75"></script>

SMTP:邮件传送

POP3/IMAP:邮件接收

## 关于vi的使用说明：

进入编辑模式i，退出编辑模式esc， 保存退出:x, 不保存退出:q, 保存:w



## 关于erlang和python的集成，用erlport来进行通信：（windows）

1 下载setuptools工具，（安装之前win7 64位下面需要修改注册表）；安装之后需要修改环境变量path，将C:\Python27\Scripts加入到path路径。

2 运行easy\_install erlport，安装erlport。（之后便可以from erlport import Port, Protocol, String）

3 例子

## Erlport for Linux ubuntu

问题：从python到erlang，不能传递长度大于249的字符串

1 先安装easy\_install,

sudo apt-get install python-setuptools

2 安装erlport

sudo easy\_install erlpor

关于gen\_fsm,

Smpp协议：（oserl开源）

Submit\_sm: esme=>mc

Pdu格式解码：

1 未分段（短信长度不超过153字节）：

{4,0,2, %4-submit\_sm 0: 2: ?

[{short\_message,"123"},

{sm\_default\_msg\_id,0},

{data\_coding,3},

{replace\_if\_present\_flag,0},

{registered\_delivery,0},

{validity\_period,[]},

{schedule\_delivery\_time,[]},

{priority\_flag,0},

{protocol\_id,0},

{esm\_class,0}, %0: is not segmented

{destination\_addr,"222"},

{dest\_addr\_npi,1},

{dest\_addr\_ton,1},

{source\_addr,"111"},

{source\_addr\_npi,1},

{source\_addr\_ton,5},

{service\_type,[]}]}

2 分段（短信长度超过153字节）：

{4,0,12, %\_CmdId, \_Status, \_SeqNum

[{short\_message,[5,0,3,1,3,3,49,49,49,49,49]},

-define(UDHL\_CONCAT, 5). % UDH Length excluded length for Concatenated SMs

-define(UDHL\_PORT\_16, 6). % UDHL for 16 bit Application Port Addressing

-define(UDHL\_PORT\_8, 4). % UDHL for 8 bit Application Port Addressing

-define(IEI\_CONCAT, 0). % Information Element Identifier for Concatenated SMs

-define(IEI\_PORT\_16, 5). % IEI for 16 bit Application Port Addressing

-define(IEI\_PORT\_8, 4). % IEI for 8 bit Application Port Addressing

-define(IEDL\_CONCAT, 3). % IE Data length for Concatenated SMs excluded length

-define(IEDL\_PORT\_16, 4). % IEDL for 16 bit Application Port Addressing

-define(IEDL\_PORT\_8, 2). % IEDL for 8 bit Application Port Addressing

%5-udh len(fix) 0: IEI\_CONCAT 3,1,3,3---信息单元，可能不止一条(其中3:length before payload 1:ref\_num 3: total\_seqment 3:cur\_segment\_num) 49,49,49,49,49---payload

{sm\_default\_msg\_id,0},

{data\_coding,3},

{replace\_if\_present\_flag,0},

{registered\_delivery,0},

{validity\_period,[]},

{schedule\_delivery\_time,[]},

{priority\_flag,0},

{protocol\_id,0},

{esm\_class,64},

{destination\_addr,"222"},

{dest\_addr\_npi,1},

{dest\_addr\_ton,1},

{source\_addr,"111"},

{source\_addr\_npi,1},

{source\_addr\_ton,5},

{service\_type,[]}]}

Erlang版本查看：

erlang:system\_info(otp\_release)

Erlang获取本机ip地址：

get\_internal\_ip()->

{ok, H} = inet:gethostname(),

{ok, #hostent{h\_addr\_list=Ips}} = inet:gethostbyname(H),

lists:nth(1, [I|| I={10,\_,\_,\_}<-Ips]).

%Ips.

## Python 命令行下打印原始数据，可用

print "%r" %( param)

## linux下kill进程的方法

1 在shell下top

2 用>和<，找到目的进程的pno，q退出。

3 sudo kill 9 pno即可。

或者用sudo top，然后在里面直接用k pno即可。

## Ubuntu启动erlang长节点的域名问题

1 –name之后的名字必须能够在linux下ping通，否则erlang下也ping不通。长名称域名必须为xxx.xxx的样式。

## Python调用父类方法

class a：

def f(self): print ‘a’

class b(a):

def f(self): a.f(self)

## 1 python webserver

用simpleHttpServer,不能并发处理http请求.

## 2 imaplib 获取邮件及解析

## 特别是附件的处理,用到body和body[text]

## 3 前端

## 上传文件

## 如何自动启动python文件

## Python编码错误问题

非utf-8编码的字符被当成了utf-8处理，应该先str.decode(codetype).encode(‘utf-8’)

## 邮箱开发总结：

邮箱开发近一月时间，时间是比较长，应该说到今天才取得相对比较满意的效果，其中的工作内容领导可能不一定清楚，现给二位领导做个简单总结如下，按时间顺序：

1 Imap/pop3/smtp协议研究

2 确定开发框架，初步选择YAWS，研究发现erlang对imap、pop3、smtp的支持比较弱，找了一个开源的erlang imap开发项目，不太靠谱。

3 邓辉指示，python对邮箱支持不错，建议用YAWS+PYTHON webserver+imap代理的架构。

4 研究python的webserver。并搭起了一个简单的python webserver。

5 研究imap、pop3、smtp库，感觉是比较强大，但是缺乏文档支持。在参考google到的一些代码后，初步发送和接收邮件实验成功。

6 将webserver和邮件代理程序集成，搭建好环境。在此过程中，发现webserver经常会由于错误崩溃，决定对webserver进行修改，增强其稳定性。

7 研究附件的发送和接收，其中涉及到附件的上传，因为是python程序，不像erlang有现成的代码，花了一段精力完成文档上传功能，完成了附件发送功能。

8 发现一个问题，对于大附件的邮件，每次查看，时间非常长，这会导致邮箱可用性非常差，再花大力气，解决大附件邮件的查看问题。这个耗费精力很长，google上没有可以参考的案例，主要是需要分段下载解析。Python文档不完整，因此花了一个周末的时间在实验。

9以上是后端的开发过程，前端的开发也在穿插进行，其中发邮件用的是百度的ueditor插件，发现经常出现异常。就是经常会填不了内容，经过第二个周末反复实验，最后解决。

10 将前端、python webserver、邮箱代理和LWORK整合，通过反复试验测试，最后找到一个方案，在不修改lwork代码的情况下，将邮箱集成进来。

11 又发现一个问题，自己发送的附件，在收信的时候打不开，经过一番折腾研究试验后，发现时python提供的接口函数的可选参数存在猫腻，这个在网上居然google不到同样的问题。晕

12 经过反复测试和修改，确定了webserver和邮箱代理的稳定性。

13 大的功能已就绪，剩下就是界面及友好性的修改了，这个也是细节非常多，本周末加班，终于取得了我认为比较满意的效果。

**综上所述，整个邮箱的开发过程，涉及前端、python webserver、imap/smtp/pop等协议的开发研究，以及和lwork的集成工作，开发过程非常紧凑，目前连续3个周末加班，感觉也非常充实，对于python、webserver、邮箱、前端开发又有了很大的进步。对以后工作会有非常大的帮助。**

## 邮箱开发总结：

1 邮件MIME的各种编码。

2 中文unicode、utf-8编码

3 python webserver

4

显示本地媒体流，并capture

**HTML**

<video id="live" width="320" height="240" autoplay></video>

<canvas width="320" id="canvas" height="240" style="display:none;"></canvas>

**JS ( jQuery )**

var video = $("#live").get()[0];

var canvas = $("#canvas");

var ctx = canvas.get()[0].getContext('2d');

navigator.webkitGetUserMedia("video",

function(stream) {

video.src = webkitURL.createObjectURL(stream);

}

)

setInterval(

function () {

ctx.drawImage(video, 0, 0, 320,240);

var data = canvas[0].toDataURL("image/jpeg");

},1000);

## 命令行发起rest请求，可用curl

export TB\_url=https://api.opentok.com/hl/session/create

headerstr="X-TB-PARTNER-AUTH: API\_KEY:API\_SECRET"

curl -X POST -H "$headerstr" $TB\_url

## 关于screen

ubuntu中，在screen中启动的进程，如果screen不退出的话，则该进程将杀不掉。

Screen 杀掉进程

screen -X -S [session # you want to kill] kill

screen 的buffer显示

Ctrl + A, ESC

and then vim-like commands:

Ctrl + u or Ctrl + d

## Inbox 245号的邮件打不开。

## 草稿箱

Eclipse快捷键大全

**使用快捷键可以大大提高我们的工作效率，下面是我在学习Android开发中，Eclipse里常用到的快捷键，希望对新手有所帮助。  
新建项目：Ctrl+N 这个基本上所有的软件都通用  
快速保存：Ctrl+S 这个也通用  
关闭文件：Ctrl+W 同样通用  
注释与取消注释为：Ctrl+/ 多行注释先选中多行，单行注释只要把光标定位在该行即可  
组织导入：Ctrl+Shift+O 快速导入需要的类，清除不需要的类（就是自动帮我们填写程序最顶上的import）  
删除整行：Ctrl+D 删除光标所在行  
定位到最后编辑的行：Ctrl+Q 这个真的很好用，尤其是对我这种记性差，常要翻上翻下看变量名，类名的名  
定位到指定行：Ctrl+L Logcat会提示哪行出错，就用这个定位  
文件重命名：Alt+Shift+R Eclipse默认是不能慢点两下左键来重命名的，需要点右键菜单  
代码对齐:Ctrl+Shift +F,整个文件代码对齐，同样适用于Aptana编辑器，即支持html,Javascript等  
自动缩进：Ctrl+i,对选中的代码自动缩进，与代码对齐不同，这个不会自动在分号后换行，不适用Aptana  
选中字母转大写：Ctrl+Shift+X  
选中字母转小写:Ctrl+Shift+Y  
System.out.println() 输入sysout按alt+/  
暂时就这么多，不定期更新**

## Sublime的使用：

[**http://www.cnblogs.com/ruochenchen/archive/2013/01/03/Matrix.html**](http://www.cnblogs.com/ruochenchen/archive/2013/01/03/Matrix.html)

jsep文档： javascript for webrtc E:\work\webrtc\draft-ietf-rtcweb-jsep-00.txt

跨域方法：

1 用Xmlhttprequest的简单请求、预检方法，可以跨域

<http://blog.csdn.net/net_lover/article/details/5172509>

2 用jsonp的callback的方式，包括jquery的ajax方法。

<http://www.cnblogs.com/chopper/archive/2012/03/24/2403945.html>

验证一下。在服务器侧加上头**Access-Control-Allow-Origin:**\*，

验证结果：没问题。

Google chrome frame：

Google Chrome Frame是一个神奇的东东

只需要在网站中写上,<meta http-equiv="X-UA-Compatible" content="chrome=1">,然后安装了 Google Chrome Frame插件的用户就可以使用IE外壳,使用Chome内核来浏览网页.如果用户希望ie默认启动Chrome Frame浏览网页,可以按照[百度百科](http://baike.baidu.com/view/2831140.htm)上面的说明来做.

jsep文档： javascript for webrtc E:\work\webrtc\draft-ietf-rtcweb-jsep-00.txt

1. opentok流程

offline 1 确定apikey和获取密钥

online：

1 client请求视频通话，用户信息

2 server创建或者获取session，并根据apikey和secrets产生token，通知client。

创建session需要和中心服务器交互，产生token不需要交互。

3 client根据session、apikey和token连接到中心服务器，并注册客户回调函数。过程如下：

首先通过REST API，获取sessioninfo，包括如下：

case "session\_id":

this.sessionId = g.textContent;

break;

case "partner\_id":

this.partnerId = g.textContent;

break;

case "session\_status":

this.sessionStatus = g.textContent;

break;

case "p2p\_server\_url":

this.p2pServer = g.textContent;

break;

case "media\_server\_hostname":

this.mediaServer = g.textContent;

break;

case "media\_server\_url":

this.mediaServerURL = g.textContent;

break;

case "messaging\_server\_url":

this.messagingServer = g.textContent;

break;

case "ice\_servers":

this.iceServers = [];

var e = g.childNodes, m, f;

for ( var h = 0, d = e.length; h < d; ++h) {

if (e[h].localName === "ice\_server") {

f = e[h].attributes;

m = {

url : f.getNamedItem("url").nodeValue

};

if (f.getNamedItem("credential")

&& f.getNamedItem("credential").nodeValue.length) {

m.credential = f.getNamedItem("credential").nodeValue

}

this.iceServers.push(m)

}

}

break;

case "properties":

之后创建websocket，通过它和各server进行交互。

执行客户回调函数。

二、视频

1 client点击视频

2 getusermedia函数调用，回调函数处理如下：

记录localstream、 initiatePeerConnection

3 initiatePeerConnection中

curObj.pc = new webkitRTCPeerConnection(servers);

curObj.pc.onicecandidate = curObj.pc\_onICE();

curObj.pc.onaddstream = curObj.pc\_onRemoteStreamAdded();

curObj.pc.addStream(curObj.localStream);

curObj.pc.createOffer(curObj.pc\_onGotLocalDescription(), null, sdpConstraints);

this.pc\_onICE = function(){

curObj.pc.createOffer(curObj.pc\_onGotLocalDescription(), null,

};

this.pc\_onRemoteStreamAdded = function(){

var curObj = this;

return function(e){

var remoteStreamURI = webkitURL.createObjectURL(e.stream);

curObj.owner.onRemoteStream(remoteStreamURI);

};

};

Lwork前端client收到通知处理：比如视频

Common.js：load\_interval ：loadContent.vedio\_Notice('video', data['video']);

在webvideo.js onVideoNotice中：设置$('.dynamic').find('.video') .find('.p2p\_vedio\_notice')

Common.js：绑定lwork.template中的dom元素 .vedio\_notice 即loadContent.bind('vedio\_notice', loadContent.vedio\_noticehandle, 'class');

Call webvideo.js的handleVideoInvitation

Linux下查看及修改文件属性

<http://summerafternoon.diandian.com/post/2012-05-17/19607191>

webrtc getusermedia 远程桌面共享

webrtc流程图：

<http://www.w3.org/TR/webrtc/#call-flow-browser-to-browser>

<http://www.w3.org/TR/webrtc/#peer-to-peer-data-api>

<http://www.iwebrtc.com/blog/webrtc1-0/>

<http://www.iwebrtc.com/blog/webrtc-1-0-real-time-communication-between-browsers-11/>

<http://www.iwebrtc.com/blog/web-real-time-communication-webrtc-media-transport-and-use-of-rtp-02/>

<http://tools.ietf.org/html/draft-ietf-rtcweb-rtp-usage-05#section-4>

<http://www.html5rocks.com/en/tutorials/webrtc/basics/?redirect_from_locale=zh#toc-rtcpeerconnection>

单独编译各模块 <http://www.cppblog.com/tx7do/archive/2013/09/01/202908.html>

webrtc datachannel用于P2P

<https://github.com/peer5/sharefest>

[http://sharefest.peer5.com](http://sharefest.peer5.com/)

conversant

## 安装erlang：

先安装gcc：

sudo apt-get update

sudo apt-get install build-essential

sudo apt-get install build-essential libncurses5-dev openssl libssl-dev

<http://docs.basho.com/riak/1.2.1/tutorials/installation/Installing-Erlang/#Installing-on-GNU-Linux>

1 最简单的方法用kerl安装，可以激活不同版本的erlang。

2 linux手工安装erlang，在./configure的时候出现curses错误，“ No curses library functions found”，此时，ubuntu下需要运行：sudo apt-get install build-essential libncurses5-dev openssl libssl-dev

tar zxvf otp\_src\_R16B03-1.tar.gz

./configure --enable-m64-build --disable-hipe --enable-halfword-emulator --with-ssl

下次试试./configure --enable-m64-build --enable-halfword-emulator --with-ssl --enable-kernel-poll --without-javac

make && sudo make install

钱沛：./configure --enable-halfword-emulator --enable-kernel-poll --without-javac CFLAGS="-g" &&sudo make install

-g 会导致之前的那些iso出错，后来又去掉了。

安装erlang：

1 首先安装openssl，

2 其次安装

## 安装yaws：

tar xzvf yaws-1.98.tar.gz

sudo apt-get install libpam0g-dev

sudo ./configure

sudo make

sudo make install

关于开源软件的原则：

1 看文档，对于windows支持是否充分，不充分，直接到linux虚拟机下去build，不要在windows下折腾。2013/5/11 在windows上整socket.io整了一天，浪费时间。而在linux虚拟机下就很快整好。

Long poll方式实现了一个模拟的socket.io，好用。

域名注册

52 lwork服务器的端口号分配：10.32.3.52/202.122.107.66

8089： www\_voice

8086: show

8088: mail server

192.168.180.29/116.228.53.181 上海服务器

运行了青牛demo. 80端口 sname名: www 目录: www

Feizhidu:192.168.180.29/

1 前端: voip.js/api.js/lw\_lang.js/ 修改api.js中的SERVER\_ADDR\_FZD地址为：116.228.53.181

2 WWW 80端口 和青牛一起 yaws.conf 80端口 [修改snode.hrl中的SNODE为lws@localhost](mailto:修改snode.hrl中的SNODE为lws@localhost) caspar-pc

3 LW\_UA lws@localhost 修改lw\_config.erl中的 get\_ct\_server\_ip() 为 "http://116.228.53.181:8089/"

4 www\_voice 8089端口 用www\_voice.bat/www\_voice 修改sname为 www\_voice，修改call.hrl中的 SNODE为 voice@localhost，修改VOIPNODE为wrtc@localhost

5 voice voice@localhost 用go.bat/voice 修改sname为voice

6 wrtc [wrtc@localhost，修改avscfg.erl](mailto:wrtc@localhost，修改avscfg.erl)中的get（host\_ip）为本机，

老的修改，hosts.hrl中的 HOST为本机地址，SIPNODE为voice@localhost

Ubuntu上抓包

sudo tcpdump -i eth1 -s 0 -w voip2.pcap

erl\_opus.c 编译通过，试下erl\_opus.erl

2013/9/23

Linux之间传输文件方法: -r选项则包含目录

scp ./server\_error.log [lwk@lwork.hk:/home/ubuntu/lwork/wcg/service/www/docroot/](mailto:lwk@lwork.hk:/home/ubuntu/lwork/wcg/service/www/docroot/)

指定端口号： scp -P 8888 test.tar.bz2 test@test.com:~/

ls –l查看文件时间 大小

2014/1/14 工作备忘

1 目前的几个工作project，

* Fzd voip电话：

文件路径: D:\work\lwork\wcg\service\www voice D:\work\lwork\server\voice\gw\src

部署网址: wcg.shuobar.cn/testwcg.shuobar.cn/testwcg2.shuobar.cn(http://203.195.188.128)/14.17.102.2(铁通)

节点www/wcg/sip

Js打包： pack:do4voip() 生成voip.min.js

* Fzd webcall坐席:

文件路径: js: D:\work\lwork\wcg\service\www\p2pdemo.js erl: D:\work\lwork\wcg\service\www\app\_mod 主要是utility和tcp\_relayer文件夹，room、swap等文件。

部署网址: [www.94360.com](http://www.94360.com) 文件夹：docroot/test/lw2fzd.min.js

节点: www

Js打包: pack:do4webcall() 生成webcall.min.js

* Fzd webconf:

文件路径: D:\work\lwork\server\wras 和wmg

部署网址: https://testwcg2.shuobar.cn/webcall.html(http://203.195.188.128)

相关节点www/wras/wmg,其中www:接入webcall.yaws,token认证. Wras：webconf呼叫控制和chat等；wmg语音处理。

这个地方比较搞

www：webcall.yaws=>rpc:call(wras)=>rpc:call(www)(如果需要token认证)

=>rpc:call(wmg)

所以部署时，webcall.yaws要部署到www目录下，其他js和erl文件部署到wras和wmg。

Js打包： wras下的pack:do1(),生成webcall0.min.js

* New lwork:

文件路径D:\work\lwork\lwork\_new

部署网址: [http://ltalk.hk:8088](http://ltalk.hk:8088/) <https://ltalk.hk:8089/>

部署路径: /home/ayu/newlwork

节点: newwww

* 视频demo:

文件路径:前端: D:\work\lwork\lwork\_new\www\docroot\demo erl端:同New lwork

部署网址：<http://ltalk.hk:8088/demo/demo.html>

节点:同newlwork

部署网址：<https://www.94360.com/demo/demo.html>

节点:同www

视频copy and broadcast

Rtcp终结控制：

* 收包处理：
  + check\_srtcp\_digest(需要用到A\_k，存放在r\_srtcp#cryp.a\_k中，来自sdp中的crypto) ,校验rtcp包的有效性。
  + 解密：aes\_ctr\_enc解密rtcp包body（需要用到salt,key, 存放在r\_srtcp#cryp.e\_s和e\_k中，来自sdp中的crypto），获取rtcp原始包。
  + rtcp:show\_rtcp解码rtcp包
  + update\_sr\_timecode更新r\_base和vr\_base#base\_info结构中的ts\_m32和rcv\_timecode，update\_fb\_info(更新rtcp包中的rtcp\_sr.receptions或者rtcp\_rr.receptions数据到state中的in\_audio)，update\_avg\_rtt(更新rtt计数)，notify\_lost\_seqs（用于重新发送rtcp包中rtcp\_pl.NACK 指示的丢失包），notify\_video\_pli（用于根据rtcp包中的rtcp\_pl.pli更新meeting\_room中的pli指示，以便meeting\_room据此判断是否暂缓发送视频流数据，关键包不受此指示影响）

更新state中相关rtcp的数据。

send\_sr/send\_pli/send\_rmb



关于音视频编解码:

* Ilbc/isac/cng/g711:

文件路径：:\work\lwork\server\voice\erlang\_source\_so\webrtc\modules\audio\_coding\codecs

编译方式，进入各模块的build目录，比如：D:\work\lwork\server\voice\erlang\_source\_so\webrtc\modules\audio\_coding\codecs\ilbc\build

执行make compile\_sp；make compile; make ilbc;make nif即生成so文件。

* Opus模块：文件路径：D:\work\lwork\server\voice\erlang\_source\_so\opus-1.0.2

编译方式： make -f my\_make libopus; make -f my\_make decoder; make -f my\_make encoder; make -f my\_make erl\_opus; make -f my\_make nif;

Erl接口c文件路径：

生成so的命令，包含c++库，gcc -fPIC -shared -o erl\_opus.so erl\_opus.c -lopus -lstdc++ -I/otp/erlang/lib/erlang/usr/include -I/home/dvlp/yyy/opus/include -I/home/dvlp/yyy/opus/src

PIC: position independent code

即编译erl\_opus.c 链接libopus.a和标准c++库,生成erl\_opus.so动态库。

Opus编译优化

编译选项：-fvisibility=hidden -minline-stringops-dynamically -maccumulate-outgoing-args -minline-all-stringops 大约提高5%，又好像没影响？

链接选项：-Wl,-O3,-zcomberloc

* G729:

文件夹: D:\work\lwork\server\voice\erlang\_source\_so\mod\_729

媒体编解码so文件：D:\work\lwork\server\voice\erlang\_source\_so

模块：Amix、mod\_729/amr/opus/resample/webm/webrtc

需要intel的ipp库？

* VP8模块：

C文件路径D:\work\lwork\server\voice\erlang\_source\_so\libvpx-v1.1.0，

So编译方式: 进入build目录，运行：（先编译vpx build下面 make）

make -f makevp8 erl\_vp8

make -f makevp8 nif

安全宝cdn功能

Opus rfc6716

Opus: LP(SILK)/MDCT(CELT)

LP is for NB, MB, or WB audio and frame sizes from 10 ms to 60 ms, and requires an

additional 5 ms look-ahead for noise shaping estimation. Like Vorbis [VORBIS-WEBSITE] and many other modern codecs, SILK is inherently designed for variable bitrate (VBR)

coding, though the encoder can also produce constant bitrate (CBR)

streams. The version of SILK used in Opus is substantially modified

from, and not compatible with, the stand-alone SILK codec previously

The MDCT layer is based on the Constrained-Energy Lapped Transform

(CELT) codec [CELT]. It supports NB, WB, SWB, or FB audio and frame

sizes from 2.5 ms to 20 ms, and requires an additional 2.5 ms look-

ahead due to the overlapping MDCT windows. The CELT codec is

inherently designed for CBR coding, but unlike many CBR codecs, it is

not limited to a set of predetermined rates. It internally allocates

bits to exactly fill any given target budget, and an encoder can

produce a VBR stream by varying the target on a per-frame basis. The

MDCT layer is not used for speech when the audio bandwidth is WB or

less, as it is not useful there. On the other hand, non-speech

signals are not always adequately coded using linear prediction.

Therefore, the MDCT layer should be used for music signals. deployed by Skype.

2.1.1. Bitrate

Opus supports all bitrates from 6 kbit/s to 510 kbit/s.

8-12 kbit/s for NB speech,

o 16-20 kbit/s for WB speech,

o 28-40 kbit/s for FB speech,

o 48-64 kbit/s for FB mono music, and

o 64-128 kbit/s for FB stereo music.

2.1.2. Number of Channels (Mono/Stereo)

2.1.3. Audio Bandwidth

2.1.4. Frame Duration

Opus can encode frames of 2.5, 5, 10, 20, 40, or 60 ms.

It can also

combine multiple frames into packets of up to 120 ms

2.1.8. Constant/Variable Bitrate

2.1.9. Discontinuous Transmission (DTX)

CUDA

2014/1/27 选择

Cocoonjs： <http://stackoverflow.com/questions/18728990/does-webrtc-work-with-phonegap-cordova>

收尾工作：

1 www应用服务器，修改snode.hrl中的SNODE宏定义。

2 用户管理服务器

Lw\_config配置文件配置voice、video、http\_proxy，如下

{voice\_server,"127.0.0.1"}.

{video\_server,"127.0.0.1"}.

{http\_proxy,"10.32.3.66", 808}.

定时获取cdr出错

3 后台用户管理服务器，修改snode.hrl中的SNODE宏定义，怎么创建管理用户、公司、用户？通过8083端口下的orgmanagement来创建

创好了公司之后，自动admin/888888登录即可。用户可以在管理界面上创建。

批量导入用户信息，要安装xlrd的python包。

4 语音服务器voice

A 拷贝voice/code目录

B 修改yaws.conf，端口号

C 修改siphost:myip() 为本机固定ip=10.32.2.4

D 修改session:ssip() 为ss ip

E 修改voice文件，节点名为voice1@ip

F 修改www\_voice，节点名改为www\_voice@ip

5 媒体服务器 -最新（15.2.15）

A 修改application目录下的go

6 视频会议服务器 D:\work\lwork\_newest\server\voice\tconf\_ver（svn上的，只是电话会议服务器，不行，少文件），部署在52服务器：/home/ubuntu/wconf

a 修改conf\_api.erl的WRTC宏定义；

b 修改yaws.conf,增加conf的映射conf\_api

c

以上都不对，代码在D:\work\lwork\server\voice\wbak\_from\_dvlp，来自dvlp机器的wbak目录下，部署目录在wconf，运行makecopy、makeconfig、goo即可。

LibvpxV1.1.0的编译

a 在build目录下： sudo ../configure --extra-cflags=-fvisibility=protected --enable-pic --enable-shared --disable-debug --disable-install-docs --enable-shared --enable-realtime-only --enable-error-concealment；configure选项会显示在vpx\_config.c文件中。

b sudo make 生成libvpx.a(972KB)

c make –f makevp8 erl\_vp8 生成erl\_vp8\_temporal.o

d make –f makevp8 nif 生成erl\_vp8\_temporal.so

7文件服务器,server/doc\_share [wftp@ubuntu.livecom](mailto:wftp@ubuntu.livecom)

启动: erl -pa ebin -s fid -setcookie yxwabc -name wftp

Fid:start()

注意，如果cookie不一致，会有上传文件静止不动或者我正在处理的现象。

8 计费服务器: server/fate\_server

9 公司管理服务器: D:\work\lwork\_newest\orgManagement\www 8083端口

http://Lwork.hk:8083/superAdmin.html

10 邮箱服务器： 启动：import websvr即可

11 radisys会议服务器硬件

Voip电话、电话会议打通

Dtls部分

Phonegap/cordava 调用native接口:

Ios: <http://www.adobe.com/devnet/html5/articles/extending-phonegap-with-native-plugins-for-ios.html>

Android:

<http://www.adobe.com/devnet/html5/articles/extending-phonegap-with-native-plugins-for-android.html>

cocoonjs+webview:

性能高，<http://support.ludei.com/hc/en-us/articles/200767118-Development-Guide>

2014.2.9

cocoonjs vs crosswalk

crosswalk only support android 4.0+,not ios; but it has

cocoonjs support android and ios.

Httptunnel

<http://blog.csdn.net/zhangxinrun/article/details/5942260>

但对于浏览器方式而言，如果所有端口被封，只能通过80、443端口

<http://tools.ietf.org/pdf/draft-hutton-rtcweb-nat-firewall-considerations-01.pdf>

研究下tcp的webcall

Erlang进程占用端口的释放办法：

### ****查看某一端口的占用情况： lsof -i:端口号****

再用sudo kill 9 pno即可

ANDROID开发

层次架构：

应用层、框架层、核心类库、运行时、硬件抽象层、linux类核

应用层通过Android SDK调用框架层，通过NDK绕过框架层调用特定类库，NDK是由C++接口组成。其中应用层基于java开发，也可通过JNI+C/C++开发。

另：通常，应用层通常用java实现，C/C++只能用于编写功能类库。也有特殊，Android 2.3API9开始,新增了android.app.NativeActivity类，开发者可以基于NDK，通过C/C++语言实现具体功能，即，开发者可以通过C/C++能实现整个应用。但控件在android中没有native的实现，所以普通的应用开发需要通过java来实现上层界面。

框架层：

由多个系统服务组成，组件管理服务、窗口管理服务、地理信息服务、电源管理服务、通话管理服务。这些服务都寄宿在系统核心进程中，以线程方式存在。服务间通过IPC进行通信。

开发而言，SDK，一系列的java功能模块。SDK的设计决定了上层应用的开发模式、效率及功能范畴。关注SDK的变迁。 Android应用是基于组件的设计方式，由若干个组件组成，组件之间不直接通信，通过框架层的系统服务，集中地调度和传递消息。

运行时：

java程序在运行阶段的二次编译，开发不会涉及到。Dalvik、.dex文件。

核心类库：

框架层的贤内助，不能以线程方式存在，需要被系统服务加载到进程空间里，通过类库的JNI接口调用。包括原生类库和第三方类库。其中第三方类库，是android提供丰富功能的保障。如：android的多媒体处理，依赖于开源opencore的支持；浏览器控件的核心支持，webkit移植而来。Android会为第三方类库封装一层jni接口，供框架层调用。

为特定应用的高效应用，OpenGL等内库以NDK形式提供给开发者，如此兼容性比较好，如果越过NDK的话，后续会有兼容性问题。

Linux内核:解决硬件移植性问题

硬件抽象层，为框架层提供接口支持。

功能架构：

界面框架：resource和layout，通过控件库和接口设计，搭建自己需要的界面。控件树，android。View。View,android.view.ViewGroup,observer模式，layout控件。应用资源体系使开发者逃离界面开发的泥沼。Xml资源文件盒数据资源文件。

数据存储： 本端存储和云存储，本端存储私有化数据。共享数据方法，数据源组件或者扩展存储设备。

github

### Create a new repository on the command line

touch README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/yuxw75/test.git

git push -u origin master

### Push an existing repository from the command line

git remote add origin https://github.com/yuxw75/test.git

git push -u origin master

**subversion转git以及git仓库合并**

公司决定转向github，所以今天遇到的问题是要把两个subversion仓库合并到一个github仓库中。

subversion转git这个问题很容易，[github推荐](https://help.github.com/articles/importing-from-subversion)使用[svn2git](https://github.com/nirvdrum/svn2git)这个脚本。官方网站上写出了详细的安装方法，如下：

$ sudo apt-get install git-core git-svn ruby rubygems

$ sudo gem install svn2git

然后建立两个子目录，存放转换后的两个subversion仓库：

$ mkdir tmp && cd tmp

$ mkdir projectA projectB

开始转换：

[charlee@ubuntu:~/tmp]$ cd projectA

[charlee@ubuntu:~/tmp/projectA]$ svn2git http://svn.example.com/projectA # 由于是标准目录结构，这里不需要指定trunk分支

[charlee@ubuntu:~/tmp/projectA]$ cd ../projectB

[charlee@ubuntu:~/tmp/projectB]$ svn2git http://svn.example.com/projectB

[charlee@ubuntu:~/tmp]$ cd ..

这样就得到了projectA和projectB两个git仓库。接下来用[stackoverflow上提到](http://stackoverflow.com/questions/1425892/how-do-you-merge-two-git-repositories)的[git-stitch-repo](https://metacpan.org/module/git-stitch-repo)将这两个仓库合并成一个。首先安装git-stitch-repo：

$ tar xzvf Git-FastExport-0.07.tar.gz

$ make

$ sudo make install

[charlee@ubuntu:~/tmp]$ git clone https://github.com/charlee/project.git

[charlee@ubuntu:~/tmp]$ cd project

[charlee@ubuntu:~/tmp/project]$ git-stitch-repo ../projectA:projectA ../proejctB:projectB | git fast-import

[charlee@ubuntu:~/tmp/project]$ git checkout master-A

[charlee@ubuntu:~/tmp/project]$ git checkout master-B

[charlee@ubuntu:~/tmp/project]$ git checkout master

[charlee@ubuntu:~/tmp/project]$ git branch -d master-A master-B

然后扔到github上：

$ git push -u origin master

大功告成。

Git的分支管理

<http://www.open-open.com/lib/view/open1328069889514.html>

卸载git ui，安装tortoisegit

Github

如何修改linux时区和时间？

# 

# [Linux下修改时间](http://www.cnblogs.com/xiao-yu/archive/2011/06/30/2094747.html)

<http://rainbird.blog.51cto.com/211214/226327/>

date -s "20091112 18:30:50"

hwclock --systohc

Lsof命令参考 

LWORK手机版如何修改代码，android和ios

Erlang tuple操作

 一直传说erlang没有[数组](http://erlang.wikidot.com/cn:array)。其实，tuple(元组) 就是数组（当然是常数组）。与数组相关的函数如下：

size(Tuple)  
%% 取数组大小

erlang:append\_element(Tuple1, Term) -> Tuple2  
%% 添加一个元素

element(N, Tuple) -> Term  
%% 取数组的第N个元素

erlang:make\_tuple(N, InitialValue) -> Tuple  
%% 创建一个数组元素值均为InitialValue的数组（该数组包含N个元素）。

setelement(Index, Tuple1, Value) -> Tuple2  
%% 修改下标为Index的数组元素的值。

list\_to\_tuple(List) -> Tuple  
%% 链表转数组

tuple\_to\_list(Tuple) -> List  
%% 数组转链表

可以看出，Tuple数组相关的功能还是比较全的。

Mnesia数据库升级

-record(third\_login\_map, {third\_party,acc,name,self\_intro,pwd=[],activated=false, test\_remove\_last\_field=[]}).

假如最后一个field被删除了，升级数据库方法如下：

transform\_test\_no()-> %% for mnesia database updating, very good

Transformer = fun(Item)->

%setelement(1,Item,third\_login\_map)

L = tuple\_to\_list(Item),

L1=lists:sublist(L, length(L)-1),

list\_to\_tuple(L1)

end,

{atomic,ok}=mnesia:transform\_table(third\_login\_map,Transformer, record\_info(fields, third\_login\_map) ).

# [Ubuntu代理设置(版本1204)](http://blog.csdn.net/makefish/article/details/8472492)

分类： [Linux](http://blog.csdn.net/makefish/article/category/613176)2013-01-06 12:48 218人阅读 [评论](http://blog.csdn.net/makefish/article/details/8472492#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/makefish/article/details/8472492#report)

Ubuntu代理设置(版本1204)

1，上网代理（如浏览器等）：  
从界面进去：点击左侧设置图标，选择hardware/Network/Network proxy: 选择Manual。然后输入代理服务器地址。  
  
2，apt代理：  
gedit ~/.bashrc  
http\_proxy=www.yourproxy.com:port  
export http\_proxy  
  
3，svn代理：  
~/.subversion/servers  
http-proxy-exceptions = \*.yourproxy.com  
http-proxy-host = www.yourproxy.com  
http-proxy-port = port  
  
4，wget代理：  
wget  http\_proxy=www.yourproxy.com:port

Erlang编译so或者dll的nif, <http://www.erlang.org/doc/tutorial/nif.html#id64988>

Turbostat

Svn repos库 备份:<http://www.iitshare.com/linux-svn-migration.html>

Newlwork 到时候用配置10.32.7.28 掩码 255.255.255.0 网关：10.32.7.254吧

Newlwk/L\*\*\*\*\*

中兴ltalk相关

1 根据工号查用户数据

(company@10.32.3.38)43> db:get\_employee({1,"10025188"}).

{employer,{1,"10025188"},

[230,157,142,231,187,141,230,156,139],

[1,4,12,1],

<<"888888">>,[],[],30.0,undefined}

部分是[1,4,12,1]

ayu ltalk/aixunda2014

3.8记录

1 kp\_www工程，auth\_handler.erl实现了register、registered\_login、unregistered\_login的测试。需要增加邮箱激活的测试。

今天实现了erlang自动编译、自动测试。E:/work/kp/Com.erl,com:start().

源程序包含-include\_lib("eunit-2.2.3/include/eunit.hrl"). 所有以\_test结尾的函数会自动运行。

3.9工作备忘:

2 周一工作备忘，谭志红提的问题，让红点闪烁。还有浏览器刷新之后，如何保留红点提醒cookie？浏览器任务栏闪烁。

Chrome live reloader： 自动刷新工具

**On Top Replica 区域置顶工具**

**Window on top工具也非常实用，整个窗口的置顶工具，已经安装**

**Jsunit**

<http://llying.iteye.com/blog/258605>

**Erlang错误日志**

<http://blog.csdn.net/gavinr/article/details/9956407>

beiwang:

1 lwork database

2 newlwork继续完善

3 kp

4 newlwork的新设计

5 我要上课 wysk webrtc的p2p实验

6 newlwork datachannel传输超大文件

7 <https://peercdn.com/> 用datachannel

What file types work with PeerCDN?

Any static file should work with PeerCDN. We're focusing primarily on large assets like video, audio and images.

## Who is behind PeerCDN?

PeerCDN is built by [**Feross Aboukhadijeh**](http://feross.org/), [**Abi Raja**](https://github.com/abi), and [**John Hiesey**](https://github.com/jhiesey). We're huge fans of JavaScript, WebRTC, and open web technologies.

3.12记录

1 完成了bind\_test、unbinded\_third\_login\_test、unactivated\_binded\_third\_login\_test、activated\_binded\_third\_login\_test。

下一步进行简历管理，基本上不改变前端传来的结构，直接存入mnesia。是不是要另外简历一张简历的数据表。目前看不需要，直接放入third\_login\_map的一个成员中.还是单独的一张resume表吧

田丽华：

聊天界面关闭按钮不能设置到右上角吗

对啊，那个大家都用习惯了，对很多表情都有印象

直接从资源管理器，拖文件或者图片到聊天窗口

讨论组是否能能大家知道谁是发起人呢

Video全屏的时候自动切换到对方

全屏显示界面修改。

谭志红：

1。能让PC加入会议，能让PBX小号码加入Lwork回拨会议。

2。能在移动终端上直接采用终端呼叫全球电话，类似于PC－全球语音

3。Lwork能同企业内网小号码互拨，可Lwork 拨打小号码，也能让小号码加入会议。

4。Lwork用户的语音呼叫权限需可控可调 (可选)

5 工号@公司 或者 邮箱作为账号登录

6 1.建立讨论组后，我想删掉，不知道在哪里删？ 请考虑增加删除功能

ps –xf |grep ftp

kill 进程号

关于ubuntu ftp server的安装、配置

<http://www.metsky.com/archives/551.html>

linux用户及用户组管理命令

<http://justcoding.iteye.com/blog/1929063>

js录音程序recordjs， audio标签播放

webrtc传送离线语音

招聘、嵌入地图，搜索3公里内的招聘企业，或者求职者。

Python smtp查询连接是否active

import smtplib

def create\_conn():

conn = smtplib.SMTP('smtp.gmail.com', 587)

...

return conn

def is\_connected(conn):

try:

status = conn.noop()[0]

except: # smtplib.SMTPServerDisconnected

status = -1

return True if status == 250 else False

或者用sendmail函数是否产生exception来检测，是否已经登录。

This method will return normally if the mail is accepted for at least one recipient. Otherwise it will raise an exception. That is, if this method does not raise an exception, then someone should get your mail. If this method does not raise an exception, it returns a dictionary, with one entry for each recipient that was refused. Each entry contains a tuple of the SMTP error code and the accompanying error message sent by the server.

This method may raise the following exceptions:

[**SMTPRecipientsRefused**](http://docs.python.org/2/library/smtplib.html#smtplib.SMTPRecipientsRefused)

All recipients were refused. Nobody got the mail. The **recipients** attribute of the exception object is a dictionary with information about the refused recipients (like the one returned when at least one recipient was accepted).

[**SMTPHeloError**](http://docs.python.org/2/library/smtplib.html#smtplib.SMTPHeloError)

The server didn’t reply properly to the HELO greeting.

[**SMTPSenderRefused**](http://docs.python.org/2/library/smtplib.html#smtplib.SMTPSenderRefused)

The server didn’t accept the *from\_addr*.

[**SMTPDataError**](http://docs.python.org/2/library/smtplib.html#smtplib.SMTPDataError)

The server replied with an unexpected error code (other than a refusal of a recipient).

Unless otherwise noted, the connection will be open even after an exception is raised.

下一步，更新公司信息

<http://emarketing.biz.163.com/net/wangluoxianzhuang_5.html>

优先：

2 修改个人信息的时候，如果账号是激活状态 ，且公司邮箱没有改的话，就不需要再发验证邮件了 ok

3 他要改企业邮箱的话 单独改。如果不一样的话，就需要重新发激活邮件 ok

4 修改个人信息的时候成功后，返回和登录一样数据。暂不改，前端写cookie

1 公司信息输入、查询 okvi

4 seeker登录，如果点发布职位，选择角色，则弹出输入企业邮箱，发送新的switch\_role，带上邮箱地址。解绑邮箱则发送switch2seeker ok

5 公司信息需要共享：邮箱验证之后，完善公司需要根据邮箱后缀名,列出相应的公司，如果没有则进入创建公司,

微信

curl -o index.html <http://wcg.shuobar.cn:8088>

cat /proc/cpuinfo | grep MH

taskset 指定任务运行在指定的cpu上。

<http://blog.csdn.net/yfkiss/article/details/7464968>

<http://www.cnblogs.com/edwardlost/archive/2010/10/23/1858991.html>

taskset –c 0 ./test

查看cpu情况：top 按1 看各cpu的运行情况，找到test对应的pid，比如18888，然后

taskset –pc 1 18888

top -p 6457 [f -> j] 查看某进程当前运行的cpu核心号

turbostat 查看cpu状态

<http://microbyte.blog.chinaunix.net/uid-15007890-id-3037121.html>

备忘：

1 Traffic.sh 的用法

./traff.sh eth0 &

crontab –e 然后增加0 \* \* \* \* /usr/bin/cuttrafflogs.sh

2 根据top，中的pid，显示proc/pno 中的limits

nmap -sT 14.17.102.3

gdb /usr/local/lib/erlang/erts-5.10.2/bin/beam.smp core

ulimit -n 4096

tar -zcvf - ebin|openssl des3 -salt -k shuoba2013\_14 | dd of=beams.ver.log

范例一：将整个 /etc 目录下的文件全部打包成为 /tmp/etc.tar  
[root@linux ~]# **tar -cvf /tmp/etc.tar /etc** <==仅打包，不压缩！  
[root@linux ~]# **tar -zcvf /tmp/etc.tar.gz /etc** <==打包后，以 gzip 压缩  
[root@linux ~]# **tar -jcvf /tmp/etc.tar.bz2 /etc** <==打包后，以 bzip2 压缩

爬虫 dhtcrawler

<http://www.cppblog.com/kevinlynx/archive/2013/06/20/201175.aspx>

Rebar erlang的发布工具非常好

<http://www.cnblogs.com/panfeng412/archive/2011/08/14/2137990.html>

Ubuntu修改ip地址和网关

<http://www.cnblogs.com/empire/archive/2011/01/10/1931877.html>

ubuntu 重启系统

sudo shutdown -r now 现在立即重启

ubuntu自启动脚本

<http://linux.chinaitlab.com/administer/871943.html>

1） 将你的启动脚本复制到 /etc/init.d目录下

    以下假设你的脚本文件名为 test。

    2） 设置脚本文件的权限

    $ sudo chmod 755 /etc/init.d/test

    3） 执行如下命令将脚本放到启动脚本中去：

    $ cd /etc/init.d

    $ sudo update-rc.d test defaults 95

    注：其中数字95是脚本启动的顺序号，按照自己的需要相应修改即可。在你有多个启动脚本，而它们之间又有先后启动的依赖关系时你就知道这个数字的具体作用了。该命令的输出信息参考如下：

    update-rc.d: warning: /etc/init.d/test missing LSB informationupdate-rc.d: see

    卸载启动脚本的方法：

    $ cd /etc/init.d

$ sudo update-rc.d -f test remove

Lyx:和他很不合拍，算了。

## 关于说吧

8.31.2014 21：40：

[gw\_test\_copy@58.221.60.37 和gw\_copy1@58.221.60.37节点](mailto:gw_test_copy@58.221.60.37%20和gw_copy1@58.221.60.37节点%20) sip用pcmu，看是否会down。

[gw\_test1@58.221.60.37](mailto:gw_test1@58.221.60.37)节点，sip是g729，[g729节点g729@58.221.60.37](mailto:g729节点g729@58.221.60.37)，即avscfg:get\_node(erl\_g729). 为'g729@58.221.60.37'。

明天看结果。

问题1： w2p中的codec释放有问题。可能会导致codec用光。

4.30.2014昨天已经解决。但是昨晚发现占用太快，释放太慢，导致ilbc用满，并发量500左右就会出现。今天w2p中解决，stun\_locked的时候占用。今天再上去验证下。

奇怪！换上去居然出现segment error，好好研究下

其中w2p.erl是有问题的。

奇怪下面这个,也出现异常，看来是codec的acquire和release的时机问题。

算了，决定还是回退版本吧。回退到rtp\_sup的版本。回退版本也麻烦。

再试下简单把acquire\_codec放到stun\_locked处来处理。

4.30 21：00之后 wrtc3节点版本是回退到gw\_run的，wrtc1还是最新版本。运行没有问题，但是www退出了，因为没有启用ulimit –n 40960,

5.1 20:48分开始www的report.log中出现wrtc1中的ilbc用完的错误，即

=ERROR REPORT==== 1-May-2014::20:49:13 ===

Error in process <0.15388.261> on node 'wrtc1@14.17.107.196' with exit value: {{badmatch,{1,1024}},[{rrp,rrp\_get\_web\_codec,1,[{file,"src/rrp.erl"},{line,777}]},{w2p,acquire\_codec,1,[{file,"src/w2p.erl"},{line,157}]},{w2p,handle\_cast,2,[{file,"src/w2p.erl"},{line,94}]},{my\_server,lprocess,3,[{file,"src/my\_server.er...

结果到20：53分，wrtc1就segment error了。但是wrtc3没有出现过ilbc满的情况。

从23点开始，启动wrtc3和wrtc4，都是gw\_run老版本试试。

5.2-14：00，发现wrtc3udp不够，当达到124个用户的时候。原因是昨晚已经异常了，没有重启。

发现wrtc1昨晚20：53分down之前，ilbc资源已经不够了，特别是，4.29晚开始，ilbc资源已经异常了。比如：ilbc为1的资源，释放之后54406: 14-04-29 16-57-11 <0.25734.68> : codec released {{ilbc,1,277,280,{140,140}},再也没有被申请过，要么是并发问题导致该数据区没有释放，要么释放出现异常。

考虑这样一种解决方法，用wcg\_node\_monitor启动wcg节点，并且监控wcg节点的资源使用情况，如果接近最大值，则将此节点不纳入选择范围，进入待删除范围（wcg\_disp:pause\_node(Node)），但还在资源收集范围，待其资源占用数为0，则退出。与上面删除动作的同时，增加一个新的节点，节点名

5.2 晚：800线稳定运行一晚，turbostat的cpu显示96%，top月80%。

5.3 昨天下午开始通过设置3个节点（wrtc3/4/5全部为gw\_run），每个节点限制路数在400路之下，目前没有看到ilbc资源异常的问题，也没有节点down的现象。估计之前路数达到600上的时候，这样会出现异常，并且掉住之前的资源不释放，而且，如果并发数太多的话，icdc的循环次数会太多。

今天准备上4个节点，每个节点限制在300路。量放到900线

5.3晚上18：20开始，增加wrtc6，如下



增加一个判断，如果ilbc资源号，大于当前的calls，则说明，有泄漏。记录下。

5.3 20:11:21 Wrtc4 down了 在此之前一直overload，之后手工重启，后来wrtc5又down了。来自wcg\_err.log的分析。

后续陆续都down了，特别是wrtc5，居然串口没有反应。

5.4 19:37 wrtc3 down了

把wrtc3起来Wrtc5又down了

10：00 在rtp和rrp上增加ip和port的过滤，先换wrtc5。发现rrp上加了，浏览器听不到，rrp回退后可以了。全部替换wrtc3/4/5/6的rtp

5.5 20:00左右wrtc5 down了

5.5 22:00 今晚wrtc6和wrtc3表现非常奇怪，量特别少，测试下来发现wrtc6的呼叫，qos越来越差。*rtt: "99.22", jitter: "10.91",越来越大rtt: "88.60", jitter: "18.77"*

22：15 重启了wrtc3之后，wrtc3好了很多。

5.6：15：00 躺床上想到一个方法，用wcg\_monitor起wcg节点，监控他，并且做如下工作：

1 wcg收到的udp包，都发到wcg\_monitor，wcg\_monitor收到后存放到数据区中，如果再收到，替换原来的udp包，一直到收到wcg节点down的消息后，monitor将此包写入文件，并注释，此时wcg节点down。

2 wcg收到的udp包，如果ip/addr非法，则发到monitor进行记录

5.7 11:18 好像只升级了wrtc3/wrtc4，wrtc5/wrtc6没有last包过来

问题2： my\_timer:send\_after(?TALKTIMEOUT,timeover); 1 振铃的时候就开始设置定时器？ 2 并且此定时器没有清除？

待解决：

问题3 wcg介入节点加上自动和手动进入故障切换状态

已经增加了手动进行切量的操作。如果wcg节点down，则发出10000的ccalls

问题4 Check\_origin的问题

已经解决

问题5 Voip\_ua的最大通话时长定时器确定已经cancel了？

已经解决，在voip\_ua中

Error in process <0.11047.0> on node 'wcg@14.17.107.196' with exit value: {{badmatch,1},[{rrp,handle\_info,2,[{file,"src/rrp.erl"},{line,395}]},{my\_server,lprocess,3,[{file,"src/my\_server.erl"},{line,79}]}]}的问题，似乎在access not allow跨域的时候出现。确认下

4/27

196上将14.17.102.3上的so同步到196的wcg上，看wcg segment error的现象能否消除？

1 将14.17.102.3上的so同步到196的wrtc2目录下，看是否有segment error的现象。

有了产生了Segmentation fault (core dumped

2 重新恢复之前的so，再试

4/28 下午1：40左右： wrtc2又出现./go: line 2: 7986 Segmentation fault (core dumped) yaws --conf yaws.conf --name wrtc2@14.17.107.196 --erlarg "-setcookie sbxyz +K true"

发现每次出现上面的问题都是起了2个wcg节点。

E:\work\repos\_bak\code\server\voice\erlang\_source\_so\c\_src目录下的文件都有锁保护，而E:\work\repos\_bak\code\server\voice\gw\csrc的没有。我用E:\work\repos\_bak\code\server\voice\erlang\_source\_so\c\_src的全部覆盖了E:\work\repos\_bak\code\server\voice\gw\csrc，

错了，g.729没有办法改。

Sublime 用正则表达式搜索，比如

(ilbc,1,)|(ilbc\:1 )，表示搜索“ilbc,1,”或者“ilbc:1”的字符串

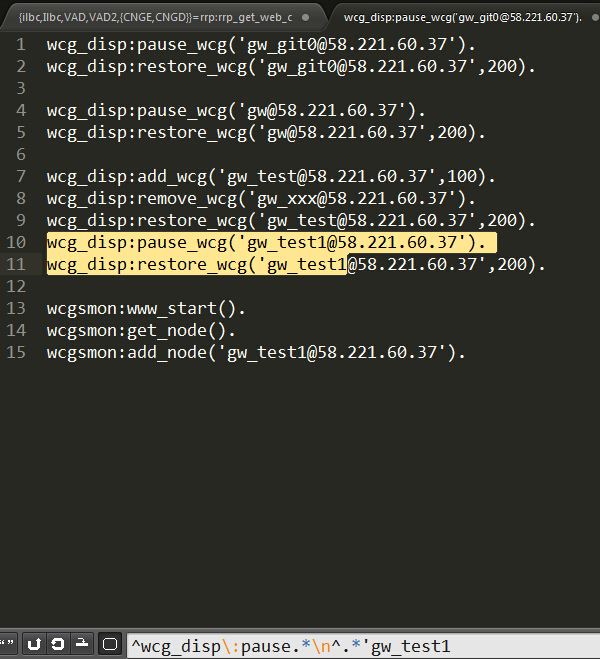
正则表达式搜索中文，**[[:unicode:]]，在sublime中验证通过**

Sublime将erl\_xxx:xxx(a,b,c…)替换为?APPLY(erl\_xxx,xxx,[a,b,c…])

(erl\_[^:()$\s]+):([^:()\s$]+)\(((.\*))\) 用?APPLY($1,$2,[$3)]替换,有点问题，如果参数有函数调用的话。

**Erlang动态库的加载路径设置：**

linux一般默认加载库的地址是：/usr/lib和/usr/local/lib，但是如果你自己新编了库不在这两路径下，你就要加一下地址。在/etc/ld.so.conf加入你要添加的路径，加好之后ldconfig一下，可能要用到root权限的。以后就可以用了



Ltalk cookie yxwabc name ltalk [company@10.32.3.38](mailto:company@10.32.3.38) tmp:month\_charge()

www: sudo yaws --name ayu@ltalk.com --conf yaws.conf --erlarg "-setcookie yxwxyz" –daemon

[rateserver@10.32.3.38](mailto:rateserver@10.32.3.38) [service@10.32.3.38](mailto:service@10.32.3.38) [company@10.32.3.38](mailto:company@10.32.3.38)

erl -name service@10.32.3.38 -setcookie yxwxyz -pa ebin -pa yxa/ebin -s xengine

E:/work/webrtc支持amr编码总结.txt

Ssh自动登录

<http://www.douban.com/group/topic/19654908/> 免用户名/免密码

# [Linux Shell远程执行命令（命令行与脚本方式）](http://www.cnblogs.com/ilfmonday/p/ShellRemote.html)

http://www.cnblogs.com/ilfmonday/p/ShellRemote.html

voip\_instance.lwStartVoip("yc138997633888","sz9918017813673@my\_token@my\_finger",{userclass:"registered",qos\_report:function(x){console.log(x)},talking:function(x){console.log("talking!!!!!!")},ringing:function(x){console.log("ringing")},peerhangup:function(){console.log("peerhangup!!!!!!")}})

voice节点关闭打印和日志功能

logger:disable(all).

部署完成58.221.59.169节点管3个wcg（58.221.59.168，59.238，60.117），rtp不check ip和port，qos要好些。

58.221.60.37上面管的3个节点，是check ip和port的。

防火墙ufw

–

ufw delete allow 15000:17000/udp

ufw delete allow 15000:17000

备忘：

嗯，对了，还有个问题要考虑下，上传个人图像 企业logo 办公环境图片，放在不同的文件夹下面

Python生成pdf文档 <http://blog.chinaunix.net/uid-17291169-id-3241613.html>

Erlang：生成pdf文档： <http://beebole.com/blog/erlang/convert-html-to-pdf-with-full-css-support-an-opensource-alternative-based-on-webkit/>

解决pdf中文乱码问题

<http://www.myexception.cn/flash/1463420.html>

kp\_partner  16:04:06  
这周末把简历 和公司信息部分搞定  
还有屏蔽公共邮箱、简历导出  
 公司信息填写 已有公司选择 新建公司。  
kp\_partner  16:05:31  
 公司主页

貌似离发布职位还有点 时间。

就是求职意向填写的时候，给个大概的符合的岗位数目，比如，填写30000的时候，显示多少职位满足条件，以免他填的过高，这个后面可以考虑

1 未登录的进入登录页面，登录之后进入用户原本的页面。

2 增加模板内容填充，格式kp?con=xxx, meta\_data会自动找xxx.htm的文件填入kp.template。

3 不同角色显示不同菜单项

重整旗鼓  22:18:37  
current和登录的信息显示还没搞，明天搞下。那几个页面能确认么？  
  
都要考虑到。 C:\Users\ADMINI~1\AppData\Local\Temp\3]6GJ(EWN[K2P[Z6B`6B`2H.gif  
重整旗鼓  9:52:45  
恩，放在content当中，就没有问题了  
kp\_partner  9:52:50  
感觉我们这个目录结构太乱了，后面越来越多 也只有我们自己知道。  
重整旗鼓  9:53:20  
目录结构？你是说url  
kp\_partner  9:53:34  
不是URL，是 文件目录  
这个先就这样吧。   
kp\_partner  9:54:38  
第一步是功能。   
你整下那个模板结构吧。 C:\Users\ADMINI~1\AppData\Local\Temp\3]6GJ(EWN[K2P[Z6B`6B`2H.gif

重整旗鼓  10:23:31  
上次那个红色下划线那个，感觉还比较麻烦，放到后面，如何？  
重整旗鼓  11:31:49  
我把导航的menu部分也改了，href改成了kp?con=xxx的方式，你把相应的文件名改为xxx.htm吧，里面的内容也改成去掉header/footer的形式哦。

Wcg需要的so()： erl\_cng.so/erl\_g729.so/erl\_ilbc.so/erl\_isac\_nb.so/erl\_opus.so/erl\_vad.so/erl\_resample.so/erl\_amr.so/erl\_amix.so

dd if=shash |openssl des3 -d -k shuoba2013\_14|tar xf –

提高语音质量：

1 增加缓存防抖动

2 ss侧的rtcp发送去掉。或者搞成rrp一样。

Finereport试用

1 新版lwork计费问题 ok

2 新版lwork视频问题 ok

3 git版本管理问题

说吧防攻击措施

新思路：

1 是不是从ss那侧发来的包导致down？增加ss侧的打印

2 将codec单独搞个节点

3

1 换上gw\_git版本， 换成opus 换成opus之后，同样被攻击的down了。今天看到还是ilbc（7.20上午），现在将gw\_git换成opus，gw\_git0还是ilbc。同样被攻击了。没用。

2 测试防火墙功能，将端口disable，直接发包到该端口，看是否能收到包。

测试结果：

ufw allow 22

ufw allow 80

ufw allow 8088

ufw allow 8081

ufw allow 5060

ufw allow 19305

ufw allow

ufw allow proto udp from any to any port 55000:57000

<http://wiki.ubuntu.org.cn/Ufw%E4%BD%BF%E7%94%A8%E6%8C%87%E5%8D%97>

这时候，如果ufw enable，则媒体不通，的确是收不到包，结果ok。

原理，通过server主动连接client的方式，来规避攻击。需要使用19305端口，这个有待确定

测试下不同端口号的情况，比如向外发送到端口号1，看端口号2发来的消息能否收到。

2 增加ice这个环节，配合ufw和www/wcg

3 通话状态，手工发送一个媒体包，看是否当成有效包，按理，应该扔掉，端口/地址不符

4 在rrp中增加向monitor记录收包的情况，带上ip地址/端口，没用，这个包手工去试，根本不会down。

5 js和www中修改加密规则，已修改验证在www\_t上， 还没有部署到www上。7.19日晚上11点多已部署 上https？ 实践证明，加密是没有用的。不对，兼容了以前的接口。

6 判断rtp包中的seq/ts，是否匹配，顺便将缓存放进去。

7 tokens路径修改，token的url改为 /lwork/auth/get\_tokens?uuid=lwk321；2 tokens获取的机制有问题，我这里会控制一下，获取token的请求我会做频率控制，避免以前大量带宽被占用的情况。 包括获取请求的源ip地址，可否将所有的合法的ip地址列表发我一份？已经修改，在www\_t上验证通过，还没有部署到www。

8 把rtp中的rpc:call(monitor\_node,set\_last….去掉。全部换成gw\_git版本了。

9 将接口的节点信息隐藏起来，最好不用poll的方法，用之前的long poll/。

10 7.20下午，down机比较多，发现有大量的rtp bad msg，导致大量写文件，会不会是写文件这个原因造成的？现在将这个地方注释掉，观察一下。（From 7.20 18：00）发现只有gw\_git0节点(ilbc)down了一次（7.20 22:02:07），而gw\_git1（opus）一直没有down过，再观察。如果iblc再down，就改为opus了。7.21 下午 gw\_git1 down过一次，gw\_git0down了3次。

11获取端口不从第一个开始，始终往上增1.以免乱包目标明确

9 打包gw\_git

今天7.26.2014，可能正如吴起讲的，感觉不像攻击，感觉是拨打和挂机频繁会导致down掉，应该是那几个底层的编解码函数保护不够。

先复现记录操作历史，底层操作函数单独节点，所有的erl\_xxx的调用全部用宏定义替换，远程调用，并记录调用信息，如果返回{badrpc,nodedown},则停止记录。

用正则表达式。

1 在调用erl\_xxx的地方增加调试日志记录。具体：（根本）

1. 通过rpc:call的过程，将对底层的调用打印出来
2. 在erl\_xxx.c中，增加in\_use的判断；erl\_cng等还没有保护
3. 在。C中，下标增加游标，逐次递增。
4. 在c中，增加保护，在icdc和idtr的时候

已经加完，那么怎么在有攻击的时候启动和退出呢：（已经在37上验证）

1. 将monitor进程移到www中，从www获取要监控的wcg。
2. Monitor监控到有wcg被down的时候，设置2分钟定时器，设置down状态。超时的话，则清楚down状态。
3. www在choose\_wcg的时候，去wcgsmon那边看下，是否有down标志，
   1. 如果有down标志，则调用gw\_test1节点的统计呼叫次数，如果成功，表明gw\_test1活着，则选择gw\_test1节点。
   2. 其他情况，choose\_wcg不变
4. 另外www在add\_wcg和remove\_\_wcg时，都会同步到wcgsmon监控任务中。

2 在www地方增加choose wcg的策略，具体：（暂时就选择最小的）

1. >50的不选
2. <3的优选
3. 访问monitor节点的down信息，如果down之内的3秒不接受新的呼叫。
4. 每个节点轮流使用，各使用10分钟，除非达到100，就转走。每次选择从并发数小的开始。

下一步计划：

1 将各个节点运行路径统一到gw\_git0，工作路径统一在一个目录的子目录下，空的。减少工作量。

2将手机上的新改动及gw\_test同步到196上。

手机部分：

1 将url改为wcgsj.shuobar.cn

2 增加加密工作。服务器已经加好，wcg域名增加mobile1的域名，以前的还保留

# [[Erlang 0033] 接入Erlang控制台的几种方法](http://www.cnblogs.com/me-sa/archive/2012/01/18/erlang0033.html)

<http://www.cnblogs.com/me-sa/archive/2012/01/18/erlang0033.html> 这篇文章很值得读

erl -setcookie sbxyz -name node\_3@14.17.107.196 -remsh [www\_t@14.17.107.196](mailto:www_t@14.17.107.196)

要退出，则ctrl+g q退出。

查看硬盘空间

 df -lh

sudo nethogs eth1

任务计划

炼化Lwork

1. Lwork服务器在客户Khobar办公室需能同时通过专线和internet连接到香港的SS中来，首选专线，当专线出现故障时，再选择internet，确保电话畅通。

2. Lwork用户沙特必须要通过内网和外网两个地址连接到Lwork服务器，当用户在沙特的6个内网办公室登陆Lwork时，必须首选内网，因为内网实现了专线，效果有保障，当内网出现问题，或者客户在外部网络时，则需要能通过外网登陆Lwork。 这样才能保证客户的使用面广泛，要不只能在内网或者外网访问，都是不完善的。

3. 请考虑实现通过PBX话机直接拨叫Lwork用户的功能， 只有这个功能实现了，Lwork才能和PBX真正融为一体。

4. 如有可能，请将Lwork PC to PC的基本呼叫功能，从视频呼叫里面分离出来，放到电话呼叫里面，增加一个PC to PC的选项。这样才是常规的使用方式。

目前任务

1 墨龙游戏 接口整理及提供(目前作为第一优先级) 至少需要2周时间，墨龙CTO希望，本周先提供点对点音频 点对点视频 打电话 短信这四个接口。

2 说吧红米手机单通问题(第一优先级,困难:目前没有服务器) 比较麻烦，人力时间机器都缺乏

3 协鑫开发，讨论过，工作量比较大，客户要求3/4个月内提供。

4 手机客户端外包，这个是共同需求。

5 中石化炼化公司conf pc加入/ phone to pc

6 RTX,电话会议测试

7 版本roadmap

8 ltalk和中兴IT部的数据同步。中兴组织架构及人员信息的变动同步到ltalk中，初步预计需要2周时间，难点，组织架构中兴的编号规则和ltalk不一样，而且，中兴的部门也在频繁调整，不是静态不动的，因此工作量比较大，需要重新调整编号规则，和中兴统一，否则，无法自动同步数据。

9 增加一条，说吧平台的防攻击。平静了一个月之后，从8月27日开始，攻击又开始了。

1/2/9 目前是最高优先级。9今天晚上已经找到代码漏洞了，修改还比较麻烦，目前已经暂时做了规避。减少影响范围

接口整理：

1 VOIP落地

2 pc2pc

3 video2video

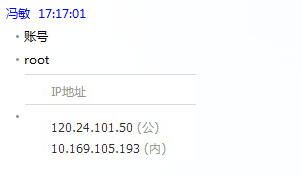
4 callback

5 conf

6

基本定位攻击来源了，g729模块被攻，以下两个ip地址的包

119.145.99.35

  
  
逄立冬 2014/9/5 17:17:48  
  
pw：mllwk888

几个事情：

1 优化上行话音质量

在rtp接收处，增加20个包的缓冲。并设置10个包间隔的循环定时器，以下2种情况下要将包发出去：

1. 包的seq和预期一致,不进缓存直接丢过去.
2. 包达到20个包的长度
3. 定时器到达后,和上次相比,冲里面的内容没有变.
4. 好像起作用了,乱序的包会正序后发出
5. 增加如果上行丢包,立即发送send\_nack的消息

2 部署国内webrtc/sip网关，部署fc2fc上的sip网关

3 增加环回的动作

start\_sip\_call(State=#state{test=true, rtp\_pid=RtpPid, rrp\_port=RrpPort, rrp\_pid=RrpPid}) ->

PeerAddr = [{remoteip,["127.0.0.1",RrpPort]}],

ok = rrp:set\_peer\_addr(RrpPid, PeerAddr),

rtp:info(RtpPid, {media\_relay,RrpPid}),

State#state{start\_time=now(),status=hook\_off};

4 信令跟踪抽空弄下

Sipsocket\_udp搞了个recre接口,重新生成signal文件

5 看看最新的webrtc的sdk接口文件

6

有空看看段的wmg,其中关于transport,看看,似乎可以一个udp端口,对应多个流.

另外有空将g729的icdc和xdtr搞成串行.

老刀建议日志，用于定位bug:2,3,6

固定信息: 时间,主叫号码,被叫号码

Rtp上行: 每秒收到的总字节数,字节速率,包总数,包速率,转发到rrp的包总数,包速率.  
Rtp下行: 每秒收到来自rrp的包总数,包速率;下行发送的总字节数,字节速率,包总数,包速率.

Rrp上行: 每秒收到来自rtp的字节数/字节速率/包总数/包速率,发送的总字节数,字节速率,包总数,包速率,编码后的字节总数/速率/包总数/包速率.  
Rrp下行: 每秒收到来自udp的字节总数/速率/包总数/包速率;下行发送到rtp的字节总数/速率/包总数/包速率.

记录每秒的jitter等值

1：客户端sdk增加接口获取一下数据  
上行 （1）：每秒语音采集字节数，总字节数  
（2）：每秒语音编码字节数，总字节数  
（3）：每秒语音发送成功字节数，总字节数  
下行 （1）：每秒语音接受成功字节数，总字节数  
（2）：每秒语音解码字节数，总字节数  
（3）：每秒播放语音字节数，总字节数  
  
服务器同上，将上下行编码解码字节数据做为日志记录

erlang:get\_stacktrace()

法国服务器: lwkfrnc H…… => aixunda2014

and fc2fc H…….a

58.221.60.37/119.29.9.55 w…….

2014.12.20 工作总结

1 kp recruit模块/通信模块整合 上线运营

2 taobao运营

1 低价宣传 2 用户名/密码/手机绑定（可选）。

3 app 推送bug

4 iphone app

5 营帐系统，代理系统

1 将用户数据库从fc2fc新版lwork独立出来。

2 创建dth公司

3 单独的app，专为dth用。

4 提供开通/激活/去激活接口

5 余额查询访问接口

6 安中

7 会议讨论 语音识别/电话会议

8 app电话会议

9 企业通讯录

10 短信/IM/视频

12 企业自注册/自充值

13 说吧bgp,将全部话路切换到bgp，其他作为备用。

14 杭州web电话及录音项目。

15 app的意见反馈是乱码

16 dth组网及测试,

17 unlogined problem

19 带宽检测

20 p2p判断主被叫是否在一个节点，不在一起走落地。

21 voip出去，如果voice与gw跨机器，则如果被叫挂机，主叫不会释放。跟跨机器无关，是因为fc2fc上发出的invite消息中本机ip地址取得是内网ip，所以BYE消息，没有收到。

22 app做sip被叫单通，app=》主叫的方向不通。

23 移动呼叫中心

24 p2p及sip被叫回铃音及主叫呼叫提示音修改

25 整改rtp和rrp，各自负责自己的code/decode，rtp里面进行jitter buff，窗口按照网络状况rtcp。

26 腾讯bgp服务器购买

27

1 短信编码问题

2 二次拨号

3 法国服务器部署

5 香港服务器部署

4 客户端做被叫

6 媒体流不转发的方式

视频

7 回拨

8 苹果客户端语音sdk

9 g.729加入webrtc

10 dth的对账系统对接

11 视频会议客户端

12 android/iphone自充值功能（手机充值卡和支付宝充值）

13 电话会议voip做被叫 客户端

14 lwork/ltalk的维护

15 处理下rtcp 443下不能接收到的问题

16 处理下nack发到客户端不能起作用的问题

内部：

1 444端口分发，端口直接对应到mediapid。

2 w2p改造为多方统一控制处理。mediapid为成员

3 目前的rrp中的媒体处理，需要放到统一媒体处理模块，混音处理。进来的流是经过处理的；产生的流，可以1秒定时器判定前3名，声音最大的，进行混音。

4 视频，

5

杭州 徐总

考虑到公司的成长阶段和当前的盈利情况，为咱们公司制定了一个定制化的比较优惠的价格。按licence收费，整体上价格也比较优惠

按licence，每个license 1000元/月,包含

1、Lwork平台接口开发费用；

2、10,000分钟/月的电话时长；

3、每月的维护和运营费用；

Yangzong:

1 **拨号之前检测带宽，20ms发送2kbit包，检测回来的包，包括抖动/时延/丢包率。**

**G.729: PT load 20BYTE/20ms**

**Total： 54+20=74×50=3700BPS=29.6kbps**

**G.711: load=160BYTE**

**Total： 160+54=214×50=10700BPS=85.6kbps**

**带宽 编码方式**

**不用gw**

**>100kbps G.729/ilbc/g.711a/g.711u 20ms**

**>50k g.729 20ms/ilbc**

**>40k g.729 40ms g.729 20ms**

**>30k g.729 40ms**

**Gw >16k g.729 40ms/60ms/80ms/100ms**

**>10k amr/120ms**

**Fc2fc的端口号被限制了。**

**Webrtc的设置codec过程**

**Callmaker.java->StartCall=>**

**voiceBase.setRemoteIP(rslt.peerIP);**

**voiceBase.initCall(rslt.codec, rslt.setNack);**

**voiceBase.setDestPort(rslt.peerPort);**

**voiceBase.setDestRtcpPort(rslt.peerRtcpPort);**

**String rtn = doStartCall();**

**voiceBase.initCall：webrtcAPI.setSendCodec=>voice\_engine\_jni.cc->VoiceEngine\_setSendCodec=>voe\_data->codec->SetSendCodec**

**doStartCall:**

**String receive = voiceBase.startVoiceReceive();**

**if (receive.compareTo("ok") == 0){**

**String send = voiceBase.startVoiceSend();**

**voiceBase.startVoiceReceive:**

**controls.add(receiver);**

**controls.add(listen);**

**return LogicCalc.and(webrtcAPI, voiceChannel, controls, 0);**

**相当于MobileListen.start() && VoiceReceiver.start()**

**voiceBase.java->setAudioCodec=>** **webrtcAPI Jni VoiceEngine.java->setSendCodec**

**setSendDestination**

**voice\_engine\_jni.cc->VoiceEngine\_setSendDestination=> voe\_data->GetTransport(channel)** **->SetSendDestination**

**setRemoteSsrc**

**rtp\_rtcp\_impl.cc->ModuleRtpRtcpImpl::SetRemoteSSRC,** **ModuleRtpRtcpImpl 在Channel构造函数里面创建，即 \_rtpRtcpModule.reset(RtpRtcp::CreateRtpRtcp(configuration));**

**statistics\_proxy\_.reset(new StatisticsProxy(\_rtpRtcpModule->SSRC()));**

**rtp\_receive\_statistics\_->RegisterRtcpStatisticsCallback(**

**statistics\_proxy\_.get());**

**voe\_rtp\_rtcp.impl.cc**

**技术：**

**1 729 so替换。还不行，到底是啥问题？下标不从0开始。似乎没有问题了，观察中 2015/2/1**

**2 qvoice录音**

3 带宽检测

4 大学城lwork部署

**5 G.729上android， 张丛耸在搞。等待测试**

**6 客户端动态修改ip。**

**7 短信SMPP**

**8 g729**编解码还有问题 声音不清楚。搞两套系统 一套稳定版本用于市场给客户演示 一套用于内部测试用。

**商务：**

**1 大学城合同，已交由祝小林处理**

**2 2345浏览器**

**3 三网通信**

**4 吴起合同**

**5 成都**

8 云服务器购买 ok

查看erlang node down的方法：

[**http://blog.yufeng.info/archives/2746**](http://blog.yufeng.info/archives/2746)

[**http://blog.yufeng.info/archives/tag/cerl**](http://blog.yufeng.info/archives/tag/cerl)

**find . -name "\*.a" | xargs rm –f**

**find . -name "\*.pcm" | grep 1623365634**

**find . -name "\*.pcm" | egrep "firstqq(2679605877|1802258594|2486153570|2485707570|2486152475|1447045579|1446569129|2486902191|477054974|1461348840|1458217839|1448141315|1456507541|1436553129|1447915518|1437223036|439695130|1440570715|1464335536|1439745838|1449765502|1439238895)" | xargs -t -I{} cp {} ../vcr\_bak/**

**linphone+freeswitch**

**1 g729 2byte problem，导致程序崩溃，可以用linphone**

**2 reinstall frnace server**

1. **jf service for app**
2. **documents for app**
3. **gw1节点出现udp overrange的错误。**

**ltalk reset charge**

[**company@10.32.3.38**](mailto:company@10.32.3.38) **tmp:month\_charge().**

## Linux dns设置

修改/etc/resolv.conf

nameserver 202.102.24.35

指定了dns服务器的地址

## Erlang call java

**JAVA\_ERLANG**

**Ubuntu安装jdk1.7** [**http://openjdk.java.net/install/**](http://openjdk.java.net/install/)sudo apt-get install openjdk-7-jre

Erlang安装需要jinterface

## 快捷方式

ln -s 源文件 目标文件

## LTALK相关

**带宽占用：**

**1 有信：**

**40ms一个包 76Byte（包括mac/ip/udp/rtp），另外还有cn包，64字节，300kB/min（上下行）**

**微会：**

**30ms一个包，101Byte，620KB/min（上下行不对称）**

**Ltalk：**

**上行：120ms一个包，132Byte，55KB/min，8.6kb/s**

**下行：60ms一个包，93Byte，93KB/min，12.4kb/s。**

**注册/按账号充值/绑定号码/计费扣费/**

**5 营帐系统，代理系统**

**1 将用户数据库从fc2fc新版lwork独立出来。(需要么？)**

**2 创建dth公司**

**3 单独的app，专为dth用。**

**4 提供开通/激活/去激活接口**

**5 余额查询访问接口**

**目标：**

1. **2月底完成dth的相关运营工作，包括和营帐系统/用户的交互，主要包括：**
2. **3月底前完成自注册/自支付系统及相关的计费工作。**
3. **4月底前完成代理模式的相关工作。**

**具体内容：**

1. **dth相关运营工作，包括以下接口实现：**

* 接口：

1 营帐系统=》lwork （和羊总相关）

开通：字段 group（填dth） account pwd 绑定sip号码

激活： account

去激活： account

绑定上海联通号码： account phone 绑定DID号码

2 lwork=》营帐系统（和羊总相关）

查询剩余分钟数： 字段： account

查询详单： 字段： account，start/end起始日期/结束日期

3 app=》lwork （和羊总无关）

注册： 字段 account/pwd/username/

注册成功后强制要求用户设置登录密码

登录： 使用username登录，username/pwd

修改密码： account/oldpwd/newpwd

* 数据库支持，部署数据库节点。

1. **自注册/自支付系统及相关的计费工作：**

* 用户侧接口：基本完成，增加话单或者余额查询接口，同dth
* 分享，通过链接方式，由后台根据id重新生成apk文件。
* 二维码
* 支付相关接口，
  + 支付宝相关申请流程
  + 支付宝接口实现
  + App界面及接口实现
* 系统管理相关工作
  + 数据库节点部署
  + 计费策略节点
  + 系统计费查询及话单管理

1. **代理模式的相关工作。在2的基础上增加**

* 用户侧接口：增加推荐号
* 系统管理：
  + 增加代理portal管理页面，根据用户权限，可查询相关子树的注册/充值/话费使用情况。相关的代理申请及批准流程，提成策略设置/显示/申请/提醒。

**实施计划：**

**1 支付宝申请同步进行**

支付宝支付

**增加mnesia分布式处理部分，目前是lw\_register表。Node\_dist\_mng.erl中。**

**语音识别：**

**HTK：E:\work\thirdparty\识别 **

****

**Gongzuozongjie :**

**1 陈**

**2 袁**

1. **羊**
2. **验证码**
3. **微拓**
4. **Qq**
5. **Kp**
6. **Ios**
   1. **注册功能及界面**
   2. **登录功能及界面**
   3. **拨号盘界面**
   4. **拨打功能及通话界面及状态更新（直拨）**
   5. **通讯录及查询功能**
   6. **更多页面：个人资料/拨号盘声音设置/版本号显示、查询及更新下载/绑定号码显示/关于ltalk**
   7. **软件通话，拒绝来电功能。**
   8. **手机通话，拒绝软件来电功能**
   9. **通讯录及通话记录号码的状态显示功能**
   10. **微信/微博分享**
   11. **二维码及网站下载**
   12. **版本升级及下载**
   13. **被叫功能**
   14. **ios推送功能**
   15. **话单查询功能**
   16. **拨打功能及通话界面及状态更新（回拨）**
   17. 6个月后**上架appstore**
7. **Dth的did及附属号码/话单查询**
8. **微信公众平台开发**
9. **回拨扣费 ok**
10. **国际费率扣费**

2 申报材料

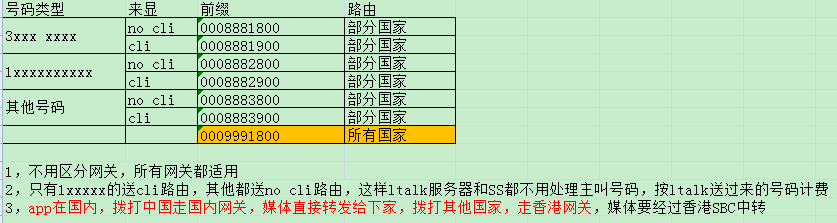
3ios p2p ios⬄android

Android《=》 ios ok

ios=ios

4 跨节点的p2p

5 android amr ios ilbc的问题



**主叫号码是座机的情况：前缀0008883900 第一路由送给百信网络的企业线，前缀为6；第二备份路由选择no cli的，比如东方科技的nocli。**

**友盟查问题：**

**export dSYMPath="$(find ~/Library/Developer/Xcode -iname '\*.dSYM' -print0 | xargs -0 dwarfdump -u | grep 2F05F8A0-B54E-3C6C-AACC-8C4A553380CD | sed -E 's/^[^/]+//' | head -n 1)";dwarfdump --arch=armv7 --lookup 0x43121 "$dSYMPath"**

**mnesia分布式部署数据不一致问题：**

[**https://www.safaribooksonline.com/library/view/erlang-programming/9780596803940/ch13s04.html**](https://www.safaribooksonline.com/library/view/erlang-programming/9780596803940/ch13s04.html)

**mnesia:set\_master\_nodes, mnesia:stop, mnesia:start**

**fc2fc 回拨 没钱会有异常service\_not\_available**

**修改系统的文件fd限制修改**

**vi /etc/security/limits.conf 后面增加**

**\* soft nofile 165536**

**\* hard nofile 165536**

**要做的事：**

**Linux查看流量**

**sudo ethstatus -i eth0**

**1 opennebula安装**

**Apt-get的代理设置** /etc/apt/apt.conf文件

Acquire::https::proxy <http://10.32.3.66:808>

Acquire::http::proxy “http://10.32.3.66:808”

Acquire::ftp::proxy “http://10.32.3.66:808”

**Openmeetings install**

[**http://www.linuxidc.com/Linux/2014-11/108863.htm**](http://www.linuxidc.com/Linux/2014-11/108863.htm)

**安装swftools**

wget <http://swftools.org/swftools-0.9.2.tar.gz>

tar -zvxf swftools-0.9.2.tar.gz

wget <http://www.ijg.org/files/jpegsrc.v7.tar.gz>

tar -zvxf jpegsrc.v7.tar.gz

wget <http://sourceforge.net/projects/freetype/files/freetype2/2.3.12/freetype-2.3.12.tar.gz/download>

tar -zvxf freetype-2.3.12.tar.gz

**关于swftools编译失败 make clean一下**

[**http://sffytian.blog.51cto.com/472262/370337**](http://sffytian.blog.51cto.com/472262/370337)

**关于xz解压缩失败 安装完xz 重新登陆就ok了？**

sudo apt-get install zlib1g-dev

**wget** [**http://download.savannah.gnu.org/releases/freetype/freetype-2.6.tar.gz**](http://download.savannah.gnu.org/releases/freetype/freetype-2.6.tar.gz)

**运行openoffice,**

**soffice --accept="socket,host=localhost,port=8100;urp;StarOffice.ServiceManager" --norestore --nofirststartwizard --nologo --headless &**

**soffice -headless -accept="socket,host=127.0.0.1,port=8100;urp;" -nofirststartwizard &**

**bigbluebutton安装**

[**http://docs.bigbluebutton.org/install/install.html**](http://docs.bigbluebutton.org/install/install.html)

1. **Update your server**
2. 安装jdk8， <http://www.cnblogs.com/a2211009/p/4265225.html>
   1. 解压到/usr/lib/jvm/jdk1.7.0\_60下
   2. sudo vim ~/.bashrc

**2. Install PPA for LibreOffice 4.4**

sudo apt-get install software-properties-common

sudo add-apt-repository ppa:libreoffice/libreoffice-4-4 （不行 出错,手工下载路径如下）D:\software\bigbluebutton\ LibreOffice\_4.4.6\_Linux\_x86-64\_deb.tar.gz

**3. Install key for BigBlueButton**

wget http://ubuntu.bigbluebutton.org/bigbluebutton.asc -O- | sudo apt-key add -

echo "deb http://ubuntu.bigbluebutton.org/trusty-090/ bigbluebutton-trusty main" | sudo tee /etc/apt/sources.list.d/bigbluebutton.list

sudo apt-get update

**4. Install ffmpeg**

To install ffmpeg, create a file called install-ffmpeg.sh and copy-and-paste in the following script

sudo apt-get install build-essential git-core checkinstall yasm texi2html libvorbis-dev libx11-dev libvpx-dev libxfixes-dev zlib1g-dev pkg-config netcat libncurses5-dev

FFMPEG\_VERSION=2.3.3

cd /usr/local/src

if [ ! -d "/usr/local/src/ffmpeg-${FFMPEG\_VERSION}" ]; then

sudo wget "http://ffmpeg.org/releases/ffmpeg-${FFMPEG\_VERSION}.tar.bz2"

sudo tar -xjf "ffmpeg-${FFMPEG\_VERSION}.tar.bz2"

fi

cd "ffmpeg-${FFMPEG\_VERSION}"

sudo ./configure --enable-version3 --enable-postproc --enable-libvorbis --enable-libvpx

sudo make

sudo checkinstall --pkgname=ffmpeg --pkgversion="5:${FFMPEG\_VERSION}" --backup=no --deldoc=yes --default

Next, make the script executable and run it.

$ chmod +x install-ffmpeg.sh

$ ./install-ffmpeg.sh

check that ffmpeg is installed by typing the command ffmpeg –version