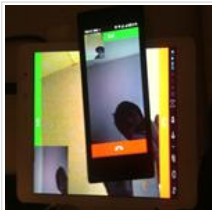


个人资料



qzlink

访问：42994次
积分：781
等级：[BLOG > 3](#)
排名：千里之外

原创：18篇
转载：130篇
译文：1篇
评论：10条

文章搜索

文章分类

voip (6)
asterisk (8)
freeswitch (34)
vc++ (0)
j2ee (14)
FMS/RED5 (0)
android/ios (3)
mysql (10)
jfinal (10)
jodd (2)
linux (54)
svn (12)
ios (1)
maven (3)
技术集合 (4)
lua (10)

[2017直通软考，拿证无忧](#) [程序员简历优化指南！](#) [程序员1月书讯](#) [云端应用征文大赛，秀绝招，赢无人机！](#)

lua定时器与定时任务的接口设计

2016-09-20 17:21 303人阅读 评论(0) 收藏 举报

分类：

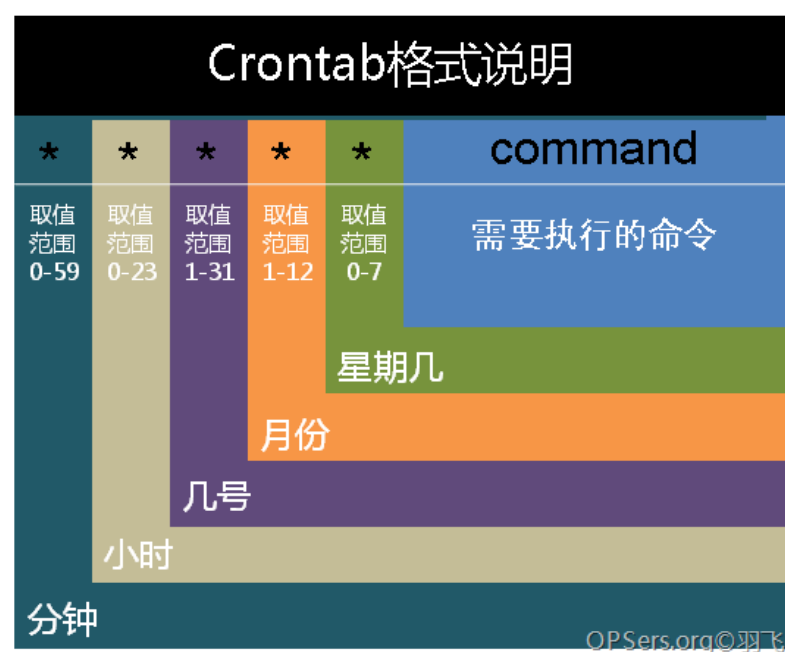
[linux \(53\)](#) [lua \(9\)](#)

在所有的服务器编程当中，定时任务永远是一个不可或缺的需求。最直接的需求就是，每天凌晨0点0分的时候总是有一大堆的各种精力重置。怎么来设计这个接口呢，想了几个方案：

- 每秒触发
- 每分钟触发
- 每整点触发
- 每天触发
- 每个月触发

oh no!不靠谱啊,如果这接口真设计成这样，得有多烂，灵光一现,unix下的crontab表达式非常完美的解决了这个问题。

附上crontab表达式的语法说明如下：



crontab特殊的符号说明：

PHP (1)

文章存档

2017年01月 (7)

2016年12月 (5)

2016年11月 (3)

2016年10月 (9)

2016年09月 (23)

展开

阅读排行

内存溢出的多种原因 (2482)

手机视频通话软件: (2060)

openfire 开发OO群 (1284)

android 打开 ppt p (1082)

MvSQL的经典用法 (823)

Jfinal 配置文件工具 (795)

提供手机视频通话: (771)

一组惊艳的NodeJs (728)

Freeswitch callbac (674)

利用phoenix进行H (672)

评论排行

手机视频通话软件: (10)

android 打开 ppt p (3)

Mvsal存储过程 (三 (0)

MvSQL中复合索引 (0)

Asterisk Oueue呼 (0)

企业版IDP的申请及 (0)

asterisk对接DID大 (0)

asterisk 命令 (0)

提供手机视频通话: (0)

使用shc加密shell脚 (0)

推荐文章

* 造轮子 | 如何设计一个面向协议的 iOS 网络请求库

* Android新特性介绍，ConstraintLayout完全解析

* Android 热修复 Tinker 接入及源码浅析

* 创业公司做数据分析 (六) 数据仓库的建设

* 【死磕Java并发】-----深入分析synchronized的实现原理

最新评论

android 打开 ppt pptx do
wgcla7: 哎，互联网一点信任都没有，垃圾搞来搞去干嘛！

"*"代表所有的取值范围内的数字。特别要注意哦！

"/"代表每的意思，如"/5"表示每5个单位

"-"代表从某个数字到某个数字

","分散的数字

crontab文件的使用示例：

30 21 * * * 表示每晚的21:30

45 4 1,10,22 * * 表示每月1、10、22日的4:45

10 1 * * 6,0 表示每周六、周日的1:10

0,30 18-23 * * * 表示在每天18:00至23:00之间每隔30分钟

0 23 * * 6 表示每星期六的11:00 pm

* */1 * * * 每一小时

* 23-7/1 * * * 晚上11点到早上7点之间，每隔一小时

* 8,13 * * 1-5 从周一到周五的上午8点和下午1点

0 11 4 * mon-wed 每月的4号与周一到周三的11点

0 4 1 jan * 一月一号的4点

看起来很复杂的样子，但其实够用就好，我们也不需要实现全部特性。

- 实现一个毫秒级别的定时器Update
- 根据这个update函数实现一个秒级别定时器
- 然后每秒取得自然时间与表达式中 分、时、几号、月份、星期几 分别匹配就可以实现了
- 由于定时器除了增加以外，可能还需要一个删除功能，那就再提供一个定时器命名的功能，用于增删改查定时器是本身
- 再加个测试函数。。完美

直接上代码：

```
1  -----
2  --任何一个记录产生一个实例
3  local Clock = {}
4  local Clock_mt = {__index = Clock}
5
6  local function __checkPositiveInteger(name, value)
7      if type(value) ~= "number" or value < 0 then
8          error(name .. " must be a positive number")
9      end
10 end
11
12 --验证是否可执行
13 local function __isCallable(callback)
14     local tc = type(callback)
15     if tc == 'function' then return true end
16     if tc == 'table' then
17         local mt = getmetatable(callback)
18         return type(mt) == 'table' and type(mt.__call) == 'fun
19     end
20     return false
21 end
22
23 local function newClock(cid, name, time, callback, update, arg
24     assert(time)
25     assert(callback)
26     assert(__isCallable(callback), "callback must be a functio
```

手机视频通话软件开发流
liuxin19900614: 服务器
安装好了不能用, 还望指
点, 是wind8的系统

android 打开 ppt pptx do
wlf535944903: 付费买源
码行么? 或者请您帮忙解
决问题 付费。能否留个
QQ? 或者加我157 0122
986

手机视频通话软件开发流
qzlink:
@qq1357725556: 只要你的
网络没问题, 可以用的。

手机视频通话软件开发流
qq1357725556: 下载
了, 拨了号码但是下面的
语音通话和视频通话都不能用

手机视频通话软件开发流
shen173869710: 加不到
你啊 你加一下我QQ 希
望你可以提供有偿的服务

android 打开 ppt pptx do
itcastldh: 啊咧, 求源码

手机视频通话软件开发流
ruru7989: 能否指点一下
关于网络电话方面的事情?
我的QQ1106989757

手机视频通话软件开发流
Xiao_ZJ: 你好, 我的
QQ:2725374959 可以交
流下吗

手机视频通话软件开发流
weishengwen2: 能留下
联系方式不? 跟你交流下
这方面的技术



```
27     return setmetatable({
28         cid      = cid,
29         name     = name,
30         time     = time,
31         callback = callback,
32         args     = args,
33         running  = 0,
34         update   = update
35     }, Clock_mt)
36 end
37
38 function Clock:reset(running)
39     running = running or 0
40     __checkPositiveInteger('running', running)
41
42     self.running = running
43     self.deleted = nil      -- 如果已经删除的, 也要复活
44 end
45
46 local function updateEveryClock(self, dt)
47     __checkPositiveInteger('dt', dt)
48     self.running = self.running + dt
49
50     while self.running >= self.time do
51         self.callback(unpack(self.args))
52         self.running = self.running - self.time
53     end
54     return false
55 end
56
57 local function updateAfterClock(self, dt) -- returns true if
58     __checkPositiveInteger('dt', dt)
59     if self.running >= self.time then return true end
60
61     self.running = self.running + dt
62
63     if self.running >= self.time then
64         self.callback(unpack(self.args))
65         return true
66     end
67     return false
68 end
69
70 local function match( left, right )
71     if left == '*' then return true end
72
73     -- 单整数的情况
74     if 'number' == type(left) and left == right then
75         return true
76     end
```

```

77
78     --范围的情况 形如 1-12/5,算了,先不支持这种每隔几分钟的这种特性吧
79     _,_,a,b = string.find(left, "(%d+)-(%d+)")
80     if a and b then
81         return (right >= tonumber(a) and right <= tonumber(b))
82     end
83
84     --多选项的情况 形如 1,2,3,4,5
85     --哎,luajit不支持gfind,
86     --for d in string.gfind(left, "%d+") do
87     --其实也可以for i in string.gmatch(left,'(%d+)') do
88     local pos = 0
89     for st,sp in function() return string.find(left, ',', pos,
90         if tonumber(string.sub(left, pos, st - 1)) == right th
91             return true
92         end
93         pos = sp + 1
94     end
95     return tonumber(string.sub(left, pos)) == right
96 end
97
98 local function updateCrontab( self, dt )
99     local now = os.date('*t')
100     local tm = self.time
101     --print('updateCrontab/now:', now.min, now.hour, now.da
102     --print('updateCrontab/tm', tm.mn, tm.hr, tm.day, tm.mon,
103     --print('match:',match(tm.mn, now.min), match(tm.hr, now.h
104     if match(tm.mn, now.min) and match(tm.hr, now.hour)
105         and match(tm.day, now.day) and match(tm.mon, now.month
106         and match(tm.wkd, now.wday)
107     then
108         --print('matching',self.name,self.callback,self.runnin
109         self.callback(unpack(self.args))
110         self.running = self.running + 1
111     end
112     return false
113 end
114
115 --遍历并执行所有的定时器
116 local function updateClockTables( tbl )
117     for i = #tbl, 1, -1 do
118         local v = tbl[i]
119         if v.deleted == true or v:update(1) then
120             table.remove(tbl,i)
121         end
122     end
123 end
124
125 -----
126

```

```

127 local crontab = {}
128 crontab.__index = crontab
129
130 function crontab.new( obj )
131     local obj = obj or {}
132     setmetatable(obj, crontab)
133     --执行一下构造函数
134     if obj.ctor then
135         obj.ctor(obj)
136     end
137     return obj
138 end
139
140 function crontab:ctor( )
141     --所有的定时器
142     self._clocks = self._clocks or {}
143     self._crons = self._crons or {}
144     --累积的时间差
145     self._diff = self._diff or 0
146     --已命名的定时器,设置为弱引用表
147     self._nameObj = {}
148     setmetatable(self._nameObj, {__mode="k,v"})
149
150     --取得现在的秒数,延迟到整点分钟的时候启动一个定时
151     self:after("__delayUpdateCrontab", 60-os.time()%60, functi
152         --在整点分钟的时候,每隔一分钟执行一次
153         self:every("__updateCrontab", 60, function ( )
154             updateClockTables(self._crons)
155         end)
156     end)
157 end
158
159 function crontab:update( diff )
160     self._diff = self._diff + diff
161     while self._diff >= 1000 do
162         --TODO:这里真让人纠结,要不要支持累积时间误差呢?
163         self._diff = self._diff - 1000
164         --开始对所有的定时器心跳,如果返回true,则从列表中移除
165         updateClockTables(self._clocks)
166     end
167 end
168
169 function crontab:remove( name )
170     if name and self._nameObj[name] then
171         self._nameObj[name].deleted = true
172     end
173 end
174
175 --通过判断callback的真正位置,以及参数类型来支持可变参数
176 --返回值顺序 number, string, number, function, args

```

```

177 --总的有如下5种情况
178 --1) cid,name,time,callback,args
179 --2) name,cid,time,callback,args
180 --3) name,time,callback,args
181 --4) cid,time,callback,args
182 --5) time,callback,args
183 local function changeParamsName( p1, p2, p3, p4, p5 )
184     if __isCallable(p4) then
185         if type(p1) == 'string' then
186             return p2,p1,p3,p4,p5
187         else
188             return p1,p2,p3,p4,p5
189         end
190     elseif __isCallable(p3) then
191         if type(p1) == 'string' then
192             return nil,p1,p2,p3,p4
193         else
194             return p1,nil,p2,p3,p4
195         end
196     else
197         return nil,nil,p1,p2,p3
198     end
199 end
200
201 function crontab:every( cid, name, time, callback, args )
202     --支持可变参数
203     cid, name, time, callback, args = changeParamsName(cid, na
204     __checkPositiveInteger('time', time)
205     local clock = newClock(cid, name, time, callback, updateEv
206     table.insert(self._clocks,clock)
207     if name and name ~= '' then
208         self._nameObj[name] = clock
209     end
210     return clock
211 end
212
213 function crontab:after( cid, name, time, callback, args )
214     cid, name, time, callback, args = changeParamsName(cid, na
215     __checkPositiveInteger('time', time)
216     local clock = newClock(cid, name, time, callback, updateAf
217     table.insert(self._clocks,clock)
218     if name and name ~= '' then
219         self._nameObj[name] = clock
220     end
221     return clock
222 end
223
224 --增加计划任务,精度到达分钟级别
225 --表达式:分钟[0-59] 小时[0-23] 每月的几号[1-31] 月份[1-12] 星期几[1
226 --          星期天为1 ,

```

```

227 --      "*"代表所有的取值范围内的数字
228 --      "-"代表从某个数字到某个数字
229 --      "/"代表每的意思，如"/5"表示每5个单位，未实现
230 --      ","分散的数字
231 --  如："45 4-23/5 1,10,22 * *"
232 function crontab:addCron(cid, name, crontab_str, callback, arg
233     cid, name, crontab_str, callback, args = changeParamsName(
234     --print(cid, name, crontab_str, callback)
235     local t = {}
236     for v in string.gmatch(crontab_str, '[%w._/,%-*]+') do
237         --如果可以转成整型直接转了，等下直接对比
238         local i = tonumber(v)
239         table.insert(t, i and i or v)
240     end
241     if table.getn(t) ~= 5 then
242         return error(string.format('crontab string, [%s] error!
243     end
244
245     local time = {mn = t[1], hr = t[2], day = t[3], mon = t[4]
246     local clock = newClock(cid, name, time, callback, updateCr
247     table.insert(self._crons, clock)
248     if name and name ~= '' then
249         self._nameObj[name] = clock
250     end
251 end
252
253 return crontab

```

再看看测试代码：

```

1 -- 传说中的测试代码
2 local function RunTests()
3     -- the following calls are equivalent:
4     local function printMessage(a )
5         print('Hello', a)
6     end
7
8     local cron = crontab.new()
9
10    local c1 = cron:after( 5, printMessage)
11    local c2 = cron:after( 5, print, {'Hello'})
12
13    c1:update(2) -- will print nothing, the action is not done
14    c1:update(5) -- will print 'Hello' once
15
16    c1:reset() -- reset the counter to 0
17
18    -- prints 'hey' 5 times and then prints 'hello'
19    while not c1:update(1) do
20        print('hey')

```

```

21     end
22
23     -- Create a periodical clock:
24     local c3 = cron:every( 10, printMessage)
25
26     c3:update(5) -- nothing (total time: 5)
27     c3:update(4) -- nothing (total time: 9)
28     c3:update(12) -- prints 'Hello' twice (total time is now 21
29
30     -----
31     c1.deleted = true
32     c2.deleted = true
33     c3.deleted = true
34
35     -----
36     --测试一下match
37     print('-----')
38     assert(match('*',14) == true)
39     assert(match('12-15',14) == true)
40     assert(match('18-21',14) == false)
41     assert(match('18,21',14) == false)
42     assert(match('18,21,14',14) == true)
43
44     --加一个定时器1分钟后执行
45     cron:update(1000)
46
47     --加入一个定时器每分钟执行
48     cron:addCron('每秒执行', '* * * * *', print, {'..... cr
49
50     cron:update((60-os.time()%60)*1000)
51     cron:update(30*1000)
52     cron:update(31*1000)
53     cron:update(1)
54     cron:update(60*1000)          --打印两次
55     end

```

也可以直接到 <https://github.com/linbc/crontab.lua> 下载代码

参考资料：

<http://www.cise.ufl.edu/~cop4600/cgi-bin/lxr/http/source.cgi/commands/simple/cron.c>

<https://github.com/kikito/cron.lua>

顶 踩
0 0

我的同类文章

linux (53)		lua (9)	
• 使用shc加密shell脚本	2017-01-19	• Mysql 增量备份和...	2017-01-17
阅读 19		阅读 16	
• Linux mail 命令	2017-01-17	• mysql 安装指定版...	2017-01-13
阅读 53		阅读 21	
• MySQL数据的binlo...	2017-01-09	• etcd : 从应用场景...	2017-01-02
阅读 28		阅读 17	
• linux下c语言实现tai...	2016-12-24	• 利用log4j+mongod...	2016-12-16



jQuery MiniUI



快速开发
Java、.N

猜你在找

各类接口测试实战合集
Qt网络编程实战之HTTP服务器
软件测试速成篇
Lua进阶教程
Lua与c++交互实战基础篇

使用spring定时器注解实现定时任务
Java 定时器的使用每天定时执行任务
Java 中Timer和TimerTask 定时器和定时
定时任务实现Timer TimeTask
jsp定时器web程序定时执行任务

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件
Docker OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript
数据库 Ubuntu NFC WAP jQuery BI HTML5 Spring Apache .NET
API HTML SDK IIS Fedora XML LBS Unity Splashtop UML
components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App
SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby
Hibernate ThinkPHP HBase Pure Solr Angular Cloud Foundry Redis
Scala Django Bootstrap

