**主机资源应用程序设计**

# 一、整体结构



如上图所示，主机资源有两个主线程：SELECT线程和定时器线程。

SELECT线程是主处理线程，负责管理的进程的检查/启动/重启、接收管理的进程的心跳连接并接收心跳信息更新进程信息、自动连接监控服务器，注册监控项，告警信息的发送，接收监控服务器的停启指令并执行。

定时器线程负责定时取主机的CPU信息、内存信息、硬盘信息并保存至主机信息当中，供select线程使用。负责定时通知SELECT线程检查连接状态，发送心跳等；定时通知SELECT线程检查进程，进程是否存在/是否无心跳要重启等；定时通知SELECT线程发送主机资源信息给监控服务器；定时将进程信息写入monirun.xml

# 二、运行说明

支持WINODWS/LINUX/AIX/HP\_UX主流操作系统

SELECT线程采用非阻塞模式。

## 启动初始化

1. 读取XML配置，初始进程信息。
2. 初始化连接池

连接池序号从0开始，大小为监控服务器连接2+侦听端口1+199，总共为202。即只能有199个有心跳的管理进程。

1. 初始化进程信息，从monirun.xml读取上次管理的进程信息(单独停止本程序，但不停止管理的进程，适合增加管理的进程)
2. 开始侦听
3. 启动SELECT线程
4. 启动定时器线程。

## SELECT线程

1. 扫描连接池，需要连接的调用连接。
2. 响应读事件，读数据并处理或从连接池取一个未使用连接信息建立新连接。
3. 响应写事件，连接监控服务器的返回信息，判断连接是否成功。
4. 检查监控服务连接，发送心跳或主动断开连接
5. 监控信息，发送主机资源信息。
6. 检查进程信息，若没有启动，有则检查心跳，根据配置是否重启，若有不正常事件发生，发送告警信息。

## 定时器线程

1. 定时取主机的CPU、内存、硬盘信息
2. 定时通知SELECT线程检查
3. 将进程信息写入monirun.xml

# 三、配置说明

|  |
| --- |
| <?xml version="1.0" encoding="GB2312"?>  <package>  <head>  <公共配置>  <日志级别 remark="日志跟踪级别0严重错误1重要错误2一般错误3警告4提示警告5调试信息">5</日志级别>  <日志文件 remark="记录日志信息的文件名,前加日期">monitor.log</日志文件>  <日志目录 remark="日志存放的目录"></日志目录>  <主机ID remark="主机ID">1000</主机ID>  <主机名称 remark="主机名称">风控服务器</主机名称>  <节点ID remark="节点ID">1010</节点ID>  <心跳时间 remark="和监控服务器的心跳时间,秒">5</心跳时间>  <监控服务连接过期时间 remark="当连接长时间未有数据时断开,秒">600000</监控服务连接过期时间>  <连接过期时间 remark="管理的进程心跳连接过期时间,秒">600</连接过期时间>  <心跳侦听端口 remark="接收管理的进程的心跳端口">29000</心跳侦听端口>  <无心跳重启时间 remark="当配置为无心跳重启时，超时此时间无心跳则重启">30</无心跳重启时间>  </公共配置>  <监控服务端口配置>  <H1 ip="20.13.0.217" port="13911" remark="监控服务器H1端口" />  <H2 ip="20.13.0.217" port="13912" remark="监控服务器H2端口" />  </监控服务端口配置>  <进程守护>  <1001>  <进程名称>通讯接口机服务器</进程名称>  <程序名 remark="程序的名称，不带路径">testdaemond.exe</程序名>  <程序路径 remark="绝对或相对路径"></程序路径>  <运行目录 remark="绝对或相对目录">D:\newmonitor</运行目录>  <启动参数 remark="启动进程所需的参数">1001</启动参数>  <停止脚本 remark="停止进程的脚本，为空直接kill掉"></停止脚本>  <无心跳是否重启 remark="为1时表示两个心跳时间没有收到心跳则重启进程">1</无心跳是否重启>  <是否接受控制 remark="为1时表示可由此程序进行停止">1</是否接受控制>  </1001>  </进程守护>  </head>  <pack>  </pack>  </package> |

**日志级别：**为5时将DEBUG信息写入文本日志，大于5将一些更多的信息写入文本日志，在正常运行一段时间后无问题建议将日志级别设为4，压力测试时设为3或2。

**日志文件：**文件名不带路径，假设配置的是x，当为文件日志时，文件名为yyyymmddx，当为报文日志时，文件名为yyyymmddx.log。

**日志目录：**不设置时则为当前目录下的log目录，此log目录会自动建立，当设置目录后，要确定目录确实存在，程序不会自动建立目录。

**主机ID：机器的编号**

**主机名称：主机中文名称**

**节点ID**：本应用程序的节点编号

**心跳时间**：和监控服务器的心跳时间，单位秒，注意JAVA版的监控服务器不主动发送心跳，也无心跳应答。

**监控服务连接过期时间**： 当连接超时此时间未收到数据主动断开连接，注意监控服务器无心跳应答，所以这个值要设置的很大，否则和监控服务器的连接会断开。

**连接过期时间**：心跳连接的过期时间，当此连接建立但超时此时间未收到数据时，关闭此连接。

**心跳侦听端口**：接受所守护的进程的心跳连接。心跳报文为报文头+报文体组成，报文头固定为8个字节，表示报文体的长度，不足8位前补0。报文体为KV格式报文，无固定报文头，其中必须有node\_id字段，此字段和进程编号相同，表示是哪个进程所发的心跳包。

**监控服务端口配置：**配置和监控服务器的两个连接，第一个为H1，第二个为H2。

**进程守护**：配置守护的进程信息，子节点名为进程在整个总线范围内的进程编号

【**进程编号**】：唯一编号，进程发送心跳时必须在KV报文体的node\_id节点填上此值。

**进程名称**：进程的中文名称

**程序名**：运行的程序名

**程序路径**：运行程序所在的目录

**运行目录**：进程的运行目录

**启动参数**：进程启动的参数

**停止脚本**：停止进程的脚本，若无则使用kill来停止进程

**无心跳是否重启**：当进程没有心跳时，是否由进程守护重启该进程(kill后重启)

**是否接受控制**：当接收控制时，可以进程守护来重启、停止、KILL该进程，这样可以在总线的任意一点来控制该进程的状态

**注意**：通过进程ID检查是否和进程信息中的一致，windows和unix下面不一样，windows检查进程ID对应的程序名及程序目录否和进程信息中的一致,**但JAVA程序比较特殊，调用WINDOWS API根据进程ID取JAVA的程序名和程序目录出错，取不出，如果有管理java程序的话，建议每次重启时停止所有管理的进程并删除monirun.xml再启动monitor，否则java程序有可能会启不全**。Unix下面检查进程ID对应的程序名、运行目录及程序目录否和进程信息中的一致。

**因此，在windows下面，管理的进程配置里，如果程序名相同，那么程序目录一定不要相同。**