# 例行性工作调度

## 种类

- crontab指令 — 例行性
  - 可循环的周为分钟/小时/每月/每年
  - 指令执行相关/etc/cron.d/支持
  - （必须有crond服务）
- at指令 — 单发性
  - 处理仅执行一次就结束调度的指令
  - （必须有atd服务）

## 仅执行一次的工作调度

- atd的启动
  - systemctl restart atd
  - systemctl enable atd
  - systemctl status atd
- atd的运行方式
  - 工作目录 — /var/spool/at/
    - 使用at指令产生所要执行的工作调度，并将此工作以文本文件的方式写入此工作目录中
    - 由工作管理程序(atrun)服务的常用和执行
    - 此时记录的工作内容，除了原本的执行指令外，还会将当时的环境变量、umask、工作目录等等，一并记录下来，未来执行时，会以当时的环境来执行
  - 安全限制
    - 允许与拒绝文件，写在此文件中的用户才能使用at，不在其中的用户不能使用at（即使其也不在/etc/at.deny）
    - /etc/at.allow — 严格
    - 当/etc/at.allow不存在时，搜寻此文件；写在此文件中的用户不能使用at
    - /etc/at.deny — 松散
- 用法
  - at [-mldv] TIME
  - at -c 工作号码
- 选项与参数
  - -m — 当at的工作完成后，即使没有输出信息，也将以email通知用户该工作已经完成
  - -l — "at -l 等于"atq"，列出目前系统上所有该用户的at调度
  - -d — "at -d 等于"atrm"，取消一个在at调度中的工作
  - -v — 使用较明显的时间格式列出at调度的工作列表
  - -c — 列出后面接的该工作的实际指令内容
- TIME的时间格式
  - 定义出"何时要执行at这项工作"的时间
  - 功能
  - 在今日的HH:MM时刻执行；若该时刻已超过，则明天的HH:MM执行该工作 — HH:MM
  - 强制规定在某年某月某天的某时某刻进行该工作 — HH:MM YYYY-MM-DD
  - 格式
    - 用上HH:MM[am|pm] [Month] [Date]
    - 在某个时间点/再加几个时间点后才进行 — HH:MM[am|pm]或now + number [minutes|hours|days|weeks]
- at指令
  - 下达at指令后，会进入at shell环境，可下达多条指令，必须使用绝对路径下达指令 — at shell环境
  - 由于在at工作调度的使用上，系统会将该项at工作放立出当前用户的at目录中，直接交给系统的at程序接管，下达完at指令后可以脱机离线，系统将继续执行其工作调度 — 脱机继续工作
  - 工作管理
    - 查询系统当前所有的at工作调度 — atq
    - 移除该工作调度 — atrm 工作编号
  - 判断是否要立刻执行后台程序（当前CPU的工作负载小于0.8时，才进行后下达的工作调度任务） — batch指令 — 闲时启动

## 总体工作流程

- crond
  - 1.主动读取/etc/crontab，/var/spool/cron/*，/etc/cron.d/*等配置文件，并根据"分、时、日、月、周"等时间设置各调度各项工作
  - 2.根据/etc/cron.d/hourly的设置，主动启动/etc/cron.hourly目录下，及之所有应该目录下的可执行文件
- anacron
  - 3.根据/etc/cron.hourly/0anacron文件，每小时主动执行anacron，并调用/etc/anacrontab配置文件
  - 4.根据/etc/anacrontab文件中的设置，每天、每周、每月查分析/etc/cron.daily/，/etc/cron.weekly/，/etc/cron.monthly/内部可执行文件，以执行固定周期需要执行的工作

## 循环执行的例行性工作调度

- 用户的设置
  - 安全限制
    - /etc/cron.allow — 不在此文件内的用户不允许使用crontab
    - /etc/cron.deny（系统默认保留） — 记录在此文件的用户不可使用crontab
  - 工作目录
    - /var/spool/cron/ — 用户通过crontab指令创建工作调度后，该工作将被记录到"/var/spool/cron/用户名"的同名目录中（不可使用vi直接编辑该文件，输入语法会造成无法执行cron）
    - /var/log/cron/ — cron执行的每一项工作都将被记录到此（可检查此目录判断是否被植入木马）
  - crontab指令
    - 用法 — crontab [-u username][-l|-e|-r]
    - 选项与参数
      - -u — 只有root才能进行这个任务，即帮其他用户新建/删除工作调度
      - -e — 编辑crontab的工作内容 — "crontab -e"这个crontab其实就是/usr/bin/crontab这个可执行文件
      - -l — 查阅crontab的工作内容
      - -r — 移除所有的crontab的工作内容（仅要移除一项使用-e编辑）

**0 12 * * * mail -s "at 12:00" dmtsai < /home/dmtsai/.bashrc**

#分 时 日 月 周 |<==========指令串==========>|

| 代表意义 | 分钟 | 小时 | 日期 | 月份 | 周 | 指令 |
|---|---|---|---|---|---|---|
| 数字范围 | 0-59 | 0-23 | 1-31 | 1-12 | 0-7 | 所要执行的指令 |

| 特殊字符 | 代表意义 |
|---|---|
| *（星号） | 代表任何时间都接受时间意思！举例来说，范例一内那个日、月、周星期 *，意代表着“不论何日，何日的礼拜几的 12:00 都执行记录指令”的意思！ |
| ,（逗号） | 代表分隔时段的意思。举例来说，如果要下达的工作是 3:00 与 6:00 时，就会做：<br>0 3,6 * * * command<br>时间参数还是有几点，不过第二行是 3 & ，代表 3 与 6 都适用！ |
| -（减号） | 代表一段时间范围内，举例来说，8 点到 12 点之间每小时的 20 分都进行一项工作：<br>20 8-12 * * * command<br>因每周第二天是减号 8-12 啦，代表 8,9,10,11,12 都适用啦(意思！ |
| /n（斜线） | 那个 n 代表数字，亦即是“每隔 n 单位间隔”的意思，例如每五分钟进行一次，则：<br>*/5 * * * * command<br>很简单吧！用 * 号 /5 来搞定，也可以写成 0-59/5 ，相同意思！ |

- 工作指令编辑中字段及辅助字符的意义
- 系统的配置文件
  - cron服务检测 — 最迟检测限制单位为"分钟"，故cron会每分钟去读取一次/etc/crontab和/var/spool/cron里面更新的内容
  - crond服务读取配置文件位置
    - 用户运行 — /var/spool/cron/*
    - 系统运行
      - /etc/crontab（纯文本文件） — 功能 — 对于系统的例行性任务，直接编辑写入此文件即可
      - /etc/cron.d/* — 功能 — 可放置自定软件的crontab定时指令的配置文件脚本。

```
01 * * * * root run-parts /etc/cron.hourly
```

    - run-parts脚本会在大约5分钟内随机选一个时间来执行/etc/cron.hourly目录内的所有可执行文件（放置在/etc/cron.hourly/内的文件必须是能被直接执行的脚本，而不是分、时、日、月、周的设置值）
    - /etc/cron.d/0hourly

## 可唤醒停机期间的工作任务

- anacron
  - 功能 — 检测并执行超过调度期限日未被执行的调度任务（每小时被主动执行一次，/etc/cron.hourly/0anacron）
  - 用法
    - **anacron [-sfn] [job]..**
    - **anacron -u [job]..**
  - 选项与参数
    - s — 开始一连串的执行各项工作（job）— 会依据时间记录文件的数据判断是否进行
    - f — 强制进行，而不去判断时间记录文件的时间戳记
    - n — 立刻进行未进行的任务，而不延迟（delay）等停时间
    - u — 仅更新时间记录文件的时间戳记，不进行任何工作。
    - job — 由 /etc/anacrontab 定义的各项工作名称。
  - 配置文件 — /etc/anacrontab — 文件结构图示

```
RANDOM_DELAY=45          # 随机给予最大延迟时间，单位是分钟
START_HOURS_RANGE=3-22   # 延迟多少小时内应该要执行的任务时间

1       5       cron.daily          nice run-parts /etc/cron.daily
7       25      cron.weekly         nice run-parts /etc/cron.weekly
@monthly 45     cron.monthly        nice run-parts /etc/cron.monthly
天数    延迟时间  工作名称定义      实际要进行的指令串
# 定义每日为天，延迟时间单位为分钟，工作名称定义可自订，指令串到通常与 crontab 的设置相同
```

  - 天数 — anacron执行每下次的间期（/var/spool/anacron内的时间记录）相差的天数，若超过此天数，就准备开始执行，若未超过，则不执行后续指令
  - 延迟时间 — 当确定超过天数特别要同执行调度工作，则按此设置值延迟此时间，即让它相同此时间差值的资源冲突
  - 工作名称定义 — 无实际意义，常与启动目录资源名称相同
  - 指令串 — 通过run-parts来处理

试用版 XMind:ZEN 试用版