Lec 15 例行性工作排程

什么是例行性工作排程

有些工作是例行性的：定期备份
有些工作是临时发生的：有人约了去徒步

Linux 工作排程的种类：at、cron

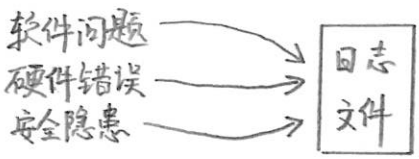
at：
处理仅执行一次的排程，必须要有 atd 这个服务 ← SUSE SLES12 上找不到了

crontab：
处理例行性的排程，必须要有 crond 这个服务 ← SUSE SLES12 上是 cron.service
（慢慢被淘汰了，以后推荐使用 systemctl）

Linux 系统上常见的例行性工作（CentOS）

① log rotate
适时地将日志文件的数据挪一挪，让旧数据与新数据分别存放。

② logwatch



主动分析日志文件

③ locate DB 的更新
更新 locate 的数据库

④ man page DB 的更新

⑤ tmpwatch
移除暂存文件

仅执行一次的工作排程

atd 的启动

systemctl ...

at 的运作方式

使用 at 这个指令来生成要运作的工作，并将这个工作以文本文件的方式写入 /var/spool/atjobs 目录内。
该工作便能等待 atd 这个服务的取用与执行了。
可以利用 /etc/at.allow 与 /etc/at.deny 这两个文件来对 at 的使用进行限制。

if /etc/at.allow exists:
　　其中的使用者才能使用 at，其余的用户不行
elif /etc/at.deny exists:
　　其中的使用者不能使用 at，其余的用户可以
else
　　只有 root 才能使用 at

在默认情况下，系统通常会保留一个空的 /etc/at.deny 文件。

实际使用 at 指令

例子见原书。

当使用 at 时会进入一个 at shell 的环境来让用户下达工作指令，此时最好使用绝对路径。

at 的 standard output 、standard error output 会被传送到执行者的 mailbox 去。（因为 at 在运作时，会跑到当时下达 at 指令的工作目录去）

若没有 stdout 或 stderr，则不会有 mail，可设置 [-m] 让 at 无论如何也会发 mail 说明是否执行了指令。

同时，at 也是"背景执行"的。

at 、atq 、atrm 、batch

↑ query　↑ remove　↑ 系统有空时才进行背景任务

by the way，uptime 可以观察 1、5、15 分钟的平均工作负载。

循环执行的例行性工作排程

使用者的设定（限制）

/etc/cron.allow 与 /etc/cron.deny，与 at 的逻辑相同。

crontab 的记录在 /var/spool/cron 下。 ← /var/spool/cron/<user>

cron 执行的每一项工作都会被记录到 /var/log/cron 这个日志文件中。

语法

crontab [-u <username>] [-l|-e|-r]

进入 vi 的编辑画面 → i542016@vsa11779357:~> crontab -e

6 个字段：

代表意义	分钟	小时	天	月	周	指令
取值范围	0~59	0~23	1~31	1~12	0~7	<...>

0=7=Sunday

特殊字符：

*	any
,	and
-	range
/<num>	interval

系统的配置文件：/etc/crontab 、/etc/cron.d/*

`crontab -e` 是针对使用者的 cron 来设计的；← 注意 /var/spool/cron/<user> ☆

`vim /etc/crontab` 是针对系统的例行性任务的。

crontab -e ：/usr/bin/crontab 这个可执行文件

vim /etc/crontab ：纯文本文件，需 root 身份

cron 这个服务的最低侦查频率限制是分钟，所以 cron 会每分钟去读取一次 /etc/crontab 与 /var/spool/cron 里面的内容。

```
root @vsa11779357:~#  cat  /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# Example of job definition:
#        .---------- minute (0-59)
#        |  .------- hour (0-23)
#        |  |  .---- day of month (1-31)
#        |  |  |  .- month (1-12) OR jan,feb,mar,apr,...
#        |  |  |  |  .- day of week (0-6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sa
#        |  |  |  |  |
#        *  *  *  *  *  <user-name>  <command to be executed>
```

注：

MAILTO=root

当 /etc/crontab 这个文件中的例行性工作的指令发生错误时，或是该工作的执行结果有 stdout 或 stderr 时，默认由系统直接发一封 mail（stdout与stdin）给 root。

PATH=...

执行文件的搜寻路径（Lec10 BASH 的执行文件路径问题）

<user-name>

执行<command>的身份

crond 服务读取配置文件的位置
- /etc/crontab
- /etc/cron.d/*

} 与系统的运作比较有关的配置文件

- /var/spool/cron/* ← 与用户自己的工作比较有关的配置文

/etc/cron.d/* 是什么样的呢？

```
root@vsa11779357:~#  ls  /etc/cron.d
dsupdstat
sap.corp-byd-webmin-sysstats

root@vsa11779357:~# cat  /etc/cron.d/dsupdstat
10 * * * *  root  /opt/imal/bin/dsupdstat-prepare-clearlogs.sh
```

如果开发了自己的软件，该软件要拥有自己的 crontab 定时指令时，就可以将配置文件放/etc/cron.d

/etc/cron.hourly/：系统每小时定时执行；由 crond 执行

/etc/cron.daily/：系统每日定时执行
/etc/cron.weekly/：系统每周定时执行
/etc/cron.monthly/：系统每月定时执行

} 由 anacron 执行

总结：

写入 /var/spool/cron/<user>

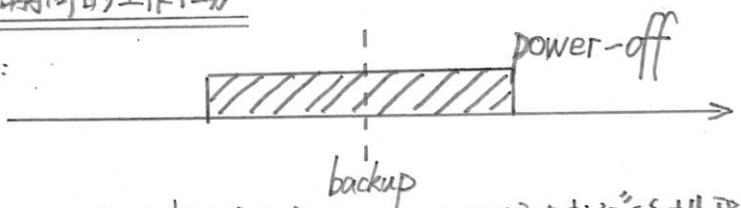
个人的行为	crontab -e	由于/etc/crontab是大家都能读的，所以crontab -e更...
系统维护管理	vim /etc/crontab	管理方便，同时容易追踪
开发软件	vim /etc/cron.d/<file>	使用全新的配置文件

一些注意事项

1. 资源使用
   "不要集中在同一时间启动"
2. 取消不要的输出项目
   "/dev/null"
3. 安全问题
   "木马"
4. 周与日 不共存

可唤醒停机期间的工作任务

case：



anacron：主动帮用户进行"时间到了但没有执行"的排程

执行完毕或无需执行任何排程时anacron就停止了。

什么是 anacron

anacron 每个小时被 crond 执行一次，然后 anacron 再去检测相关的排程任务有没有被执行。

anacron与 /etc/anacrontab

anacron 其实是一支程序，并非一个服务！

anacron 这支程序在 CentOS 中已经进入 crontab 的排程，每小时会被执行一次。

/etc/cron.d/0hourly （其配置文件在 /etc/cron.hourly/中）

/etc/cron.hourly/0anacron

让 anacron 最先进行，是为了让时间戳先更新，以避免 anacron 误判 crontab 未进行的...

/usr/sbin/anacron (/etc/anacrontab)

.exe　　.conf { cron.daily / cron.weekly / cron.monthly

伪代码略。