

【1-11】进程与定时任务

笔记本： 备课_linux

创建时间： 2022/4/17 8:53

更新时间： 2022/4/29 11:47

作者： 兰鸣人花道

1、进程的基本概念

- windows的进程?
 - qq
 - chrome
 - 进程和程序的概念
 - 程序/软件用代码决定程序的行为，存在存储介质里，比如硬盘
 - 程序运行起来，就是操作系统把存储介质上的可执行文件加载到内存中去运行，这个运行的程序就叫做进程
 - 程序是静态的文件
 - 进程是程序的一次执行过程，进程是动态的
-

2、进程的创建

- 可执行程序运行起来，就叫进程
-

3、查看进程的命令ps

- shell就是一个进程
 - 进程ID，也叫PID
 - PID, process(进程)
 - 操作系统用数字来标识进程
 - ps命令显示进程的ID
 - 语法
 - 不带参数，同一个用户同一个终端
 - -ef, e表示所有，f表示full-format,全格式化，尤其是PPID和command内容
 - aux, ax表示所有，u表示user-oriented
 - ps -ef | grep java
 - ps -ef结果说明：
 - 1、UID，用户ID
 - 2、PID，进程ID
 - 3、PPID，父进程ID
 - 4、C，cpu占用率
 - 5、STIME，开始时间
 - 6、TTY，开始此进程的TTY---终端设备
 - 7、TIME，此进程运行的总时间
 - 8、CMD，命令名
-

4、后台运行进程

- 前台进程和后台进程
 - 前台进程
 - shell里面正在执行的和用户进行交互的进程叫前台进程
 - 比如ps，ls都是前台进程，时间短

- 有的命令执行时间长，比如备份程序
 - 特点，占用终端
 - 后台进程
 - 可以在命令后加`&`在后台执行
 - 比如`bash test.sh &`
 - 后台运行，意味着不从标准输入读入字符
 - 后台运行的进程我们叫后台进程，或者后台任务
 - 进程前后台切换
 - `jobs`
 - 列出后台进程命令
 - 比如执行 `bash test.sh &`
 - `fg`
 - 将后台进程切换到前台运行
 - `fg`，上一个后台命令
 - `fg <number>` 将后台进程调到前台，比如`fg 1`
 - `ctrl + z` 可以挂起进程，注意`jobs`显示的状态为`stopped`，**暂停了**
 - `bg`
 - `bg <number>` 可以将进程在后台运行，**继续在后台执行**
 - 比如执行`ctrl + z` 挂起进程后，再用命令`bg 1` 让进程在后台继续执行
-

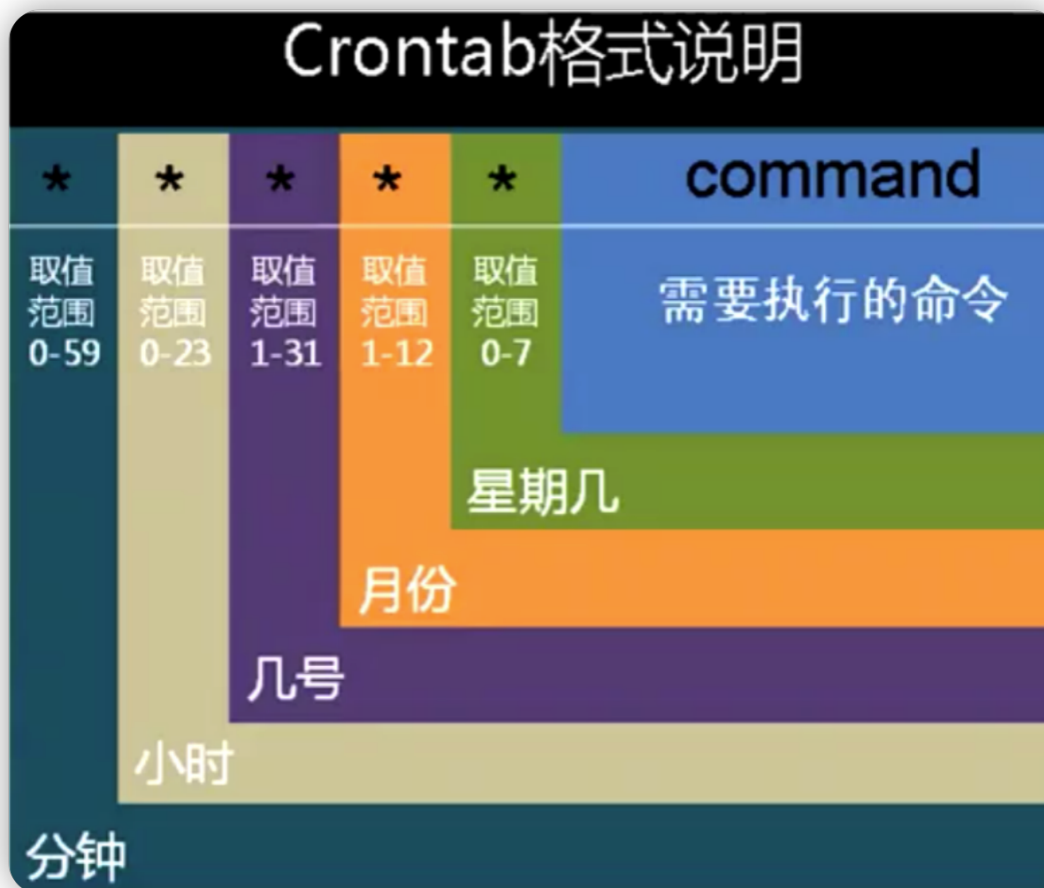
5、如何终止进程

- 进程的两种终止方式
 - 自行终止
 - 任务执行完成，比如`ps`，`ls`
 - 用户让其退出，比如`vi`，`exit`
 - 异常退出
 - 手动杀死进程
 - `kill PID`，杀死进程
 - `kill -9 PID`，强制杀死进程
 - `ctrl + c`
 - 只有`owner`和`root`才能杀死进程
-

6、定时任务

- 为什么需要定时任务
 - 定时备份
 - 日志太大，定时清理下目录
 - 定时同步数据
 - 清理`logs`目录下超过1个月的日志文件
 - 每天对数据库做一次备份(导出)
 - 思路?
 - 我可能会去写一个脚本，然后每天执行?
- linux下的定时任务
 - `crontab` 周期性的执行
- `crontab`
 - 格式说明
 - 前5个字段表示时间，最后一个表示要执行的操作
 - 分钟 0到59
 - 小时 0到23
 - 日，就是几号的意思 1-31
 - 月份 1-12
 - 星期几，从0到6
 - 特殊字符说明
 - 星号(`*`) 代表任何时刻都接受的意思，例如`month`字段如果是`*`号，表示每月
 - 逗号(`,`) 可以用逗号隔开的值指定一个列表范围，例如，“1,2,5,7,8,9”
 - 减号(`-`) 可以用整数之间的减号表示一个整数范围，例如，2-6 表示“2,3,4,5,6”
 - 正斜线(`/`)

- 可以用正斜线制定时间的间隔频率，如"0-23/2"表示每两个小时执行一次
- 同时正斜线可以和星号一起使用，例如*/10，如果用在minute字段，表示每十分钟执行一次



- 语法:
 - `crontab [-u user] [-l|-r|-e]`
 - `-u`: 只有root用户才能进行这个任务，也即帮助其他用户创建/删除crontab的工作调度
 - `-l`: 查看
 - `-e`: 编辑
 - `-r`: 删除所有的crontab工作内容，如果要删除单个，请用`crontab -e`
- 工作原理
 - 当用户使用crontab这个命令来新建工作调度之后，该项工作会被记录到`/var/spool/cron`里面，而且以账号来判断
 - 如，root使用crontab后，它的工作会被记录到`/var/spool/cron/root`
 - 注意不要用vi直接编辑该文件，可能会由于某种原因比如语法问题，导致无法执行cron
 - `/var/log/cron`文件，可以查看cron的日志
- 实例:

```
[root@localhost test]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# 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,sat
# | | | | |
# * * * * * user-name command to be executed

# 表示在每天的午夜0:30运行backup命令
30 0 * * * backup

# 每1分钟执行一次command
* * * * * command

# 每小时的第3和第15分钟执行
3,15 * * * * command

# 在上午的8点到11点的第3分钟和第15分钟执行
3,15 8-11 * * * command

# 每天晚上的22: 00执行
* 22 * * * command

# 每周星期一晚上的22: 00执行
* 22 * * 1 command
```
