嵌入式操作系统

7 Ubuntu中的系统初始化

陈香兰（xlanchen@ustc.edu.cn）

计算机应用教研室@计算机学院
嵌入式系统实验室@苏州研究院
中国科学技术大学
Fall 2014

December 9, 2014

Outline

Outline

Ubuntu的init系统历史变迁

- Ubuntu是Debian的下游版本。
- Linux发行版在内核启动之后，都需要启动一个init进程（1号进程）

1. System-V init，又写作sysvinit
2. Upstart：http://upstart.ubuntu.com
   event-based init daemon
   ubuntu-6.10开始逐步用upstart代替原来的System-V init
3. Systemd：https://wiki.ubuntu.com/systemd
   根据目前了解的信息
   1. ubuntu-15.04将正式开始
   2. ubunut-14.10已经可以安装，待完善。

Outline

upstart

- Upstart is an event-based(基于事件的) replacement for the /sbin/init daemon which handles
  1. starting of tasks and services during boot,
  2. stopping them during shutdown and
  3. supervising them while the system is running.
- In essence, Upstart is an event engine: it
  1. creates events,
  2. handles the consequences of those events being emitted and
  3. starts and stops processes as required.
- 关于Upstart的详细介绍，
  参见http://upstart.ubuntu.com/cookbook/，或者
  参见/usr/shard/doc/upstart下的README（需要解压缩），或者
  使用下列命令可以查看upstart和init相关信息

  ```
  man init
  ```
  或者
  ```
  man upstart
  ```
- Ubuntu-14.04：upstart-1.12.1

阅读upstart的README

- /usr/shard/doc/upstart下的README（需要解压缩）

1. Where are initscripts installed?
2. How are initscripts started and stopped?
3. What order are initscripts started and stopped in?
4. How do I find the current/previous runlevel?
   - 运行runlevel命令
   - 运行man runlevel
5. How do I change the runlevel?
6. How do I change the default runlevel?
7. How do I shutdown the machine?
8. How do I change the behaviour of Control-Alt-Delete?
9. ...

阅读upstart帮助信息 I

man upstart

- 关于init
  - init is the parent of all processes on the system, it is executed by the kernel and is responsible for starting all other processes; it is the parent of all processes whose natural parents have died and it is responsible for reaping those when they die.
- 关于job
  - Processes managed by init are known as jobs and are defined by files in the /etc/init directory.
  - job与job配置文件
- 关于事件event
  - job会因为事件的发生而自动的start或者stop （根据其配置文件）

阅读upstart帮助信息 II

- 关键的几个事件
  - startup事件
    - ★ 由init产生
  - starting、started、stopping、stopped事件
    - ★ 产生：随着job状态的改变
- Job的状态和lifecycle（参见man upstart）

Upstart中的事件

- Init即Upstart，是一个基于事件的守护进程
- 事件代表系统状态的变化，产生的事件会发送给init进程
  - job的状态也是系统状态，因此job状态的变化会产生相关事件
- 事件可以带参数，如runlevel事件
- job的配置文件中，列出了job启动、停止的条件，
  这些条件与事件有关
- job在运行时，可能会根据需要触发某些事件
  （具体参见相关配置文件）

Upstart中的job及其配置文件 I

- 一个job由其job配置文件定义：

$$< \text{name} > .\text{conf}$$

- Job的状态（参见man upstart）

序号	当前状态	Goal	
		start	stop
1	waiting	starting	n/a
2	starting	pre-start	stoping
3	pre-start	spawned	stopping
4	spawned	post-start	stopping
5	post-start	running	stopping
6	running	stoping	stopping
7	pre-stop	running	pre-stop/stopping(*)
8	stopping	killed	killed
9	killed	post-stop	post-stop
0	post-stop	starting	waiting

Upstart中的job及其配置文件 II

- Job的lifecycle（参见man upstart）：考虑启动一个job
  1. Initially the job is ”at rest” with a goal of ’stop’ and a state of ’waiting’.
  2. The goal is changed from ’stop’ to ’start’ indicating the job is attempting to start.
  3. The state is changed from ’waiting’ to ’starting’.
  4. The starting event is emitted denoting the job is ”about to start”.
  5. Any jobs whose ’start on’ (or ’stop on’) condition would be satisfied by this job starting are started (or stopped respectively).
     - ★ 在/etc/init目录中搜索包含starting的文件
  6. The starting event completes.
  7. The state is changed from ’starting’ to ’pre-start’.
  8. If the pre-start stanza exists, the pre-start process is spawned.
     - ★ 在/etc/init目录中搜索包含pre-start的文件

Upstart中的job及其配置文件 III

9. If the pre-start process fails, the goal is changed from 'start' to 'stop', and the stopping and stopped events are emitted with appropriate variables set denoting the error.
10. Assuming the pre-start did not fail or did not call "stop", the main process is spawned.
11. The state is changed from 'pre-start' to 'spawned'.
12. Upstart then ascertains the final PID for the job which may be a descendent of the immediate child process if expect fork or expect daemon has been specified.
    - ★ 在/etc/init目录中搜索包含expect的文件
13. The state is changed from 'spawned' to 'post-start'.
14. If the post-start stanza exists, the post-start process is spawned.
    - ★ 在/etc/init目录中搜索包含post-start的文件
15. The state is changed from 'post-start' to 'running'.

Upstart中的job及其配置文件 IV

16. The started event is emitted.
    For services, when this event completes the main process will now be fully running. If the job refers to a task, it will now have completed (successfully or otherwise).
17. Any jobs whose 'start on' (or 'stop on') condition would be satisfied by this job being started are started (or stopped respectively).

- Job配置文件中的stanzas（节）
  参见upstart.ubuntu.com/cookbook/#stanzas-by-category
  1. Process Definition
     - ★ exec，pre-start，post-start，pre-stop，post-stop，script
  2. Event Definition
     - ★ manual，start on，stop on
  3. Job Environment
     - ★ env，export

Upstart中的job及其配置文件 V

4. Services, tasks and respawning
   - ★ normal，exit，respawn，respawn limit，task
5. Instances
   - ★ instance
6. Documentation
   - ★ author，description，emits，version，usage
7. Process environment
   - ★ apparmor load ， apparmor switch，cgroup ，console none，console log，console output，console owner，chdir，chroot，limit，nice，oom score，setgid，setuid， umask
8. Process Control
   - ★ expect fork，expect daemon，expect stop，kill signal， kill timeout，reload signal

Upstart中的job及其配置文件 VI

- Ubutun-14.04中，job都在

  /etc/init/

  目录中列出

  - 查看/etc/init/目录
  - ttyX.conf：伪终端数目的，即Ctrl+Alt+F(1~6)调出的Console
    - ★ 以tty1.conf为例，阅读

tty1.conf

- tty1.conf是定义了一个job，名称为tty1

```
# tty1 - getty
#
# This service maintains a getty on tty1 from the point the system is
# started until it is shut down again.

start on stopped rc RUNLEVEL=[2345] and (
            not-container or
            container CONTAINER=lxc or
            container CONTAINER=lxc-libvirt)

stop on runlevel [!2345]

respawn
exec /sbin/getty -8 38400 tty1
```

start on/stop on：表示启动/停止job的事件条件
respawn：每当命令结束,就重启该命令
exec：执行指定的命令

upstart管理的ubuntu启动过程： I

1. 内核启动/sbin/init（即Upstart，1号进程）
2. Upstart触发startup事件
   - 在/etc/init下搜索包含startup的文件，查看相关job
3. 因startup事件而start on的几个job得到运行，其中包括mountall
   - 阅读mountall.conf
4. mountall job触发相关事件，包括virtual-filesystems，local-filesystems，remote-filesystems，all-swaps，filesystem，mounting，mounted
   - 在/etc/init下搜索包含virtual-filesystems的文件
5. 响应virtual-filesystems事件的job中包含udev
6. ...
7. 在最后一个文件系统被mount之后，mountall会发出filesystem事件

upstart管理的ubuntu启动过程： II

- 在/etc/init下搜索包含filesystem的文件

8. rc-sysinit job调用telinit命令，并传递缺省启动级别给它

/etc/init/rc-sysinit.conf，第60行

```
# Switch into the default runlevel
telinit "${DEFAULT_RUNLEVEL}"
```

9. telinit命令触发runlevel事件
   - 在/etc/init下搜索包含runlevel的文件
10. runlevel事件导致其他job被启动，包括/etc/init/rc.conf（用于兼容启动systemV的init系统）
    - 阅读/etc/init/rc.conf文件

mountall.conf I

```
# mountall - Mount filesystems on boot
#
# This helper mounts filesystems in the correct order as the devices
# and mountpoints become available.

description "Mount filesystems on boot"

start on startup
stop on starting rcS

expect daemon
task

emits virtual-filesystems
emits local-filesystems
emits remote-filesystems
emits all-swaps
emits filesystem
emits mounting
emits mounted
```

mountall.conf II

```
script
    . /etc/default/rcS || true
    [ -f /forcefsck ] && force_fsck=" —force-fsck"
    [ " $FSCKFIX" = " yes" ] && fsck_fix=" —fsck-fix"

# Doesn' t work so well if mountall is responsible for mounting /proc, heh.
    if [ -e /proc/cmdline ]; then
        read line < /proc/cmdline
        for arg in $line; do
            case $arg in
                -q|—quiet|-v|—verbose|—debug)
                    debug_arg=$arg
                    ;;
            esac
        done < /proc/cmdline
    fi
    # set $LANG so that messages appearing in plymouth are translated
    if [ -r /etc/default/locale ]; then
         . /etc/default/locale || true
        export LANG LANGUAGE LC_MESSAGES LC_ALL
    fi
```

mountall.conf III

```
    exec mountall --daemon $force_fsck $fsck_fix $debug_arg
end script

post-stop script
    rm -f /forcefsck 2>dev/null || true
end script
```

task：job完成指定工作后，回到waiting状态（即初始状态）
emits：列举会触发的事件

runlevel事件

- 使用

  man 7 runlevel

  可以查看runlevel事件相关内容
- runlevel事件：The runlevel event signals a change of system runlevel.
  - 8个runlevel：0～6以及S（或s）
    其中，0，1，6保留。
    0：halt;
    6：reboot;
    1：用于让系统进入单用户模式，随后系统进入S级别
  - Ubutnu的默认启动级别是2
  - 一种runlevel代表了一个环境，一个环境由运行在这个runlevel中的各种服务以及其他系统组件组成
  - runlevel切换时，根据runlevel定义需要stop或者start一些服务
- 事件的产生：telinit或shutdown

Ubuntu的默认启动级别

- Ubutnu的默认启动级别是2
- 如何修改默认启动级别？
  - 在/etc/init/rc-sysinit.conf中，修改DEFAULT_RUNLEVEL的值

/etc/init/rc-sysinit.conf，第12行

```
# Default runlevel, this may be overriden on the kernel
command-line
# or by faking an old /etc/inittab entry
env DEFAULT_RUNLEVEL=2
```

/etc/init.d/

- 存放服务(services)或者任务(tasks)的执行脚本
  - 观察/etc/init.d目录
  - 阅读文件/etc/init.d/README
- [原来]只要安装了一个程序(特别是服务程序daemon)，它可以在系统启动时运行，那么它必定会在/etc/init.d/中有一个脚本文件
- [现在]用来兼容systemV的init系统
- 看rc脚本
  - rc脚本文件中的关键for循环

/etc/rc?.d/

- 8个运行级别对应8个目录
  1. /etc/rc0.d/
  2. /etc/rc1.d/
  3. /etc/rc2.d/
  4. /etc/rc3.d/
  5. /etc/rc4.d/
  6. /etc/rc5.d/
  7. /etc/rc6.d/
  8. /etc/rcS.d/
- 这些目录下的文件是一些到/etc/init.d/中脚本的符号链接
- 系统缺省的runlevel为2，以/etc/rc2.d/目录为例，阅读README
  - S打头表示在启动时运行
  - 数字表示执行的先后顺序
  - 若见到K打头的文件，表示相关服务应被停止（具体参见README）
- 使用命令

  ls /etc/rc*.d -la

  查看各个rcX.d目录的内容

Outline

Thanks !

The end.