第11章 Linux日常管理和维护

本章内容

- 11.1 进程管理
- 11.2 任务计划
- 11.3 Linux系统启动过程
- 11.4 维护GRUB 2
- 11.5 设置GRUB 2加密
- 11.6 GRUB 2配置案例

11.1 进程管理

■ Linux是一个多任务的操作系统,系统上可以同时运行多个进程,正在执行的一个或多个相关进程称为一个作业。用户可以同时运行多个作业,并在需要时可以在作业之间进行切换。

进程概念

■ 大多数系统都只有一个CPU和一个内存,但一个 系统可能有多个二级存储磁盘和多个输入/输出设 备。操作系统管理这些资源并在多个用户间共享资 源, 当提出一个请求时, 操作系统监控着一个等待 执行的任务队列,这些任务包括用户作业、操作系 统任务、邮件和打印作业等。操作系统根据每个任 务的优先级为每个任务分配合适的时间片,每个时 间片大约都有零点几秒,虽然看起来很短,但实际 上已经足够计算机完成成千上万的命令集。每个任 务都会被系统运行一段时间, 然后挂起, 系统转而 处理其它任务。过一段时间以后再回来处理这个任 务, 直到某个任务完成, 从任务队列中去除。

• Linux系统上所有运行的内容都可以称为进程。每个用户任务、每个系统管理守护进程都可以称为进程。Linux系统用分时管理方法使所有的任务共同分享系统资源。我们讨论进程的时候,不会去关心这些进程究竟是如何分配的,或者内核是如何管理、分配时间片的,我们所关心的是如何去控制这些进程,让它们能够很好地为用户服务。

■ 进程是在自身的虚拟地址空间运行的一个单独的程序。进程与程序之间还是有明显区别的。程序只是一个静态的命令集合,不占系统的运行资源;而进程是一个随时都可能发生变化的、动态的、使用系统运行资源的程序。一个程序可以启动多个进程。和进程相比较,作业是一系列按一定顺序执行的命令。一条简单的命令可能会涉及多个进程,尤其是当使用管道和重定向时。

进程特征

- 动态性: 进程的实质是程序在多道程序系统中的一次执行 过程, 进程是动态产生、动态消亡的。
- 并发性: 任何进程都可以同其它进程一起并发执行。
- 独立性: 进程是一个能独立运行的基本单位,同时也是系统分配资源和调度的独立单位。
- 异步性:由于进程间的相互制约,使得进程具有执行的间断性,即进程按各自独立的、不可预知的速度向前推进。
- 结构特征: 进程由程序、数据和进程控制块三部分组成。
- 多个不同的进程可以包含相同的程序:一个程序在不同的数据集里就构成不同的进程,能得到不同的结果;但是执行过程中,程序不能发生改变。

进程种类

- 交互式进程:一个由Shell启动并控制的进程,交互式进程既可在前台运行,也可在后台运行。
- 批处理进程: 与终端无关,安排在指定时刻完成的一系列进程。
- 守护进程: 在引导系统时启动,以执行即时的操作系统任务,比如crond、rsyslogd、named等。

ps命令

ps命令是最基本同时也是非常强大的进程查看命令。使用该命令可以确定有哪些进程正在运行以及进程运行的状态、进程是否结束、进程有没有僵死,以及哪些进程占用了过多的资源等。

命令语法:

ps [选项]

ps命令输出字段的含义

字段	含义
USER	进程所有者的用户名
PID	进程号,可以唯一标识该进程
%CPU	进程自最近一次刷新以来所占用的CPU时间和总时间的百分比
%MEM	进程使用内存的百分比
VSZ	进程使用的虚拟内存大小,以KB为单位
RSS	进程占用的物理内存的总数量,以KB为单位
TTY	进程相关的终端名
STAT	进程状态,R表示运行或准备运行,S表示睡眠状态,I表示空闲;Z表示冻结,D表示不间断睡眠,W表示进程没有驻留页,T表示停止或跟踪
START	进程开始运行时间
TIME	进程使用的总CPU时间
COMMAND	被执行的命令行

【例11.1】显示所有进程。 [root@rhel ~]# ps -e

【例11.2】显示所有不带控制台终端的进程,并显示用户名和进程的起始时间。 [root@rhel~]# ps -aux

USER	PID %CPU	%MEM	VSZ I	RSS	TTY S	STAT	START	TIME C	COMMAND
root	1 0.0 0.3	5776	3512?		Ss 08	8:27	0:02 /us	r/lib/syst	em
root	2 0.0 0.0	0	0 ?	S	08:27	0:00	(kthread	dd]	
root	3 0.0 0.0	0	0 ?	S	08:27	0:00	(ksoftire	qd/0]	
root	5 0.0 0.0	0	0 ?	S	08:27	0:00) [kworke	er/u:0]	
root	6 0.0 0.0	0	0?	S	08:27	0:00) [migrati	on/0]	
	<u></u>	(省略)						

【例11.3】 查看less进程是否在运行。

[root@rhel \sim]# ps -ax|grep less

6366 pts/1 T 0:00 less

6369 pts/1 T 0:00 less

6374 pts/1 R+ 0:00 grep --color=auto less

【例11.4】显示用户名和进程的起始时间。

[root@rhel ~]# ps -u

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 681 0.0 2.2 56720 22620 tty1 Ss+ 00:26 0:11 /usr/bin/Xorg:

root 890 0.0 0.2 8360 2880 pts/0 Ss 00:26 0:01 -bash

root 3890 0.0 0.1 6592 1072 pts/0 R+ 04:19 0:00 ps -u

【例11.5】显示用户root的进程。 [root@rhel~]# ps -u root

【例11.6】显示tty1终端下的进程。 [root@rhel~]# ps -t tty1 PID TTY TIME CMD 10697 tty1 00:00:01 Xorg

【例11.7】显示进程号为1659的进程。 [root@rhel~]# ps -p 1659 PID TTY TIME CMD 1659? 00:00:01 crond

top命令

• 使用top命令可以显示当前正在运行的进程以及关于它们的重要信息,包括它们的内存和CPU使用量。执行top命令可以显示目前正在系统中执行的进程,并通过它所提供的互动式界面,用热键加以管理。要退出top,按[q] 键即可。

命令语法: top [选项]

【例11.8】 使用top命令动态显示进程信息。

[root@rhel ~]# top

top - 09:27:04 up 59 min, 2 users, load average: 0.01, 0.08, 0.11

Tasks: 140 total, 1 running, 139 sleeping, 0 stopped, 0 zombie

Cpu(s): 0.8%us, 0.8%sy, 0.0%ni, 98.5%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 1027848k total, 917296k used, 110552k free, 212508k buffers

Swap: 2097148k total, 0k used, 2097148k free, 406880k cached

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND 2863 root 20 0 2884 1084 820 R 1.5 0.1 0:00.03 top 1 root 20 0 5776 3512 1940 S 0.0 0.3 0:02.97 systemd 2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd 3 root 20 0 0 0 0 S 0.0 0.0 0:00.01 ksoftirqd/0 5 root 20 0 0 0 0 S 0.0 0.0 0:00.08 kworker/u:0 6 root RT 0 0 0 0 S 0.0 0.0 0:00.00 migration/0 7 root RT 0 0 0 0 S 0.0 0.0 0:00.21 watchdog/0
```

//按"q"键可以退出top

杀死进程

要关闭某个应用程序可以通过杀死其进程的方式实现,如果进程一时无法杀死,可以将其强制杀死。使用kill命令可以杀死进程。在使用kill命令之前,需要得到要被杀死的进程的PID(进程号)。用户可以使用ps命令获得进程的PID,然后用进程的PID作为kill命令的参数。

命令语法:

kill [选项] [进程号]

【例11.11】强制杀死进程号为1659的进程。 [root@rhel~]# ps -ef|grep crond //查询到crond进程号为1659

[root@rhel ~]# kill -9 1659

11.2 任务计划

- 如果要在固定的时间上触发某个作业,就需要创建任务计划,按时执行该作业,在Linux系统中常用cron实现该功能。
- 使用cron实现任务自动化可以通过修改 /etc/crontab文件以及使用crontab命令实现, 其结果是一样的。

/etc/crontab文件详解

- root用户通过修改/etc/crontab文件可以实现任务 计划,而普通用户却无法修改该文件。crond守 护进程可以在无需人工干预的情况下,根据时间 和日期的组合来调度执行重复任务。
- /etc/crontab文件是cron的默认配置文件,前面3 行是用来配置cron任务运行环境的变量。Shell变量的值告诉系统要使用哪个Shell环境(在这个例子里是/bin/bash)。PATH变量定义用来执行命令的路径。cron任务的输出被邮寄给MAILTO变量定义的用户名。如果MAILTO变量被定义为空白字符串,电子邮件就不会被寄出。

• /etc/crontab文件中的每一行都代表一项任务,它的格式如下:

minute hour day month dayofweek user-name commands

/etc/crontab文件内容

内容	描述				
minute	分钟,0~59之间的任何整数				
hour	小时,0~23之间的任何整数				
day	日期,从1~31之间的任何整数(如果指定了月份,必须 是该月份的有效日期)				
month	月份,1~12之间的任何整数(或使用月份的英文简写如jan、feb等)				
day of week	星期,0~7之间的任何整数,这里的0或7代表星期日 (或使用星期的英文简写如sun、mon等)				
user-name	执行命令的用户				
command	要执行的命令或者是自己编写的脚本				

时间格式

时间格式	描述					
*	可以用来代表所有有效的值。如月份值中的星号意味着在满足其它制约条件后每月都执行该命令					
_	指定一个整数范围。比如1-4意味着整数1、2、3、4					
,	指定隔开的一系列值指定一个列表。比如3,4,6,8标明这 4个指定的整数					
	可以用来指定间隔频率。在范围后加上/ <integer>意味着在范围内可以跳过integer。如 "0-59/2"可以用来在分钟字段上定义时间间隔为两分钟。间隔频率值还可以和星号一起使用,如 "*/3"的值可以用在月份字段中表示每3个月运行一次任务</integer>					

/etc/crontab文件配置举例

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
30 21* * * root /root/backup.sh
//在每天晚上的21:30执行/root/backup.sh文件
45 4 1,10,22 * * root /root/backup.sh
//在每月1、10、22日的4:45执行/root/backup.sh文件
20 1 * * 6,0 root /bin/find / -name core -exec rm {} \;
//在每星期六、星期日的1:20执行一个find命令,查找相应的文件
0,30 18-23 * * * root /root/backup.sh
//在每天18:00~23:00之间每隔30分钟执行/root/backup.sh
0 23 * * 6 root /root/backup.sh
//在每星期六的23:00执行/root/backup.sh
```

/etc/cron.d目录

- 除了通过修改/etc/crontab文件实现任务计划之外,还可以在/etc/cron.d目录中创建文件来实现。
- 该目录中的所有文件和/etc/crontab文件使用一样的配置语法。

crontab命令简介

- root以外的用户可以使用crontab命令配置cron任务。所有用户定义的crontab都被保存在/var/spool/cron目录中,并使用创建它们的用户身份来执行。
- 以某位用户身份创建一个crontab项目,登录为该用户,然后输入crontab -e命令,使用由VISUAL或EDITOR环境变量指定的编辑器来编辑该用户的crontab。该文件使用的格式和/etc/crontab相同。当对crontab所做的改变被保存后,该crontab文件会根据该用户名被保存在/var/spool/cron/<username>文件中。
- crond守护进程每分钟都检查/etc/crontab文件、/etc/cron.d目录以及/var/spool/cron目录中的改变。如果发现了改变,它们就会被载入内存。

crontab命令语法

■ 使用crontab命令可以创建、修改、查看以 及删除crontab条目。

命令语法:

crontab [选项]

crontab [选项] [文件]

创建crontab

■ 创建新的crontab,然后提交给crond进程。 同时,新创建crontab的一个副本已经被放 在/var/spool/cron目录中,文件名就是用户 名。 【例11.12】以普通账号zhangsan登录系统,创建用户crontab条目。

[root@rhel \sim]# su - zhangsan

//以普通用户zhangsan登录系统

[zhangsan@rhel \sim]\$ crontab -e

//使用 "crontab -e"命令打开vi编辑器,编辑用户zhangsan的crontab条目

8 * * * * cp /home/zhangsan/aa /home/zhangsan/bb

//在vi编辑器内输入以上crontab条目

[zhangsan@rhel \sim]\$ su - root

口令:

//切换为root用户登录

[root@rhel ~]# cat /var/spool/cron/zhangsan

8 * * * * cp /home/zhangsan/aa /home/zhangsan/bb

//可以看到/var/spool/cron/zhangsan的内容就是刚才用"crontab -e"编辑的内容。记住:普通用户没有权限打开该文件

编辑crontab

- 如果希望添加、删除或编辑 /var/spool/cron/zhangsan文件,可以使用vi编辑 器像编辑其它任何文件那样修改 /var/spool/cron/zhangsan文件并保存退出。如果 修改了某些条目或添加了新的条目,那么在保存 该文件时,crond会对其进行必要的完整性检查。 如果其中的某个地方出现了超出允许范围的值, 它会提示用户。
- 最好在/var/spool/cron/zhangsan文件的每一个条目之上加入一条注释,这样就可以知道它的功能、运行时间,更为重要的是,知道这是哪位用户的作业。

列出crontab

【例11.13】以root用户列出zhangsan的crontab。 [root@rhel~]# crontab -u zhangsan -l

【例11.14】以普通用户zhangsan列出自己的crontab。 [zhangsan@rhel ~]\$ crontab –l

【例11.15】对/var/spool/cron/zhangsan文件做备份。 [zhangsan@rhel ~]\$ crontab -l >/home/zhangsan/zhangsancron

删除crontab

■ 删除crontab时也会删除/var/spool/cron目录中指定用户的文件。

【例11.16】以用户root删除zhangsan的crontab。 [root@rhel~]# crontab –u zhangsan -r

【例11.17】以普通用户zhangsan删除自己的crontab。 [zhangsan@rhel~]\$ crontab -r

恢复丢失的crontab文件

- 如果不小心误删除了crontab文件,且在主目录下还有一个备份,那么可以将其复制到/var/spool/cron/<username>,其中<username>是用户名。
- 如果由于权限问题无法完成复制,可以使用以下命令,其中需要指定在用户主目录中复制的副本文件名。

crontab [文件]

【例11.18】 以zhangsan用户登录恢复丢失的crontab文件。 [zhangsan@rhel ~]\$ crontab -r //删除crontab文件 [zhangsan@rhel ~]\$ crontab -I no crontab for zhangsan [zhangsan@rhel~]\$ crontab /home/zhangsan/zhangsancron //恢复丢失的crontab文件 [zhangsan@rhel ~]\$ crontab -l 8 * * * * cp /home/zhangsan/aa /home/zhangsan/bb //恢复以后可以看到丢失的crontab文件条目

11.3 Linux系统启动过程

- Linux系统的启动是从计算机开机通电自检 开始直到登录系统需要经过的多个过程。
- 1. BIOS自检
- 2. 启动GRUB 2
- 3. 加载内核
- 4. 执行systemd进程
- 5. 初始化系统环境
- 6. 执行/bin/login程序

1. BIOS自检

- 计算机在接通电源之后首先由BIOS进行POST自检,然后依据BIOS内设置的引导顺序从硬盘、软盘或CDROM中读入引导块。
- BIOS的第1个步骤是加电POST自检。POST的工作是对硬件进行检测。BIOS的第2个步骤是进行本地设备的枚举和初始化。BIOS由两部分组成: POST代码和运行时的服务。当POST完成之后,它被从内存中清理出来,但是BIOS运行时服务依然保留在内存中,目标操作系统可以使用这些服务。
- Linux系统都是从硬盘上引导的,其中主引导记录(MBR) 中包含主引导加载程序。MBR是一个512字节大小的扇区,位于磁盘上的第一个扇区(0道0柱面1扇区)。当MBR被加载到RAM中之后,BIOS就会将控制权交给MBR。

2. 启动GRUB 2

■ GRUB 2是Linux系统中默认使用的引导加载程序,引导加载程序是用于引导操作系统启动。当机器引导它的操作系统时,BIOS会读取引导介质上最前面的512字节(主引导记录)。

3. 加载内核

- 接下来的步骤就是加载内核映像到内存中, 内核映像并不是一个可执行的内核,而是 一个压缩过的内核映像。
- Linux内核启动后,会执行下列4项工作:
 - 1. 检测所有硬件设备
 - 2. 驱动硬件设备
 - 3. 以只读的方式挂载根文件系统
 - 4. 启动systemd服务

4. 执行systemd进程

- systemd进程是系统所有进程的起点,内核在完成核内引导以后,即在本进程空间内加载systemd程序。
- systemd进程是所有进程的发起者和控制者。 因为在任何基于Linux的系统中,它都是第 一个运行的进程,所以systemd进程的编号 (PID)永远是1。

■ Systemd进程有以下两个作用:

- 1. 扮演终结父进程的角色。因为systemd进程永远不会被终止,所以系统总是可以确信它的存在,并在必要的时候以它为参照。如果某个进程在它衍生出来的全部子进程结束之前被终止,就会出现必须以systemd为参照的情况。此时那些失去了父进程的子进程就都会以systemd作为它们的父进程。
- 2. 在进入某个特定的服务启动集合,即是 /etc/systemd/system/default.target, 它的这个作用是由 运行目标target定义的。

5. 初始化系统环境

- Linux系统使用systemd作为引导管理程序, 之后的引导过程将由systemd完成。
- Systemd使用目标(target)来处理引导和服务管理过程。这些systemd里的目标文件被用于分组不同的引导单元以及启动同步进程。

- 1. systemd执行的第一个目标是default.target,但实际上default.target是指向graphical.target的软链接,graphical.target的实际位置是/usr/lib/systemd/system/graphical.target。
- 2. 在执行graphical.target阶段,会启动multi-user.target,而这个target将自己的子单元放在/etc/systemd/system/multi-user.target.wants目录中,这个target为多用户支持设定系统环境,非root用户会在这个阶段的引导过程中启用,防火墙相关的服务也会在这个阶段启动。
- 3. multi-user.target会将控制权交给basic.target。basic.target用于启动普通服务特别是图形管理服务,它通过/etc/systemd/system/basic.target.wants目录来决定哪些服务会被启动,basic.target之后将控制权交给sysinit.target。
- 4. sysinit.target会启动重要的系统服务,例如系统挂载、内存交换空间和设备、内核补充选项等,sysinit.target在启动过程中会传递给localfs.target和swap.target。
- 5. local-fs.target和swap.target不会启动用户相关的服务,它只处理底层核心服务,这两个target会根据/etc/fstab和/etc/inittab来执行相关操作。

6. 执行/bin/login程序

■ login程序会提示使用者输入账号及密码,接着编码并确认密码的正确性,如果账号与密码相符,则为使用者初始化环境,并将控制权交给shell,即等待用户登录。

11.4 维护GRUB 2

■ 使用引导加载程序可以引导操作系统的启动,一般情况下引导加载程序都安装在MBR (主引导扇区)中,本节主要讲述Linux系统中GRUB2的维护。

GRUB的概念

- 当计算机要引导操作系统时,BIOS会读取引导介质上最前面的主引导记录(MBR记录)。主引导记录本身要包含两类内容:引导加载程序和分区表。
- GRUB是Linux系统默认的引导加载程序。在Linux加载一个系统前,它必须由一个引导加载程序中的特定指令去引导系统。这个程序一般是位于系统的主硬盘驱动器或其他介质驱动器上。
- Linux安装程序允许用户快速、方便地配置引导加载程序, 将其存放在主硬盘驱动的主引导记录中来引导操作系统。

- GNU GRUB是一个将引导加载程序安装到主引导记录的程序,主引导记录是位于一个硬盘开始的扇区。它允许位于主引导记录区中特定的指令来加载一个GRUB菜单或是GRUB的命令环境。这使得用户能够开始操作系统的选择,在内核引导时传递特定指令给内核,或是在内核引导前确定一些系统参数(如可用的RAM大小)。
- GRUB支持直接和链式加载的引导方法。GRUB能用于几乎所有操作系统、绝大多数流行的文件系统以及几乎所有的系统BIOS所能识别的硬盘。

GRUB 2新功能

- GRUB 2采用模块化动态加载的思想,相比GRUB来讲不用在构建时将所有功能都加入,这使得GRUB 2的体积相比变得很小,整个GRUB 2的内核映像可以控制在31KB以内(GRUB的映像在百KB级别),因此GRUB 2完全可以移除GRUB中存在的stage 1.5阶段,而可以将整个映像放在GRUB中的stage1.5存放的位置。
 - (1) 图形接口。
 - (2) 使用模块机制,通过动态加载需要的模块来扩展功能。
 - (3) 支持脚本语言,比如条件判断、循环、变量和函数。
 - (4) 支持救援模式,可以用于系统无法引导的情况。

- (5) 国际化语言。包括支持非ASCII的字符集和类似gettext的消息分类、字体、图形控制台等。
- (6) 有一个灵活的命令行接口。如果不存在配置文件, GRUB 2会自动进入命令模式。
- (7)针对文件系统、文件、设备、驱动、终端、命令、分 区表、系统加载的模块化、层次化、基于对象的框架。
 - (8) 支持多种文件系统格式。
 - (9) 可以访问已经安装在设备上的数据。
 - (10) 支持自动解压。

GRUB 2中设备和分区命名规则

- GRUB 2同样以fd表示软盘,hd表示硬盘(包含IDE和SCSI硬盘)。 设备是从0开始编号,分区则是从1开始,主分区从1~4,逻辑驱动器 从5开始。
- 下面讲解一下设备和分区的使用方法。

(fd0):表示第一个软盘。

(hd0):表示第一个硬盘(大多数U盘与USB接口的移动硬盘以及SD卡也都被当作硬盘看待)。

(hd0,msdos1):表示第一个硬盘的第一个分区。

(hd0,msdos1)/boot/vmlinuz-3.10.0-327.el7.x86_64: 表示第一个硬盘的第一个分区下的boot目录下的vmlinuz-3.10.0-327.el7.x86_64文件。

(hd1,msdos1):表示第二硬盘的第一分区。

(cd): 启动光盘(仅在从光盘启动GRUB时可用)。

(cd0): 第一个光盘。

GRUB 2主配置文件

- GRUB 2的配置是通过以下三个地方的文件来完成的。
 - 1. /boot/grub2/grub.cfg文件(/etc/grub2.cfg文件是/boot/grub2/grub.cfg文件的软链接)。
 - 2. /etc/grub.d目录。
 - 3. /etc/default/grub文件(/etc/sysconfig/grub文件是/etc/default/grub文件的软链接)。

- 他们的关系是/boot/grub2/grub.cfg文件里面通过 "###BEGIN ####"这种格式按照顺序调用/etc/grub.d目录中的脚本实现不同的功能。/etc/grub.d目录中有很多数字开头的脚本,按照从小到大的顺序执行。以00_header为例,他又会调用/etc/default/grub 配置文件来实现最基本的开机界面配置。
- 比如在/boot/grub2/grub.cfg文件里面调用 /etc/grub.d/10_linux 来配置不同的内核,这里面有2 个 menuentry(菜单项),所以开机的时候会看见两个默认选项,一个是普通模式,一个是救援模式。
- 需要注意的是,最好不要直接修改/boot/grub2/grub.cfg文件,因为后期升级内核,所有的配置都会失效。如果需要自定义这个文件,可以修改/etc/grub.d目录中对应的脚本或者/etc/default/grub文件,然后通过使用grub2-mkconfig命令重新生成/boot/grub2/grub.cfg文件。

/etc/grub.d目录

- 定义每个菜单项的所有脚本都存放在/etc/grub.d目录中,这些脚本的名称必须有两位的数字前缀,其目的是在构建GRUB 2菜单时定义脚本的执行顺序以及相应菜单项的顺序,比如00_header文件首先被读取。
- 在使用grub2-mkconfig命令生成配置文件时需要加载/etc/grub.d目录。

文件	描述
00_header	设置grub默认参数
10_linux	系统中存在的多个Linux 版本
20_ppc_terminfo	设置tty控制台
30_os_prober	设置其他分区 <mark>中的系统(硬</mark> 盘中有多个 操作系统时设置)
40_custom和41_custom	用户自定义的配置

/etc/default/grub文件详解

■ /etc/default/grub文件是一个文本文件,可以在该文件中设置通用配置变量和GRUB 2 菜单的其它特性。在更改/etc/default/grub文件后,需要使用grub2-mkconfig命令更新GRUB 2配置文件才能使更改生效。

/etc/default/grub文件参数

GRUB_TIMEOUT=5

设置进入默认启动项的等候时间,默认值5秒。可以设置为-1,这样就无限等待。

GRUB_DISTRIBUTOR="\$(sed 's, release .*\$,,g' /etc/system-release)"

由GRUB的发布者设置他们的标识名。这用于产生更具信息量的菜单项名称。

GRUB_DEFAULT=saved

设置默认启动项。比如要默认从第1个菜单项启动,设置为0。要默认从第2个菜单项启动,设置为1。如果设置为saved,则默认为上次启动项。

GRUB_TERMINAL_OUTPUT="console"

选择终端输出设备。在这里可能可以选择多个设备,以空格分开。有效的终端输出名依赖于平台,不过可能包括console(PC BIOS及EFI控制台)、serial(串行终端)、gfxterm(图形模式输出)、ofconsole(开放固件控制台)或vga_text(VGA文本输出,主要用在Coreboot)。

GRUB_CMDLINE_LINUX="crashkernel=auto rhgb quiet" 手动添加内核启动参数。

GRUB_DISABLE_RECOVERY="true"

设置是否启用修复模式。

GRUB_DISABLE_LINUX_RECOVERY="true"

设置是否创建修复模式菜单项。

GRUB_HIDDEN_TIMEOUT=0

不显示菜单,但会显示空白界面,设置时间内按任意键出现菜单。

GRUB_HIDDEN_TIMEOUT_QUIET=true

显示空白界面时是否不出现倒计时。

GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"

添加默认内核启动参数。

GRUB_TERMINAL=console

设置使用终端还是图形界面来引导。

GRUB_GFXMODE=640x480

设置图形界面分辨率。只能使用显卡支持的分辨率,具体可以在grub模式下输入vbeinfo查看。

GRUB_DISABLE_LINUX_UUID=true

设置是否使用UUID引导,即使用root=/dev/sda×还是root=UUDI=xxx。

GRUB_INIT_TUNE="480 440 1"

在GRUB启动时,调整扬声器的音量。

11.5 设置GRUB 2加密

■由于GRUB 2负责引导Linux系统,作为系统中的第一道屏障的安全性非常重要,对GRUB 2进行加密可以实现安全性。

GRUB 2加密简介

■ 在默认情况下,GRUB 2对于所有可以在物 理上进入控制台的人都是可访问的。任何 人都可以选择并编辑任意菜单项,并且可 以直接访问GRUB命令行。要启用认证支 持,必须将环境变量superusers设置为一 组用户名(可以使用空格、逗号、分号作 为分隔符),这样将只允许superusers (超级用户) 中的用户使用GRUB命令行、 编辑菜单项、以及执行任意菜单项。

GRUB 2密码支持格式

- 明文密码:密码数据没有经过加密,安全性差;
- PBKDF2加密密码:密码经过PBKDF2哈希算法进行加密,在文件中存储的是加密后的密码数据,安全性较高。

设置GRUB 2 PBKDF2加密口令

■ 先要使用grub2-mkpasswd-pbkdf2命令PBKDF2加密口令,然后在/etc/grub.d/00_header文件中添加超级用户和PBKDF2加密口令,最后使用grub2-mkconfig命令生成GRUB 2配置文件。

设置GRUB 2明文密码

- 按以下步骤在GRUB 2中使用明文密码保护 GRUB 2。
 - 1. 修改/etc/grub.d/00_header 文件
 - 2. 生成GRUB 2配置文件

GRUB 2解锁

■ 设置完GRUB 2密码以后重启Linux系统,在GRUB 2启动菜单界面编辑启动菜单项,此时需要输入用户名和密码解锁之后才能进行编辑。输入用户名和密码界面如下图:

Enter username: zhangsan Enter password: -

11.6 GRUB 2配置案例

- 破解root用户密码:
- 1、开机按光标上下键停止启动,在GRUB2菜单处按<e>进入编辑模式
- 2、在 linux16 行,按<End>键定位到这行末尾,添加 rd.break console=tty0
- 3、按<Ctrl-x>继续启动,然后输入下面命令行:
 - # mount -o remount,rw /sysroot/
 - # chroot /sysroot/
 - # passwd root # 或者 echo redhat123 | passwd --stdin root
 - # touch /.autorelabel
 - # sync
 - # exit # 两次 或者 ctrl + d
- 4、重新启动后,使用指定的密码登录系统,从而验证密码是否正确。

- 将网卡名称eno16777736更改为eth0
- 1. 修改/etc/default/grub文件 在GRUB_CMDLINE_LINUX变量中添加"net.ifnames=0 biosdevname=0"
- 2. 生成GRUB 2配置文件,执行下面命令: grub2-mkconfig -o /boot/grub2/grub.cfg
- 3. 查看网卡名称 重新启动Linux系统,使用ifconfig查看网卡名称。