

Capítulo 4 - Inicialização do sistema Linux

4.1. Objetivos

Entender como funciona o processo de inicialização do sistema Linux e as diferenças entre as distribuições Debian e Red Hat.

Em qual diretório ficam os arquivos de inicialização do Linux?

Esses arquivos de inicialização ficam todos no /etc.

Os arquivos de inicialização que irei comentar são:

- /etc/inittab
- /etc/fstab

4.2. Processo de inicialização do Linux

A inicialização do sistema começa com um boot loader (Grub ou Lilo) no qual o usuário escolhe qual sistema operacional será inicializado.

O que acontece no Linux o selecionamos no boot loader?

O boot loader inicia o carregamento do kernel na memória RAM e passa o controle do sistema para ele. Uma vez que o kernel já esteja controlando a máquina é iniciada a fase de subir os serviços necessários para a utilização do sistema. Este último estágio é o que trataremos agora.

Existem dois estilos de init: o BSD e o System V. Em um sistema padrão System V, haverá níveis de inicialização (runlevels), sendo eles:

- Nível 0:** desliga o sistema;
- Nível 1 ou S ou s:** carrega os serviços mínimos essenciais ao sistema. Nesse modo, os serviços de rede tais como servidores web e compartilhamento de arquivos não são iniciados. É usado para a manutenção do sistema;
- Níveis 2 a 5:** multiusuário;
- Nível 6:** reinicia o sistema;

Quando um sistema Linux se inicializa, o processo init é responsável por levantar o sistema ao nível de execução padrão, que geralmente é 2,3 ou 5, dependendo da distribuição e o do uso previsto para a máquina.

4.3. Arquivo /etc/inittab

E em qual arquivo eu posso definir o runlevel padrão do meu sistema?

O arquivo em que eu defino o runlevel padrão na inicialização é o /etc/inittab

Fazendo uma comparação com o Windows, runlevel seria escolher em que nível quero que o meu sistema inicie.

Seriam aquelas opções: Modo de Segurança, Modo Normal e etc.

Mas muito mais complexo e completo falando em GNU/Linux. Então, no arquivo /etc/inittab configuro em que nível meu sistema vai iniciar, isto é, aqui eu falo se meu sistema vai levantar em GNU/Linux single (modo administrativo), parte gráfica, se vai ser só modo texto etc. A principal função desse arquivo é definir o que chamamos de runlevel, não que ele só faça isso, veremos outras funções dele daqui a pouco.



Dica de Segurança

A distribuição Slackware possui um sistema de inicialização diferente do Debian e Red Hat (System V), seu estilo de inicialização é o BSD (não é cobrado na LPI).

A distribuição Ubuntu não possui /etc/inittab. O Ubuntu usa o upstart e utiliza um arquivo para cada item, antes contido no inittab. Os arquivos ficam dentro do diretório /etc/event.d.

Diferenças entre Red Hat e Debian quanto as funções dos números do runlevel:



Debian:

0- Halt (Desliga a máquina)

1- Single, ou seja monousuário - é o root e mais ninguém aqui!

2 - 5 - O Debian do nível 2 ao 5 trabalha com multiusuário e não faz distinção de modo texto ou gráfico. Você tem que definir o que realmente quer que ele inicie. O runlevel padrão do Debian é 2.

6- Reboot (Reinicia o sistema)



Red Hat:

- 0- Halt (Desliga a máquina)*
- 1- Single, ou seja monousuário - é o root e mais ninguém nesse nível*
- 2- Multiusuário sem o compartilhamento de arquivos NFS*
- 3- Multiusuário com rede e modo texto*
- 4- Não usado, porém pode ser personalizado pelo o usuário*
- 5- Multiusuário completo com parte gráfica (padrão)*
- 6- Reboot (Reinicia o sistema)*

No arquivo `/etc/inittab` eu preciso definir qual vai ser meu runlevel padrão.



```
# vi /etc/inittab
```

Linha no arquivo responsável por definir o runlevel padrão:



```
id: X :initdefault:
```

No qual X é o número do runlevel definido.

Para descobrir o runlevel padrão, digite na linha de comando:



```
# runlevel
```

Posso colocar os números 0 e 6 como padrão?

Não. Se o 0 for colocado como padrão, toda vez que o sistema foi ligado ele será desligado automaticamente de modo infinito até que você vai lá e arrume. O 6 é a mesma coisa, o sistema ligaria e reiniciaria.

Então, quando um Linux inicia, são usados vários scripts para configurar inicialmente o sistema. As técnicas de inicialização do sistema diferem conforme a distribuição.

4.4. Diretório /etc/init.d

No Linux, **existe um diretório chamado /etc/init.d** que guarda os **scripts responsáveis pelos serviços (também chamados de daemons)**. Ou seja, contém scripts individuais para cada serviço do sistema. Por exemplo, o script /etc/init.d/gdm é um script em Shell que inicia ou interrompe o gerenciador de login gráfico na máquina. Esses scripts possuem uma forma padronizada para iniciar ou parar o serviço, exemplo:

Para derrubar o modo gráfico no Debian:



```
# /etc/init.d/gdm stop  
ou  
# invoke-rc.d gdm stop
```

Para levantar o modo gráfico no Debian:



```
# /etc/init.d/gdm start  
ou  
# invoke-rc.d gdm start
```

E o que seria xdm, gdm e kdm???

Eles são gerenciadores de parte gráfica, ou melhor, as telinhas para fazer o login em modo gráfico!

Cada um de um cliente gráfico:

- kdm - Login KDE
- gdm - Login GNOME
- xdm - Padrão do GNU/Linux

No Red Hat, você pode iniciar e parar serviços com o comando service.

Sintaxe:



```
# service daemon stop/start
```

Exemplo, parando e iniciando o servidor web apache:



```
# service httpd stop  
# service httpd start
```

Arquivo importante em `/etc/init.d` que é responsável por tarefas essenciais para preparar o sistema para uso, como por exemplo, montar os sistemas de arquivos:

Debian:



```
/etc/init.d/rcS
```

Red Hat:



```
/etc/rc.sysinit
```

Além do `/etc/init.d`, temos os seguintes diretórios que possuem links apontando para os scripts em `/etc/init.d`. Cada diretório é responsável por um runlevel:

Diretório que guarda links para scripts essenciais de inicialização:

```
/etc/rcS.d
```

Diretório que guarda links para scripts do runlevel 0:

```
/etc/rc0.d
```

Diretório que guarda links para scripts do runlevel 1:

```
/etc/rc1.d
```

Diretório que guarda links para scripts do runlevel 2:

```
/etc/rc2.d
```

Diretório que guarda links para scripts do runlevel 3:

```
/etc/rc3.d
```

Diretório que guarda links para scripts do runlevel 4:

```
/etc/rc4.d
```

Diretório que guarda links para scripts do runlevel 5:

```
/etc/rc5.d
```

Diretório que guarda links para scripts do runlevel 6:

```
/etc/rc6.d
```

Então, se não quero iniciar a parte gráfica mais tenho um dos gerenciadores de cliente gráfico instalado, tenho que tirar o serviço no diretório do runlevel padrão.

Bom sabemos que o nível padrão do Debian é o 2, então ele irá se basear em qual diretório?

```
/etc/rc2.d
```

Dando um **ls** nesse diretório:



```
#ls /etc/rc2.d  
S10sysklogd S19nis S20nfs-kernel-server S23ntp-server S99rmnologin  
S10syslog-ng S20acct S20ntop S25mdadm S99stop-bootlogd  
S11klogd S20gpm S20samba S50proftpd  
S15bind9 S20inetd S20ssh S89cron  
S18portmap S20makedev S21gdm S91apache2
```

Posso ver todos os arquivos que são responsáveis para iniciar os serviços naquele determinado level (nível). Esses arquivos na verdade são links que apontam para os verdadeiros scripts de serviço que ficam em /etc/init.d.

Onde, temos dois tipos de arquivos:

SXXnome e KXXnome

Que significam:

- XX** - Número que especifica a ordem de iniciar ou derrubar
- S** - Iniciar o serviço
- K** - Não iniciar

Então, por exemplo meu nível no inittab está setado como 2.

Então tudo que tiver no /etc/rc2.d será executado na hora que o sistema iniciar?

Sim, porém tudo que tiver S na frente! S = start

Então, por exemplo, se eu não quero que algum serviço (programa) inicialize na hora do boot, pois não estou usando. O que tenho que fazer é trocar o S pelo K

4.5. Prática dirigida

Exemplo:



```
#ls /etc/rc2.d  
S21gdm
```

Ou seja, minha parte gráfica (gerenciador) está iniciando no level 2, mas não quero, pois não estou usando, e quero usar apenas o modo texto.

Então faço o seguinte:



```
#mv S21gdm K21gdm
```

Assim ele passa de S para K (kill). No meu caso estou considerando que estou usando o Gnome com o gerenciador de login gráfico gdm.

Um outro exemplo prático, se o seu sistema deu algum problema, você terá que entrar no modo 1 pois lá não levanta nenhum serviço e você poderá arrumar o sistema:



```
# init 1
```

Mas falando no nível 1 , ele é como se fosse o modo segurança, apenas para manutenção, então pense em mexer apenas de 2-5. O que tem que ficar claro é que você pode personalizar os níveis e usar um de acordo com a sua necessidade.

A necessidade de se ter 7 níveis é apenas personalizar meu sistema. Para complementar essa questão de serviços, eu gostaria de mostrar outras maneiras de tirar/colocar um serviço em um determinado runlevel e como iniciar e parar um serviço tanto no Red Hat quanto no Debian.

4.6. Customizando os runlevels

Toda vez que você adiciona um serviço ao seu sistema, um script precisa ser colocado em /etc/init.d e um link dele precisa ser criado nos diretórios de runlevels corretos.



No Red Hat, a ferramenta usada é o `chkconfig`;



No Debian, a ferramenta usada é o `update-rc.d`;

4.6.1. Comando `chkconfig`:

Para listar em quais diretórios o serviço está:



```
# chkconfig --list serviço
```

Exemplo:



```
# chkconfig --list httpd  
httpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

O `chkconfig` nos mostrou que o serviço `httpd` está disponível somente nos runlevels 2,3,4 e 5. Para adicionar um novo arquivo de serviço ao diretório /etc/init.d e inicializar todos os runlevels certos com os links para o script de serviço rc:



```
# chkconfig --add serviço
```

Exemplo:



```
# chkconfig --add httpd
```

Para desabilitar a inicialização de um serviço durante o boot:



```
# chkconfig -del serviço
```

ou



```
# chkconfig serviço off
```

O chkconfig também pode ser usado para configurar um serviço para ser iniciado (ou não) em um nível de execução específico. Por exemplo: para desligar o httpd nos níveis de execução 3, 4 e 5, use o seguinte comando:



```
# chkconfig --level 345 httpd off
```

Sintaxe:



```
# chkconfig -level níveis serviço on/off/reset
```

O update-rc.d faz a mesma coisa que o chkconfig.



O que o chkconfig faz mesmo?

Cria as ligações entre os arquivos nos diretórios rc?.d e o script em /etc/init.d/.

4.6.2. Comando update-rc.d:

Para remover todos os links de um script, por exemplo faz-se:



```
# update-rc.d -f serviço remove
```

Suponha que acabou de colocar um script chamado firewall.sh no /etc/init.d e é preciso que ele seja iniciado no runlevel padrão (2) e fechado quando se desliga ou reinicia (runlevels 0 e 6) e quer que ele seja um dos últimos a ser iniciado/parado.

Para isso:



```
# update-rc.d firewall.sh start 97 2 . stop 97 0 . stop 97 6 .
```

Como se pode ver, as configurações são separadas por um ponto (.) e há um também no final da linha de comando. Para facilitar as coisas, pode-se usar o update-rc.d com as configurações padrão. Isto é: iniciar nos runlevels 2, 3, 4 e 5 e finalizar nos 0, 1 e

6. Para isso:



```
# update-rc.d serviço defaults
```

Complementando as outras funções do /etc/inittab.



Dica de Segurança: Então, nesse exato momento se apertamos CTRL+ALT+DEL. Nosso sistema sem pensar duas vezes vai reiniciar. Famosa solução Windows! Mas isso não é muito bom deixar habilitado. Afinal qualquer pessoa pode chegar perto e reiniciar o servidor! Então, por uma questão de segurança, é melhor deixarmos isso desabilitado!

E a linha que fala que o sistema deve reiniciar com essa sequência de teclas é:



```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

Temos duas maneiras de desabilitar, e aí vai de gosto, uma é apenas colocando o # na frente, onde estamos comentando a linha, então ele não vai ler essa linha! E a outra seria colocar algo no lugar:



```
ca:12345:ctrlaltdel: /sbin/shutdown -t1 -a -r now
```

Onde o que está em azul é o comando em si, e temos que trocar por outra coisa!



```
ca:12345:ctrlaltdel:/bin/echo "Você apertou CTRL + ALT + DEL"
```

Assim toda vez que a sequência for pressionada, aparecerá essa mensagem na tela!

Para atualizar o arquivo `/etc/inittab` sem reiniciar o sistema, basta executar o comando:



```
#init q
```

4.7. Arquivo /etc/fstab

O que faz o /etc/fstab?

O arquivo /etc/fstab é o arquivo responsável por armazenar nossa tabela de partições. Podemos dizer que ele é o responsável por montar nossos dispositivos de bloco (HD, CDROM, FLOOPY) durante a inicialização do sistema. Ou seja, todos os pedaços (partições) que foram criados instalação serão definidos nesse arquivo! Quando, na instalação definimos as partições que seriam criadas, elas foram armazenadas no /etc/fstab , para que o sistema saiba qual partição montar no hora do boot. Como por exemplo:

/dev/hda1 é a partição do meu diretório /boot. Ele será montado na hora que meu Linux inicializar. Mas qual é o sistema de arquivos dele? Como ele será montado? Vamos fazer backup? Vamos passar o fsck que é um aplicativo para checagem da integridade dos arquivos?

Tudo isso é definido nesse importante arquivo de inicialização. Então, seguindo a estrutura do arquivo temos:

Origem	Destino	Sistema de Arquivo	Opções de Montagem	dump	pass
/dev/hda5	/home	ext3	defaults	0	2

Origem é a partição.

Destino é o diretório que esse dispositivo vai ser montado. Sistema de Arquivos indica qual o sistema de arquivos será usado, no caso o do GNU/Linux pode ser ext2, ext3 e reiserfs. Para pendrives costuma ser vfat.

As opções de montagem são:



auto - se o mesmo será montado no boot

noauto - para não ser montado

user - para um usuário mortal montar

nouser - apenas o root pode montar

rw - para poder gravar nessa partição

ro - para não gravar

noexec - é para não permitir que binários (executáveis) sejam executados na partição.

async - estabelece E/S assíncrona para o sistema de arquivos montado, é o oposto de sync

dev - Interpreta dispositivos especiais de caracteres ou blocos no sistema de arquivos

nosuid - Desabilita o efeito de bits suid ou sgid em arquivos executáveis (ainda veremos isso na aula de Permissões, guarde essa dúvida)



Onde a opção defaults engloba: nouser,exec,auto,rw,suid,dev

Na coluna do pass usa-se 0 para não ser checado, 1 para checar a partição / e 2 para as demais partições que deseja-se checar.

Na coluna dump, é a mesma lógica mas é 0 para não executar o dump e 1 para executar. O dump é um aplicativo de backup a nível de partição. Aqui pode ser determinado que antes de montar essa partição, é necessário realizar um backup da mesma. Isso demora um pouco o processo do boot, habilitando a opção dump!



Dica: Para gravar em partições com sistema de arquivos ntfs é necessário o uso do ntfs-3g (pesquise na internet, tem muito material bom sobre isso, ou coloque como tópico do fórum pessoal). Lembrem-se que o curso também ocorre fora do chat.