I − Por que Administrar ?

É responsabilidade dos administradores: a segurança, a estabilidade e a performance do servidor.

- Segurança

Um dos fatores que os administradores têm que se preocupar é com a segurança de seu sistema, dando um enfoque especial a detecção de acessos não autorizados.

- Estabilidade

Outra tarefa muito importante para os administradores é manter no ar os serviços oferecidos em seu servidor. O bom administrador busca uma estabilidade "constante".

- Performance

E é claro, deve−se oferecer os serviços da maneira mais rápida possível, atendendo, na medida do possível, a necessidade dos usuários.

Um sistema que possuir essas três características, sem dúvida apresentará uma grande ***confiabilidade***.

II − Gerenciando Usuários

Para adicionar um usuário:

```
adduser <usuário>
```

Em seguida é necessário definir uma senha para este usuário, utilizando o comando:

```
passwd <usuário>
```

Nunca se deve ter usuários desnecessários cadastrados. No caso de um usuário para teste, deve−se removê−lo logo após o teste.
Para remover um usuário e seu diretório home:

```
userdel −r usuário
```

Uma das regras de segurança mais importantes é: "evitar usar o root, fazendo−o somente quando necessário. ". Com certeza muitos problemas são evitados seguindo essa simples regra.

Para obter as permissões de outros usuários:

```
su <usuário>
```

Se o campo *<usuário>* for omitido, o *su* intenderá como *root*. E para simular um login, executando os scripts de inicialização do usuário acrescenta−se "**−**" entre *su* e o *<usuário>*, como no exemplo:

```
su − vinicius
```

III − Entendendo o Diretório /etc

O diretório */etc* contém todas as configurações do servidor, por isso deve−se conhecer todo o seu conteúdo e também ter uma preocupação especial com as permissões de arquivos nele contidos.

Alguns arquivos importantes:

- **passwd**

É neste arquivo que ficam os usuários cadastrados no sistema. Cada linha corresponde a um usuário e o caracter ":" separa os campos . Analisando o exemplo abaixo:

```
vinic:x:1001:0:Vinicius Schmidt,,,:/home/vinic:/bin/bash
```

Login : *Senha* : *Id* : *Gid* : *Nome e Dados* : *Diretório* : *Shell*

Login: ***vinic***
É a identificação do usuário que também pode ser usado para identificação do email.

Senha: ***x***
Esse "*x*" informa que a senha está em outro arquivo mais seguro.

Id: ***1001***
É o código único para o Linux identificar cada usuário. Nunca deve−se ter dois usuários com o mesmo código.

Gid: ***0***
É o código do grupo primário que o usuário pertence.

Nome e Dados: ***Vinicius Schmidt,,,***
Este campo é usado para armazenar informações sobre o usuário como nome, telefone, sala, etc.. Esses dados são separados por "," e devem obedecer um padrão.

Diretório: ***/home/vinic***
Este campo informa qual é o directório home, do usuário.

Shell: ***/bin/bash***
É a shell default do usuário. Para usuários que não precisam de shell deve−se colocar *"/dev/null".*

- **shadow**

Este é o arquivo mais visado pelos "*intrusos*", pois é nele que ficam gravadas todas as senhas de acesso ao sistema.
A senha fica necessariamente criptografada dificultando a sua descoberta por pessoas não autorizadas, porem para quem sabe como quebrar essa criptografia não é difícil fazê−lo. Por isso a necessidade de senhas complexas.

Abaixo uma linha do arquivo *shadow,* e a senha criptografada em destaque:

```
vinic:$1$04Jd3syi$iadsszEgE53un1Muk.:11046:0:99999:7:::
```

- **group**

Este arquivo descreve quais são os grupos do sistema. O primeiro campo corresponde ao nome do grupo, o segundo campo a senha (caso seja necessária), o terceiro é o chamado *GID* ou *Group ID*, e o próximo campo são os usuários pertencentes a este grupo, observando que os mesmos são separados por "**,**" .

Abaixo um pequeno exemplo desse arquivo:

```
root::0:root
bin::1:root,bin,daemon
```

- **inittab**

Este é o arquivo principal de inicialização do Linux, ele é quem dá inicio aos demais arquivos dentro do diretório */etc/rc.d* . O modo mais usado é o "3" que indica para iniciar todas as tarefas, mas continuar no console. Existe também o modo "5" que entra com a tela de login já em modo gráfico. Veja detalhes no *Apêndice E,* Níveis de Execução no Red Hat Linux.

- **fstab**

Estabelece os pontos default de montagem do sistema.

Partição	Ponto de Montagem	Tipo	Opções		
/dev/hda2	swap	swap	defaults	0	0
/dev/hda3	/	ext2	defaults	1	1
/dev/hdd1	/mnt/hd	vfat	defaults	0	0
none	/proc	proc	defaults	0	0

- **login.defs**

É o arquivo de configurações do login, possui informações valiosas para melhorar a segurança do seu ambiente. É um arquivo muito simples de configurar pois é baseado em exemplos, e seus parâmetros são simples.

- **profile**

Toda vez que um usuário loga, este arquivo é executado, por isso ele é usado para setar as variáveis de ambiente globais, dentre outras coisas.

- **hosts.deny**

Hosts que não tem permissão para acessar a máquina.

```
in.telnetd: 143.106.20.11
```

- **hosts.allow**

Hosts que tem permissão para acessar a máquina.

```
wu.ftpd: 200.240.2.183
```

Tanto nos arquivos *hosts.deny* quanto *hosts.allow* pode−se usar variáveis coringa:

ALL = Todos os IPs
LOCAL = Todos os IPs locais
REMOTE = Todos os IPs remotos

- **services**

Este arquivo descreve a relação entre os serviços e as portas mais comuns.

Descrição Porta / Protocolo

```
smtp        25/tcp    mail
www         80/tcp    http  # WorldWideWeb HTTP
```

Diretórios mais importantes:

- **/etc/rc.d**

Neste diretório estão os arquivos responsáveis pela carga dos daemons na inicialização do Linux.

- **/etc/skel**

Todos os arquivos contidos neste diretório serão copiados para o home de um usuário recém criado.

IV − Trabalhando com o LILO

O LILO (Linux Loader), como o próprio nome já diz é o responsável pela "carga" do kernel do Linux na memória. O arquivo de configuração responsável pelas configurações do LILO fica no diretório /etc e é chamado de lilo.conf. Abaixo um exemplo bem simples do conteúdo desse arquivo.

```
boot = /dev/hda
delay = 40
compact
vga = normal
root = /dev/hda1
read−only
image = /zImage−2.2.14
        label = stable

image = /zImage−2.3.99
        label = test


other = /dev/hda3
     label = dos
     table = /dev/hda
```

Para alterar o LILO, basta editar o arquivo e rodar o programa **`lilo.`**

V − Fazendo Backups

Realizar backups do sistema hoje em dia é uma tarefa essencial de todo administrador. No Linux pode−se usar o comando ***tar*** para compactar os arquivos.

Sintaxe: *tar [opções] [−f arquivo]*
Opções: x : descompactar.
t : lista o conteúdo de um arquivo.
v : mostra na tela o que está sendo feito.
z : descompacta arquivos que também estejam *gzipados*.
f : especifica o nome do arquivo.
c : cria um novo arquivo.
Exemplo:

`tar cvzf meu_backup.tgz ~` *(faz o backup de sua área pessoal)*
`tar cvzf backup.tgz /home` *(faz o backup da área dos usuários)*
`tar xvzf /root/backup.tar.gz` *(descompacta o backup)*

VI − Recompilando e Adaptando o Kernel

Os administradores devem se preocupar em manter a máquina mais enxuta possível e para isso deve−se recompilar o kernel apenas com o necessário. Uma sugestão é a separação em módulos, o que sem dúvida reduzirá o "peso" da máquina. Para recompilar o kernel deve−se baixar o fonte de uma versão estável no site http://www.kernel.org ou um mirror confiável.
Uma vez com o fonte , deve−se descompacta−lo no diretório ***/usr/src***, e então entrar no diretório ***/usr/src/linux*** .

Abaixo as etapas necessárias para "personalizar" o kernel.

`make menuconfig`
Entra no modo de configuração do kernel. Um ambiente amigável de fácil compreensão onde se pode marcar os pacotes que serão usados.

`make dep`
Acerta as dependências das bibliotecas necessárias para a compilação.

`make`
Compila o kernel.

`make modules`
Compila os módulos.

`make install`
Instala o kernel.

`make modules_install`
Instala os módulos.

VII − Agendando Processos − crontab & at

Para agendar processos muito demorados como o download de um arquivo muito grande, ou qualquer outro processo que necessite de uma execução automática, deve−se usar basicamente dois comandos: o **at** ou o **crontab**.

O comando *at* agenda a execução de um comando e logo depois que esse processo ocorre este comando não será mais executado.

Para agendar um programa com *at* usa−se:

```
echo "comandos" | at <quando>
```

ou

```
at <quando>
at> comando1
at> comando2
at> ^D      (Ctrl + D)
```

E para agendar processos que devem rodar periodicamente usa−se o comando **crontab**. Há necessidade de se editar um arquivo que possui o seguinte formato:

Minuto Hora Dia Mês Dia da Semana Comando

Onde *Minuto* é um intervalo de 0 a 59, *Hora* é um intervalo de 0 a 23, *Dia* de 1 a 31, *Mês* de 1 a 12 e os *Dias da Semana* podem variar de 0 a 6 correspondendo a variação de domingo a sábado.

As linhas que começam com o caracter "#" são ignoradas.

Para editar:

```
crontab −u <user> −e
```

Para listar:

```
crontab −u <user> −l
```

Exemplo:

```
# Para executar um comando de 2 em 2 minutos
0−59/2  *   *   *   *                /bin/ls

# Para executar um comando de 4 em 4 horas e às 18 horas
*       0−23/4, 18 * * *             /usr/bin/comando
```

VIII − Syslogd − A Caixa Preta do Linux

Para analisar o que vem ocorrendo ou já ocorreu no sistema, o linux possui o *syslogd*, que funciona como uma caixa preta, guardando em arquivos, informações como: data e hora do boot , login de usuário, e outros dados importantes para analisar o servidor.
O arquivo responsável pelas configuração do *syslogd* é o */etc/syslog.conf .*

```
*.=info;*.=notice                       /usr/adm/messages
*.=debug                                /usr/adm/debug
```

Os registros deste arquivo são divididos em ***facility*** , ***priority*** e ***destino***, podendo haver derivações como dois conjuntos "facility.priority" para um mesmo destino. As "facilities" podem ser: *auth, auth−priv, cron, daemon, kern, lpr, mail, mark, news, security* (*mesmo que auth*)*, syslog, user,uucp e local0.*

Já as "priorities" seguem em ordem crescente: *debug, info, notice*, *warning, warn* (memo que *warning*)*, err, error* (mesmo que *err*)*, crit, alert, emerg, panic* (mesmo que *emerg*)*.*

Um arquivo gerado na grande maioria dos sistemas *nix é o */var/log/messages* , este arquivo contém informações genéricas sobre todo o sistema*.* Um exemplo do conteúdo desse arquivo segue abaixo.

```
Apr  4 18:27:49 dende sshd[2080]: log: Connection from
200.245.11.158 port 1026

Apr  4 18:28:31 dende sshd[2081]: log: Connection from
200.245.11.158 port 1028

Apr  4 18:29:05 dende sshd[2080]: log: Could not reverse
map address 200.245.11.158.

May 12 09:37:00 dende su[4559]: + pts/0 vinic−root
```

Outro arquivo muito comum é o */var/log/debug* que contém informações mais detalhadas, abaixo um exemplo.

```
Apr  4 16:38:19 dende identd[1985]: Successful lookup:
1501 , 25 : vinic.root

Apr 25 13:27:36 dill kernel: VFS: Disk change detected on
device fd(2,0)
```

IX − Técnicas Básicas para Trabalhar com Redes (ifconfig, route)

O Linux é um sistema operacional totalmente compatível com redes, dos tipos mais heterogêneos.

Para listar as interfaces de rede usa−se o comando:

```
ifconfig −a
```

Para atribuir um endereço IP para uma interface usa−se:

```
ifconfig eth0 <endereço> broadcast <broadcast>
netmask <mascara>
```

Também existem as rotas para o pacote IP poder sair para fora da rede local, caso haja um roteador.

Para exibir a tabela de rotas usa−se o comando:

`route −n` ou **`netstat −nr`**

Para adicionar uma rota:

```
route add default gw <Gateway> netmask 0.0.0.0
metric 1
```

E caso haja a necessidade de excluir uma rota usa−se:

```
route del <destino>
```

X − Gerenciando os Serviços − inetd

O inetd é um daemon que pode gerenciar outros daemons, tendo uma função de supervisor. Este supervisor é executado sempre na carga do sistema, e busca a lista de serviços que devem passar pelo *inetd* no arquivo: */etc/inetd.conf .* Este arquivo possui registros no seguinte formato:

serviço tipo protocolo espera usuário servidor linha_de_comando

serviço: Fornece o nome do serviço a ser disponibilizado. Ele deve ser traduzido em um número de porta através de uma pesquisa no arquivo */etc/services.*

tipo: Especifica o tipo de conexão que será utilizada, *stream* (para conexões orientadas a protocolo) ou *dgram* (para protocolos que utilizam datagramas). Serviços baseados em TCP devem sempre ser especificados como *stream*, enquanto que serviços baseados em UDP devem sempre ser definidos como *dgram.*

protocolo: Informa o nome do protocolo usado pelo serviço. Deve ser um nome que possa ser encontrado no arquivo */etc/protocols,* como UDP, TCP, ICMP, etc...

espera: Esta opção aplica−se somente a conexões por datagramas. Ela pode ser igual a *wait* , que irá executar um daemon por vez (mono−servidor), ou *nowait* , que nos permite trabalhar com vários daemons ao mesmo tempo.

usuário: Esta é a identificação de acesso do usuário sob o qual o processo será executado. É aconselhável aplicar os usuários com a menor permissão possível como *nobody*. Apenas quando necessário deve−se usar *root.*

servidor: Fornece o caminho completo do programa servidor a ser utilizado.

linha de comando: Esta é a linha de comando a ser enviada para o servidor. Isso inclui o argumento 0, que é o nome do comando.

Abaixo um pequeno exemplo do */etc/idetd.conf*

```
imap2  stream tcp  nowait root /usr/sbin/tcpd  imapd
#pop3  stream tcp  nowait root /usr/sbin/tcpd  ipop3d
telnet stream tcp  nowait root /usr/sbin/tcpd  in.telnetd
```

Quando se faz alguma alteração nesse arquivo basta reiniciar o daemon inetd com: **`killall −HUP inetd`**

XI − Utilizando Ferramentas de Busca

Para deixar o sistema em dia, com pacotes sempre atualizados, sugere−se a pesquisa contínua nos vários sites que oferecem tais pacotes.

Ex.: http://www.freshmeat.net

XII − Instalando SSh / SShD

Problemas com o telnet (Sniffers)

O telnet antigamente era muito usado para obter acesso remoto de um servidor, mas o grande problema do telnet é que os pacotes de dados passam limpos (não criptografados) pela rede possibilitando um ataque, que não é nada mais do que a captura de todos os pacotes que percorrem na rede, inclusive senhas, números de cartão de crédito, etc ...

Solução com SSh

A solução encontrada pelos profissionais de segurança e administradores foi a substituição do *telnet* por *ssh*, que criptografa os dados dos pacotes, impossibilitando a sua fácil compreensão.

Para instalar o SSh deve−se baixar o pacote de algum lugar, como, por exemplo: *ftp://ftp2.unicamp.br/pub/security/tools/ssh/ssh−1.2.27.tar.gz*, tendo o cuidado de se baixar o produto de instituições com reconhecida credibilidade, pois do contrário, corre−se um sério risco de baixar "trojan horse". Depois de pegar o pacote:

```
tar xvzf ssh−1.2.27.tar.gz
```

Depois apenas mais três comandos dentro do diretório do ssh−1.2.27:

```
./configure ; make ; make install
```

Pronto ! O SSh já está instalado agora para colocar seu daemon no */etc/rc.d/rc.local*[1] , para isso pode−se simplesmente fazer assim :

```
echo "/usr/local/sbin/sshd" >> /etc/rc.d/rc.local
```

1 rc.local: Este arquivo é executado sempre na hora do boot, como o Autoexec.bat do DOS

Apêndice A − Recuperando a Senha do root

Quando se esquece a senha do *root* não é necessário a reinstalação de todo o sistema. Pode−se usar algumas falhas de segurança existentes quando temos acesso físico ao servidor para trocar a senha.

- ***Modo "single"***

Algumas distribuições permitem que na hora da carga (LILO), se asse o parâmetro "single" após o label correspondente a partição Linux. Na maioria das distribuições esse label é "linux", ficando assim `linux single`.

- ***Montando o Sistema de Arquivos Através de disquetes***

Pode utilizar os disquetes de instalação ou de recuperação, ou até mesmo outro sistema Linux, para montarmos o sistema de arquivos que contém o arquivo *shadow* do root que não se sabe a senha.

Basta criar um diretório vazio, ex.: **`mkdir hd,`** e usar esse novo diretório como ponto de montagem para o sistema, ex.:

```
mount −t ext2 /dev/hdxx ./hd
```

Feito isso, deve−se entrar no diretório *etc* que está dentro do diretório *hd* e editar o arquivo *shadow* excluindo o que está entre os primeiros " **:** " para a linha que correponde ao registro do *root* (possivelmente a primeira).

Apêndice B − Check−List Básico de Segurança

Para manter o servidor sempre seguro deve−se tomar cuidado com:

- ***Serviços Abertos:***

Serviços desnecessariamente abertos, são os maiores "responsáveis" pelas invasões atualmente.

- ***Senhas Fracas:***

Podem ser quebradas facilmente através de programas "Brute Force".

- ***Daemons desatualizados (sujeitos a exploits):***

Daemons com *bugs*, são visados pelos "Intrusos" devido a sua alta vulnerabilidade, e existem sites especializados em relatar tais *bugs* para qualquer pessoa.

- ***Acesso Físico:***

E é claro, deve−se tomar cuidado com o local físico onde se encontra o servidor, observando condições de possíveis acidentes, furtos, etc..

Apêndice C − Gerenciando distribuições baseadas em Red Hat

Distribuições baseadas no *Red Hat Linux* possuem uma ferramenta que facilita em muito o gerenciamento do Linux. Trata−se do **linuxconf**, que realiza de maneira integrada várias tarefas como administração de usuários, redes, e muitas outras coisas.

Para utilizar o *linuxconf* deve−se estar com as permissões de root.

Outra facilidade muito interessante que essa distribuição possui é o seu gerenciardor de pacotes, ***rpm*** (*Red Hat Package Mananger*). Para obter detalhes sobre o seu funcionamento um bom endereço é :
https://www.Dicas−L.unicamp.br/linux/rpm.html

Apêndice D − Instalando o X−Windows

Instalar o ambiente gráfico se tornou uma tarefa simples e intuitiva com o programa *XF86Setup* ou com *Xconfigurator,* que vieram substituir o *xf86config.*

O XF86Setup trabalha em modo gráfico permitindo fácil acesso entre as opções da configuração, já o Xconfigurator é mais comum nas distribuições baseadas em Red Hat, como o Conectiva Linux.

Uma outra maneira interessante de se configurar o X−Windows é editar o arquivo XF86Config encontrado nos diretórios /etc ou /etc/X11 .

Apêndice E − Níveis de Execução no Red Hat Linux

Sistemas Linux podem funcionar em vários níveis distintos de execução. Cada nível é caracterizado pelo conjunto de processos permanentes e funções oferecidas. Sistemas Red Hat Linux e derivados, utilizam seis níveis de execução, ou *runlevels*, como são mais conhecidos. Esta concepção tem suas origens no sistema Unix desenvolvido na AT&T (*System V Unix*).

Os níveis de inicialização são controlados pelos scripts que se encontram no diretório */etc/rc.d/init.d*. Uma listagem deste diretório revela os seguintes arquivos:

```
atd         gpm      keytable   mars-nwe  nfslock  rstatd
single      sshd     crond      halt      killall  named
portmap     rusersd  smb        syslog    dhcpd    httpd
linuxconf   netfs    postgresql rwhod     snmpd    xfs
functions   inet     lpd        network   random   sendmail
squid       ypbind
```

Uma rápida inspeção nos revela vários serviços conhecidos como *sendmail*, *httpd* e *named*.
Cada nível de execução é controlado através de links simbólicos existentes nos seis diretórios (*/etc/rc.d/rc1.d* até */etc/rc.d/rc6.d*). Examinemos o conteúdo do diretório */etc/rc1.d*:

```
# ls −l
lrwxrwxrwx 1 root  root 19 Mar 22 21:02 K00linuxconf −> ../init.d/linuxconf
lrwxrwxrwx 1 root  root 18 Mar 22 20:54 K05keytable −> ../init.d/keytable
lrwxrwxrwx 1 root  root 13 Mar 22 20:59 K10xfs −> ../init.d/xfs
lrwxrwxrwx 1 root  root 13 Mar 22 20:58 K15gpm −> ../init.d/gpm
lrwxrwxrwx 1 root  root 15 Mar 22 20:53 K15httpd −> ../init.d/httpd
lrwxrwxrwx 1 root  root 18 Mar 22 21:05 K30sendmail −> ../init.d/sendmail
lrwxrwxrwx 1 root  root 14 Mar 22 21:03 K50inet −> ../init.d/inet
lrwxrwxrwx 1 root  root 13 Mar 22 20:53 K60atd −> ../init.d/atd
lrwxrwxrwx 1 root  root 15 Mar 22 21:06 K60crond −> ../init.d/crond
lrwxrwxrwx 1 root  root 13 Mar 22 21:02 K60lpd −> ../init.d/lpd
lrwxrwxrwx 1 root  root 15 Mar 22 20:59 K75netfs −> ../init.d/netfs
lrwxrwxrwx 1 root  root 17 Mar 22 20:59 K90network −> ../init.d/network
lrwxrwxrwx 1 root  root 16 Mar 22 21:06 K99syslog −> ../init.d/syslog
lrwxrwxrwx 1 root  root 16 Mar 22 20:59 S00single −> ../init.d/single
```

Como se pode ver, todo o conteúdo do diretório */etc/rc1.d* consiste de links simbólicos apontando para scripts dentro do diretório */etc/rc.d/init.d*. A primeira letra dos nomes dos links simbólicos pode ser ou "*S*" ou "*K*", indicando se o processo para o qual aponta deve ser ativado (Started) ou desativado (Killed). O número que se segue a esta letra indica a ordem em que os processos devem ser encerrados ou ativados. Em nosso exemplo o primeiro processo a ser desativado é o *linuxconf*. O primeiro a ser ativado, após terem sido encerrados todos os demais processos, é o script *single*.

Cada um dos scripts residentes no diretório */etc/rc.d/init.d* aceita geralmente três parâmetros: *start*, *stop*, *restart*. Estes parâmetros indicam, respectivamente, a ativação, desativação e desativação seguida de ativação do processo.

Para determinar o nível de execução em que seu sistema está funcionando, utilize o comando */sbin/runlevel*. Este comando irá consultar o arquivo */var/run/utmp* para determinar o estado atual e o anterior. Caso o estado anterior não possa ser determinado é exibida a letra "*N*" em seu lugar:

```
# /sbin/runlevel
 N 3
```

Descrição dos Níveis de Execução

A seguir listamos os estados possíveis de um sistema Linux e sua descrição:

- Nível 0

Neste nível o sistema está parado

- Nível 1

Sistemas funcionando no nível 1 estão em modo monousuário, com um conjunto mínimo de processos ativos. O sistema de arquivos raiz (root) está montado em modo de leitura. Este nível de execução é normalmente utilizado quando a inicialização normal falha por alguma razão.

- Nível 2

A maior parte dos serviços estão ativos, com exceção dos processos de rede (como nfs, nis, named e httpd).

- Nível 3

Este é o nível normal de operação, com todos os processos ativos.

- Nível 4

Este nível não é utilizado na maior parte das distribuições

- Nível 5

Semelhante ao nível 3, com todos os processo ativos, porém com uma interface gráfica de logon

- Nível 6

É executado neste nível um reboot do sistema.

Alteração dos Níveis de Execução

Os níveis de execução podem ser mudados pelo superusuário com o sistema em funcionamento. Sempre que é alterado um nível de execução são comparados, nos dois níveis, os processos que devem ser ativados e quais devem ser desativados. O processo *init*, que é processo pai de todos os demais (PID 1), compara a lista dos processos que devem ser encerrados no diretório indicativo do nível de execução atual com a lista dos processos que devem ser ativados no nível de execução de destino. De posse desta informação o processo init determinará quais processos devem ser ativados ou desativados.

Para reiniciar o sistema basta executar o comando

```
init 6
```

Veja a lista dos links em */etc/rc.d/rc6.d*:

```
K00linuxconf
K05keytable
K10xfs
K15gpm
K15httpd
K30sendmail
K50inet
K60atd
K60crond
K60lpd
K75netfs
K80random
K89portmap
K90killall
K90network
K99syslog
S00reboot
```

Como se pode ver, a maioria dos links inicia−se com a letra "*K*", indicando que os processos serão desativados. Apenas um link inicia−se por "*S*", *S00reboot*, que aponta para o script */etc/init.d/halt*.

Similarmente, para colocar o sistema em modo monousuário

```
init 1
```

Nível de Execução Padrão

O nível em que o sistema irá funcionar é indicado pela entrada

```
id:3:initdefault:
```

do arquivo */etc/inittab*. Neste sistema o nível padrão de execução é *3*. Para alterar este nível de execução basta alterar o número "*3*" para o valor desejado. Nunca altere este valor para "*0*" ou "*6*", que indicam, respectivamente, o sistema parado ou em modo de encerramento.

Definição ou Remoção de Processos Residentes

Para desativar um serviço de um determinado nível de execução basta remover o link simbólico do diretório apropriado. Por exemplo, para desativar o serviço *httpd*, do nível de execução *3*, basta remover o link */etc/rc.d/rc3.d/S85httpd* do diretório */etc/rc.d/rc3.d*.

Similarmente, para inserir um novo serviço, basta criar um link no diretório padrão de execução, apontando para o script correspondente em */etc/rc.d/init.d*:

```
# cd /etc/rc.d/rc3.d
# ln −s /etc/rc.d/init.d S99local
```

Este script realmente existe e é normalmente utilizado para inserir os serviços locais. Pela numeração (99), este script sempre será o último a ser ativado.

Importante, certifique−se de escolher uma numeração que posicione a ativação do script na ordem correta. Se o serviço é dependente do funcionamento da rede ele deve necessariamente ser ativado após estes serviços estarem ativos.

Utilitários para Configuração dos Níveis de Execução

Até agora abordamos a configuração manual dos scripts de inicialização. Existem entretanto diversos utilitários para realizar este este trabalho.

- chkconfig

Utilitário para configuração dos níveis de execução invocado a partir da linha de comandos

- ksysv

Utilitário gráfico que permite a configuração dos níveis de execução e ativação e desativação de processos individuais

- linuxconf

Ferramenta genérica de configuração que permite o gerenciamento dos níveis de execução

A forma mais segura de se lidar com esta configuração certamente começa com o entendimento perfeito de seu funcionamento. As interfaces gráficas podem obscurecer o significado real do que se está fazendo e conduzir a uma configuração indesejável. Em qualquer situação, realizando−se o trabalho manualmente ou através de utilitários, recomendamos o backup de todos os arquivos envolvidos para poder retornar à uma situação estável em caso de problemas.

Para lidar com a ativação e desativação de processos residentes, algo que frequentemente precisamos fazer sempre que alteramos a configuração de um serviço, precisamos realizar os seguintes passos:

```
# cd /etc/rc.d/init.d
# httpd restart
```

Em nosso exemplo nos dirigimos ao diretório onde ficam todos os scripts de ativação de processos e invocamos o script httpd com o parâmetro "*restart*".

Um artifício engenhoso para realizar esta tarefa nos é fornecido, através de um alias, em sistemas *Conectiva Linux*. Este alias, disponível no ambiente do usuário *root*, chama−se *cds*:

```
alias cds='cd /etc/rc.d/init.d && ls'
```

Como podemos ver, o alias realiza dois passos: a mudança para o diretório */etc/rc.d/init.d* e a listagem de seu conteúdo. Desta forma podemos visualizar o nome de todos os scripts disponíveis, facilitando a invocação do script apropriado. Simples, mas extremamente útil.

Referências Adicionais

No transcorrer deste artigo foram feitas referências aos comandos *ln*, *init*, *chkconfig*, *inittab* e *runlevel*. A leitura da documentação destes comandos fornece informações valiosas sobre todo o processo descrito e pode ser acessada a partir do comando *man* (*man ln*, *man init*, etc).

Artigo escrito por
Rubens Queiroz de Almeita na
Revista de Informação e Técnologia
http://www.revista.unicamp.br