

ISCTE-IUL – Instituto Universitário de Lisboa
DCTI
Sistemas Operativos

Administração Linux

JO, versão 1.02 – 26-04-2011

1. Estrutura do sistema de ficheiros	2
1.1 - Introdução	2
1.2 - A directoria raiz (root)	2
1.3 - A directoria /etc	3
1.4 - A directoria /dev	4
1.5 - A directoria /usr	4
1.6 - A directoria /var	5
1.7 - A directoria /proc	6
2. Gestão de contas	8
2.1 - Conta de utilizador	8
2.2 - Criar um utilizador	8
2.2.1 - /etc/passwd	8
2.2.2 - Ambiente inicial: /etc/skel	8
2.2.3 - Criar manualmente um utilizador	9
2.3 - Remover um utilizador	9
3. Complementos	10
3.1 - Os ficheiros .profile e .bashrc	10
3.2 - O processo cron	10
3.2.1 - Formato do ficheiro	10
3.2.2 - Definir um crontab para um utilizador	12

1. Estrutura do sistema de ficheiros

Esta secção descreve as partes mais importantes do sistema de ficheiros Linux, baseado no Filesystem Hierarchy Standard (FHS). A motivação para a divisão da hierarquia do sistema de ficheiros em diferentes ramos é apresentada de seguida.

1.1 - Introdução

Os sistemas de ficheiros encontra-se normalmente organizados sob a forma de uma árvore. As folhas, ou os nós da árvore, correspondem a directorias ou ficheiros. A árvore começa na raiz (root), a que corresponde a directoria /.

1.2 - A directoria raiz (root)

A raiz do sistema de ficheiros contém várias outras directorias, comuns aos vários sistemas Unix-like, que agrupam os diferentes tipos de ficheiros, de acordo com a sua semântica:

`/bin`

Comandos necessários durante o processo de arranque do sistema operativo, que também possam ser usados pelos utilizadores (provavelmente após o processo de arranque).

`/sbin`

Comandos que não se destinam aos utilizadores normais, mas sim ao utilizador `root`. Usualmente esta directoria não se encontra na `PATH` dos utilizadores normais. Como exemplo de comandos guardados nesta directoria temos:

`ifconfig` – configura as interfaces de rede;

`modprobe` – carrega módulos no kernel do sistema operativo;

`init` – será explicado mais à frente.

`/etc`

Nesta directoria são guardados os ficheiros de configuração específicos da máquina onde o sistema operativo se encontra instalado.

`/root`

A directoria home do utilizador `root`.

`/lib`

Directorias onde estão guardadas as bibliotecas partilhadas (shared libraries) necessárias à execução dos programas.

`/dev`

Os ficheiros guardados nesta directoria servem como interface com diferentes dispositivos, por exemplo, teclado, discos rígidos, placas de rede etc... Não são ficheiros editáveis.

`/tmp`

Nesta directoria todos os utilizadores podem ler e/ou escrever. Esta serve para os diferentes programas escreverem em disco informação temporária. Nos sistemas Linux, normalmente os utilizadores têm um limite de espaço em disco para a directoria pessoal. A `/tmp` normalmente é limitada ao espaço da raiz /.

`/boot`

Nesta directoria guardam-se os ficheiros usados pelo bootstrap loader, por exemplo, LILO ou GRUB. As imagens do kernel são normalmente guardadas aqui. Esta directoria é

normalmente guardada numa partição início do disco, ou eventualmente guardada separadamente noutro sistema de ficheiros.

`/mnt , /media, /Volumes`

Estas directorias servem para montar (mount) temporariamente dispositivos de armazenamento. Tipicamente, quando se liga uma pen ou se introduz um CDROM, é criada uma sub directoria para cada dispositivo. Esta tarefa pode ser feita automaticamente pelo sistema operativo, ou pelo administrador (`root`).

`/proc, /usr, /var, /home`

Estas directorias servem como pontos onde podem ser montados outros sistemas de ficheiros importantes (que veremos a seguir). A directoria `/proc` é um caso particular de uma directoria que não reside em nenhum disco.

1.3 - A directoria `/etc`

A directoria `/etc` contém os ficheiros de configuração do sistema operativo, bem como de diversos serviços (ex: servidor web, bases de dados, etc..). Alguns dos mais importantes descrevem-se a seguir:

`/etc/rc` ou `/etc/rc.d` ou `/etc/rc?.d`

Scripts ou directorias que contém scripts que são executados no arranque do sistema operativo ou quando se muda de run level.

`/etc/passwd`

Ficheiro que contém informação sobre os utilizadores do sistema. Cada linha do ficheiro corresponde a um utilizador e inclui informação como o username, o nome real, a directoria home do utilizador, a shell, entre outros.

`/etc/shadow`

Ficheiro encriptado que contém as passwords dos utilizadores.

`/etc/group`

Ficheiro similar ao `/etc/passwd`, mas contém informação sobre os grupos.

`/etc/inittab`

Contém informação para o processo `init`.

`/etc/fstab`

Lista os sistemas de ficheiros montados automaticamente pelo sistema operativo.

`/etc/mtab`

Lista os sistemas de ficheiros actualmente montados. O conteúdo deste ficheiro é actualizado pelos scripts de arranque, e automaticamente actualizado pelo comando `mount`.

`/etc/login.defs`

Ficheiro de configuração para o comando `login`.

`/etc/profile, /etc/bash.rc, /etc/csh.cshrc`

Estes ficheiros são executados aquando do login de um utilizador ou durante o arranque da shell (BASH, C shell etc...). Estes scripts permitem ao administrador definir variáveis globais ou executar scripts automaticamente por todos os utilizadores do sistema.

`/etc/hostname`

Define o nome do sistema operativo.

`/etc/modules`

Este ficheiro contém a lista de módulos a serem carregados pelo kernel.

1.4 - A directoria /dev

A directoria /dev contém ficheiros especiais para os diferentes dispositivos. Estes ficheiros são criados aquando da instalação e mais tarde através do script /dev/MAKEDEV. A lista que se apresenta a seguir não é exaustiva e muitos dos dispositivos presentes nesta directoria necessitam de suporte no kernel.

/dev/hda

O ficheiro /dev/hda é o primeiro disco IDE master no controlador IDE principal. O ficheiro /dev/hdb é o disco slave. Da mesma forma os ficheiros /dev/hdc e /dev/hdd são os discos master e slave do controlador secundário. Cada disco está dividido em partições. As partições 1 a 4 são partições primárias enquanto que as partições com número superior ou igual 5 são partições lógicas. Cada partição é acessível a partir de um ficheiro. Por exemplo, /dev/hda1 refere-se à partição primária 1 do disco master do controlador principal.

/dev/sda

Tal como no caso dos controladores IDE, os dispositivos da família /dev/sd* referem-se a dispositivos SCSI. Nesta categoria incluem-se também os discos da família SATA I e SATA II. Por exemplo: /dev/sda1 refere-se à partição principal do disco principal do controlador SCSI (ou SATA).

/dev/null

O dispositivo /dev/null funciona como uma espécie de buraco negro: para este dispositivo pode-se enviar dados que serão simplesmente descartados. Tal como já foi visto, este dispositivo pode ser usado para reencaminhar o output de um comando quando não se está interessado em visualizar qualquer informação produzida pelo mesmo no terminal (ou consola).

/dev/random ou /dev/urandom

Estes ficheiros são geradores de números aleatórios do kernel.

/dev/zero

Este ficheiro serve apenas para obter múltiplos zeros. Qualquer leitura deste dispositivo devolverá 0.

1.5 - A directoria /usr

A directoria /dev/usr é geralmente grande uma vez que todos os programas são instalados aqui. Usualmente os ficheiros instalados na directoria /usr vêm com a distribuição. Programas instalados localmente e outros usualmente são instalados na directoria /usr/local. Esta subdivisão tem a vantagem de não ser necessário instalar todos os programas novamente quando se faz, por exemplo, uma actualização da distribuição. Algumas das subdirectorias importantes descrevem-se a seguir.

/usr/X11R6

Todos os ficheiros do sistema de janelas X.

/usr/bin

Quase todos os comandos a serem usados por todos os utilizadores. Alguns comandos encontram-se na directoria /usr/bin e outros na /usr/local/bin

`/usr/sbin`

Comandos para o administrador de sistemas.

`/usr/share/man`, `/usr/share/doc`, `/usr/share/info`

Documentação: man pages, e outro tipo de documentação do sistema operativo.

`/usr/include`

Ficheiros `.h` (headers) para a linguagem de programação C. Note-se que o sistema operativo é ele próprio escrito em C. Por razões de consistência, os headers devem estar situados na directoria `/usr/lib`, mas por razões de tradição mantem-se esta directoria.

`/usr/lib`

Local onde são guardadas a maior parte das bibliotecas dos programas.

`/usr/local`

Local onde devem ser instalados localmente programas e outros ficheiros. As distribuições de Linux não deve escrever nada aqui. Esta directoria é reservada para o uso por parte do administrador.

1.6 - A directoria `/var`

A directoria `/var` contém ficheiros cuja informação é alterada durante a execução do sistema operativo. Esta directoria é específica para cada sistema.

`/var/cache/man`

Uma cache para as man pages. Funciona como uma memória temporária.

`/var/lib`

Ficheiros que mudam durante a execução do sistema operativo.

`/var/local`

Local onde os programas situados em `/usr/local` (i.e., programas instalados pelo administrador), podem escrever temporariamente.

`/var/lock`

Lock files. Estes ficheiros costumam indicar que um ficheiro está a usar um determinado dispositivo ou aceder a um determinado ficheiro. Funciona como uma espécie de flag: um programa que tente aceder a um dispositivo, consulta primeiro esta directoria para verificar se existe um ficheiro de lock, e caso exista, não acede ao dispositivo ou ficheiro em questão.

`/var/log`

Esta directoria contém a maior parte dos logs do sistema operativo. Especialmente importantes são os ficheiros `/var/log/wtmp` criado pelo login, e o ficheiro `/var/log/messages` mantido pelo programa syslog, que guarda todas as mensagens do kernel ou programas de sistema.

`/var/mail`

Nesta directoria é guardado todo o email para os diferentes utilizadores. Alternativamente pode ser usada da directoria `/var/spool/mail`

`/var/spool`

Directoria onde são armazenados ficheiros de notícias (news), filas de impressora (queues), e outros tipos de filas de espera.

`/var/run`

Contém informação acerca do sistema operativo apenas válida até a próxima vez que o sistema é reiniciado.

`/var/tmp`

Directoria usada para os programas escreverem temporariamente. Usualmente é usada quando os ficheiros têm uma dimensão maior ou quer-se que durem mais tempo que os armazenados na directoria `/tmp`.

1.7 - A directoria `/proc`

A directoria `/proc` contém um sistema de ficheiros virtual. Estes ficheiros não existem fisicamente no disco, mas sim na memória. O conteúdo destes ficheiros serve para fornecer informação sobre o sistema operativo. Alguns dos ficheiros importantes descrevem-se a seguir.

`/proc/1`

É uma directoria com informação acerca do processo número 1. Cada processo tem uma subdirectorio na directoria `/proc`, sendo o nome o pid do processo.

`/proc/cpuinfo`

Informação sobre o processador.

`/proc/devices`

Lista dos dispositivos configurados no kernel que está de momento a correr.

`/proc/dma`

Mostra quais os canais DMA que estão a ser usados a cada momento

`/proc/filesystems`

Lista quais os sistemas operativos configurados no kernel.

`/proc/interrupts`

Mostra quais as interrupções que estão a ser usadas, e quantas interrupções de cada uma já houve.

`/proc/ioports`

Quais os portos I/O que estão a ser usados no momento.

`/proc/kcore`

Uma imagem da memória física do sistema. Tem exactamente o mesmo tamanho da memória física, mas não ocupa esse valor. Note-se que qualquer ficheiro em `/proc` não ocupa qualquer espaço em disco.

`/proc/kmsg`

Mensagens geradas pelo kernel.

`/proc/loadavg`

Contém o valor 'Load Average' do sistema.

`/proc/meminfo`

Informação sobre o uso memória, tanto física como da swap.

`/proc/modules`

Quais os módulos actualmente carregados pelo kernel.

`/proc/net`

Informação sobre os protocolos de rede.

`/proc/stat`

Várias estatísticas acerca do sistema, tais como o número de 'page faults' desde que o sistema arrancou.

`/proc/uptime`

Há quanto tempo que o sistema se iniciou.

`/proc/version`

Versão do kernel.

2. Gestão de contas

2.1 - Conta de utilizador

Quando um sistema operativo é usado por muitos utilizadores, torna-se necessário diferenciar os utilizadores, por exemplo, para manter os seus ficheiros privados. Esta necessidade mantém-se mesmo que apenas um utilizador possa estar a usar o computador em simultâneo. Cada utilizador tem um portanto um username que é único e serve para entrar no sistema operativo (login).

Uma conta de utilizador é mais do que um simples username. Representa todos os ficheiros, recursos e informação relativa a um utilizador.

2.2 - Criar um utilizador

O kernel do sistema operativo trata cada utilizador como sendo um simplesmente um número. Cada utilizador tem assim associado um número único denominado user id (ou uid). O nome do utilizador, bem como informação adicional, é guardado num ficheiro separado. Para criar um utilizador, a maior parte das distribuições de Linux têm programas para isso. Nomeadamente adduser e useradd.

2.2.1 - /etc/passwd

O ficheiro /etc/passwd é o ficheiro onde a informação principal associada a cada utilizador é guardada. Este ficheiro tem uma linha por utilizador, com campos separados por (:):

- username;
- campo onde a password foi guardada em tempos;
- user id (uid);
- group id (gid);
- nome textual do utilizador;
- directoria home;
- a shell de login.

A password de cada utilizador é encriptada e guardada num ficheiro à parte. Usualmente este ficheiro, /etc/shadow, apenas é acessível pelo administrador (root).

Exemplo da informação de um utilizador:

```
labac:x:1000:1000:Utilizador labac:/home/labac:/bin/bash
```

2.2.2 - Ambiente inicial: /etc/skel

Quando um utilizador é criado, a directoria home é inicializada com os ficheiros existentes na directoria /etc/skel. Este é o sítio ideal onde o administrador deve colocar os ficheiros que deverão ser

comuns a todos os novos utilizadores. O seu conteúdo deve ser mantido o mais pequeno possível, pois tornar-se-á difícil posteriormente actualizar o conteúdo dos diferentes utilizadores já criados.

2.2.3 - Criar manualmente um utilizador

Criar um utilizador manualmente envolve os seguintes passos:

- editar o ficheiro `/etc/passwd` com o comando `vi` e adicionar uma nova linha para o utilizador. Este comando tem como vantagem bloquear o acesso ao ficheiro, de forma a que outros programas tentem actualizá-lo ao mesmo tempo. O campo da password deve ser inicializado com `*` (asterisco) de forma a ser impossível ao utilizador efectuar o login;
- de forma igual, editar o ficheiro `/etc/group` com o comando `vi`, se for necessário criar um novo grupo;
- criar a directoria home do utilizador com o comando `mkdir`;
- copiar os ficheiros da directoria `/etc/skel` para a nova directoria home;
- mudar o dono do ficheiro e as permissões usando os comandos `chown` e `chmod`; a opção `-R` é útil para aplicar recursivamente a todos os ficheiros da directoria actual e respectivas subdirectorias:

```
cd /home/novoutilizador
chown -R username.group .
chmod -R go=u, go-w .
chmod -R go= .
```

- mudar a password do utilizador com o comando `passwd`.

Nota: estas operações devem ser realizadas como administrador, i.e., como root. Para tal pode usar-se o comando `su` para passar o utilizador actual para super user. Nas distribuições mais recentes, o administrador costuma não ter login, pelo que alternativamente é apenas possível invocar comandos como sendo administrador, precedendo o comando a executar de `sudo`.

Por exemplo, para mudar as permissões dos ficheiros usar-se-ia o seguinte comando:

```
sudo chown -R username.group .
```

Nota 2: os comandos `adduser` e `useradd` evitam os passos anteriores.

2.3 - Remover um utilizador

Remover um utilizador significa apagar toda a informação relativa ao mesmo. No entanto, os ficheiros de um utilizador podem encontrar-se espalhados pelo sistema de ficheiros. Isto é, os ficheiros de um utilizador não ficam limitados à directoria home. Por exemplo, o email é por vezes guardado na directoria `/var/mail` ou `/var/spool/mail`. É por isso necessário procurar os ficheiros do utilizador pelo sistema de ficheiros.

Existem dois comandos para remover utilizadores que fazem isso automaticamente. São eles o `deluser` e o `userdel`.

3. Complementos

3.1 - Os ficheiros `.profile` e `.bashrc`

Na directoria home de cada utilizador existem um conjunto de ficheiros com definições e/ou configurações especiais para cada utilizador. Usualmente o nome destes ficheiros começa por um `.`, o que significa que são ficheiros “escondidos” não visíveis através dos comandos `ls` ou `ls -l`. Para visualizar os ficheiros escondidos é necessário usar a opção `-a` do `ls`, por exemplo, executando o comando `ls -la`.

O ficheiro `.profile` é um script que é executado quando é feito o login do utilizador. Este script é executado depois do `/etc/profile`, que é suposto ter definições gerais para todos os utilizadores. Este ficheiro é usado por exemplo para definir variáveis de ambiente, definidas como já foi visto pelo comando `export`.

O ficheiro `.bashrc` é o ficheiro de configuração da `bash`. É executado cada vez que se inicia a `bash`.

3.2 - O processo `cron`

O processo `cron` é um processo especial que permite executar/agendar tarefas num determinado instante de tempo. Este instante pode ser definido como um intervalo regular, ou mesmo como uma determinada hora de um determinado dia. Os ficheiros de configuração dos diferentes utilizadores encontram-se dentro da directoria `/var/spool/cron/crontabs`. O seu conteúdo não deve ser alterado directamente, mas sim mediante o uso do comando `crontab`.

Adicionalmente o processo `cron` lê o ficheiro `/etc/crontab`, que tem um formato diferente dos anteriores. Por exemplo, inclui o campo de utilizador para permitir executar um determinado comando/script com as permissões do respectivo utilizador.

Todos os ficheiros de configuração da subdirectoria `/etc/cron.d` são também lidos; embora com um formato semelhante do `/etc/crontab`, estes não herdam por exemplo variáveis de ambiente, e o seu uso é mais restrito. Da mesma forma, todos os ficheiros que se encontram nas directorias `/etc/cron.daily`, `/etc/cron.weekly` e `/etc/cron.monthly` são lidos.

Uma vez ocorrida uma alteração a qualquer um dos ficheiros de configuração, não é necessário executar novamente o programa `cron`. Na verdade, este verifica de minuto a minuto se houve alguma alteração em algum ficheiro de configuração mediante a consulta do campo `modtime` do sistema de ficheiros.

O administrador deverá usar o ficheiro `/etc/crontab` e não a subdirectoria `/etc/cron.d`.

Os utilizadores podem também definir os seus comandos/scripts. Para isso deverão ter permissão no ficheiro `/etc/cron.allow` ou não estar listados no ficheiro `/etc/cron.deny`. Se estes ficheiros não existirem, dependendo da configuração do sistema, apenas o administrador (`root`) poderá usar este comando.

Para mostrar o `crontab` do utilizador deverá usar-se o comando:

```
crontab -l
```

3.2.1 - Formato do ficheiro

Um ficheiro `crontab` contém instruções para o programa `cron`. Linhas que comecem por um `#` são ignoradas. Comentários em linhas após comandos são interpretados como fazendo parte do próprio comando.

Uma linha no ficheiro representa ou um comando ou uma definição de ambiente. Por exemplo

nome = valor

A string valor pode ser definida entre aspas para conter por exemplo espaços. No entanto não são feitas substituições de variáveis ambientes, e linhas como

```
PATH = $HOME/bin:$PATH
```

não funcionam como esperado. Algumas variáveis são automaticamente definidas. A shell é definida como sendo /bin/sh e LOGNAME e HOME são definidos a partir do ficheiro /etc/passwd para o respectivo utilizador.

Os comandos são executados pelo cron quando o minuto, a hora, e o mês do ano forem iguais à data actual. Um comando é composto por 6 campos, sendo os primeiros 5 dedicados à data e hora e o sexto ao comando a executar propriamente dito. Os campos da hora e data são (por ordem):

Campo	Valores permitidos
minuto	0-59
hora	0-23
día do mês	1-31
mês	1-12
día da semana	0-7

Um campo pode também ser um asterisco (*), que significa "primeiro-ultimo" valor. O campo pode também ser definido como um intervalo. Um intervalo é definido como sendo dois números separados por um hífen. Por exemplo "0-3" na hora, significa executar o comando às 0h, 1h, 2h e 3h.

Pode também especificar-se um conjunto discreto de valores, devendo estes ser definidos entre aspas e separados por vírgulas. No caso dos minutos, "10,15,20" significa executar o comando aos 10, 15 e 20 minutos da hora, respectivamente. Nos intervalos é possível também definir o passo da repetição usando o carácter /. Por exemplo, "1-9/2" é o mesmo que "1,3,5,7,9", ou "*/2" no campo das horas significa de duas em duas horas.

Exemplo de um crontab:

```
# use /bin/sh to run commands, overriding the default set by cron
SHELL=/bin/sh
# mail any output to `paul', no matter whose crontab this is
MAILTO=paul
#
# run five minutes after midnight, every day
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 2:15pm on the first of every month -- output mailed to paul
15 14 1 * * $HOME/bin/monthly
# run at 10 pm on weekdays, annoy Joe
0 22 * * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%
23 0-23/2 * * * echo "run 23 minutes after midn, 2am, 4am ..., everyday"
5 4 * * sun echo "run at 5 after 4 every sunday"
```

O ficheiro /etc/crontab tem uma estrutura idêntica, no entanto existe um campo adicional para o utilizador, a seguir aos campos da data e hora e o comando.

3.2.2 - Definir um crontab para um utilizador

Para definir um crontab para um determinado utilizador, deve primeiro criar-se um ficheiro com os comandos para o cron. Considere o ficheiro cron.txt que contém o seguinte:

```
* * * * * date >> $HOME/ficheiro.txt
```

De seguida deverá executar-se o comando:

```
crontab cron.txt
```

Para verificar que o comando foi bem sucedido pode verificar-se com

```
crontab -l
```

O exemplo anterior executa o comando date >> \$HOME/ficheiro.txt. De minuto a minuto, uma linha é acrescentada ao ficheiro com a data e a hora a que o comando foi executado.

Para apagar o crontab anterior pode usar-se o comando:

```
crontab -r
```